



Co-funded by the
Erasmus+ Programme
of the European Union



ПРОЕКТ ЕРАЗМУС+
GAMEHUB: «СПІВРОБІТНИЦТВО МІЖ УНІВЕРСИТЕТАМИ
ТА ПІДПРИЄМСТВАМИ В СФЕРІ ГРАЛЬНОЇ ІНДУСТРІЇ В УКРАЇНІ»

ERASMUS+ PROJECT
GAMEHUB: «UNIVERSITY-ENTERPRISES COOPERATION
IN GAME INDUSTRY IN UKRAINE»

ОСНОВИ КОМП'ЮТЕРНИХ ІГОР ТА ІГРОВИХ ПРОГРАМ





Co-funded by the
Erasmus+ Programme
of the European Union



ГАМЕНУВ: «СПІВРОБІТНИЦТВО МІЖ УНІВЕРСИТЕТАМИ
ТА ПІДПРИЄМСТВАМИ В СФЕРІ ГРАЛЬНОЇ ІНДУСТРІЇ В УКРАЇНІ»

ОСНОВИ КОМП'ЮТЕРНИХ ІГОР ТА ІГРОВИХ ПРОГРАМ

Харків
«Друкарня Мадрид»
2018

УДК 004.9
ББК 32.973-018.2
в6
Н 62

Проект ЕРАЗМУС+
GameHub: «Співробітництво між університетами та підприємствами
в сфері гральної індустрії в Україні»
№ 561728-EPP-1-2015-1- ES-EPPKA2-CBHE-JP

Нікітін С. О.

Н 62 Основи комп'ютерних ігор та ігрових програм : довідник модуля. / С. О. Нікітін,
Л. О. Нікітіна. – Х. : «Друкарня Мадрид», 2018. – 138 с.
ISBN 978-617-7683-01-7

Довідник модуля «Основи комп'ютерних ігор та ігрових програм» написаний згідно з останніми освітньо - кваліфікаційними вимогами до підготовки бакалаврів з спеціальності інформаційні технології. З урахуванням кредитно-модульної системи навчальна дисципліна формується як система змістовних модулів. Довідник модуля складеться з інформації стосовно модуля та критеріїв оцінювання, лекційного та лабораторних робіт. Призначено для студентів всіх спеціальностей вузів, викладачів курсу «Основи комп'ютерних ігор та ігрових програм», фахівців у галузі інформаційні технології які підвищують кваліфікацію.

Бібліографічні описи здійснено відповідно до існуючих державних та міждержавних стандартів: ДСТУ ГОСТ 7.1:2006 «Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання».

A guide to the module “Fundamentals of computer games and gaming applications” is written in accordance with the latest educational qualification requirements for the preparation of bachelors in the specialty of information technology. Taking into account the credit-module system, the discipline is formed as a system of content modules. The module's guide will consist of information on the module and criteria for evaluation, lecture and laboratory work. It is intended for students of all specialties of high schools, course teachers “Fundamentals of computer games and gaming programs”, specialists in the field of information technology who enhance their qualifications.

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

УДК 004.9
ББК 32.973-018.2
в6

ISBN 978-617-7683-01-7

© Проект ЕРАЗМУС+
GameHub: «Співробітництво між університетами
та підприємствами в сфері гральної індустрії в
Україні», 2018
© С. О. Нікітін, Л. О. Нікітіна, 2018
© «Друкарня Мадрид», 2018



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
Цей твір ліцензовано на умовах Ліцензії Creative Commons Із Зазначенням Авторства — Некомерційна — Поширення На Тих Самих Умовах 4.0 Міжнародна

ЗМІСТ

ДОВІДНИК МОДУЛЯ

1 ВСТУП	6
2 ОПИС МОДУЛЯ	6
3 ПЕРЕЛІК КОМПЕТЕНЦІЙ ТА РЕЗУЛЬТАТИ НАВЧАННЯ	7
4 МІЖДІСЦИПЛІНАРНІ ЗВ'ЯЗКИ	8
5 МЕТА ТА ПЕРЕДБАЧУВАНІ РЕЗУЛЬТАТИ ВІВЧЕННЯ МОДУЛЯ	8
6 КАЛЕНДАРНИЙ ПЛАН СЕМЕСТРУ І СТРУКТУРА МОДУЛЯ	9
7 ФОРМИ НАВЧАННЯ	9
8 ПОРЯДОК ПРОВЕДЕННЯ АТЕСТАЦІЇ	10
10 ВИКЛАДАЦЬКИЙ СКЛАД ТА ДОПОМІЖНІ ДЖЕРЕЛА	13
11 НАВЧАЛЬНА ПРОГРАМА І МАТЕРІАЛИ	14
12 СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ	20

ЛЕКЦІЇ

ЛЕКЦІЯ 1 КОМП'ЮТЕРНІ ІГРИ	22
ЛЕКЦІЯ 2 РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ	38
ЛЕКЦІЯ 3 МЕНЕДЖМЕНТ GAME-ПРОЕКТІВ	49

ЛАБОРАТОРНІ РОБОТИ

ЛАБОРАТОРНА РОБОТА №1 ВІВЧЕННЯ ІНТЕРФЕЙСУ CLICKTEAM FUSION	70
ЛАБОРАТОРНА РОБОТА №2 РОЗМІЩЕННЯ ФОНУ І ОБ'ЄКТІВ В ДОДАТКУ CLICKTEAM FUSION	87
ЛАБОРАТОРНА РОБОТА №3 АНІМАЦІЯ ПЕРСОНАЖІВ У ДОДАТКУ CLICKTEAM FUSION	102
ЛАБОРАТОРНА РОБОТА №4 СТВОРЕННЯ ПОДІЙ І ДІЙ У ДОДАТКУ CLICKTEAM FUSION	119
ЛАБОРАТОРНА РОБОТА №5 ЗВУКОВЕ ОФОРМЛЕННЯ І ЗАВЕРШЕННЯ ГРИ В ДОДАТКУ CLICKTEAM FUSION	131



Довідник модуля

1 ВСТУП

Модуль знайомить студентів з класифікацією та основами проектування ігрових програм, основами командної розробки та особливостями розробки для існуючих платформ. Отримані знання допоможуть розробляти ігрові додатки та їх компоненти з використанням сучасних технологій та технічних засобів.

Метою викладання навчального модуля «Основи комп'ютерних ігор та ігрових програм» є ознайомлення студентів з класифікацією та основами проектування ігрових програм, основами командної розробки та особливо розробки ігрових додатків для існуючих платформ.

Основними завданнями вивчення навчального модуля «Основи комп'ютерних ігор та ігрових програм» є набуття знань про сучасні підходи до класифікації та побудови ігрових додатків, набуття навичок у розробці ігрових програмних систем та їх компонентів.

Змістовний модуль «Основи комп'ютерних ігор та ігрових програм» орієнтований на оволодіння студентами практичних навичок використання засобів розробки ігрових додатків для існуючих платформ.

Модуль має метою:

- забезпечити вміння використовувати різні мови програмування для розробки ігрових додатків;
- отримання студентами відомостей про сучасні комп'ютерні ігрові системи;
- знання класифікації ігрових програм;
- здатність розробляти правила та проектувати ігрову механіку;
- здатність використовувати технології та інструменти для створення систем з елементами штучного інтелекту;
- здатність застосовувати об'єктно-орієнтований підхід до створення складних систем;
- здатність застосовувати основні програмні та технічні засоби розробки;
- здатність виконувати командну розробку ігрової програми;
- знання ігрових платформ, що існують окремо.

2 ОПИС МОДУЛЯ

Галузь знань: 12 «Інформаційні технології».

Спеціальність: 122 «Комп'ютерні науки та інформаційні технології»

Рівень: бакалавр.

Назва дисципліни: Технології створення програмних продуктів

Назва змістовного модуля: Основи комп'ютерних ігор та ігрових програм.

Семестр: 5

Кількість кредитних одиниць: дисципліна - 4,0, модуль - 1,0

Орієнтовна кількість годин: дисципліна - 120, модуль - 30

Викладач: к.т.н. Нікітін С.О.

3 ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ ТА РЕЗУЛЬТАТИ НАВЧАННЯ¹

СПЕЦІАЛЬНІ (ФАХОВІ) КОМПЕТЕНТНОСТІ

ПК-7	Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: структурного, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.
ПК-8	Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, сховища даних і бази знань, для забезпечення обчислювальних потреб багатьох користувачів, обробки транзакцій, у тому числі на хмарних сервісах.
ПКс8-3	Здатність до системного мислення, застосування методології системного ; аналізу для дослідження складних проблем різної природи, методів формалізації та розв'язанні системних задач, що мають суперечливі цілі, невизначеності та ризики.
ПКс8-4	Здатність застосовувати теоретичні та практичні основи методології та технології моделювання, реалізовувати алгоритми моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити експерименти за програмою моделювання з обробкою й аналізом результатів.
ПКс8-6	Здатність до інтелектуального багатовимірного аналізу даних оперативної аналітичної обробки з візуалізацією результатів процесі розв'язання прикладних задач в галузі комп'ютерних наук.

ПРОГРАМНІ РЕЗУЛЬТАТИ НАВЧАННЯ

ПРОФЕСІЙНА ПІДГОТОВКА

РН-7	Знання структур даних та фундаментальних алгоритмів, методології та інструментальних засобів об'єктно-орієнтованого аналізу та проектування, особливостей різних парадигм програмування, принципів, моделей, методів і технологій проектування і розроблення програмних продуктів різного призначення.
РН-8	Знання принципів, інструментальних засобів, мов веб-програмування, технологій створення баз даних, сховищ і вітрин даних та бази знань для розробки розподілених застосувань з інтеграцією баз і сховищ даних в архітектуру клієнт-сервер.
ПНс8-3	Знання методології системного аналізу для системного дослідження детермінованих та стохастичних моделей об'єктів і процесів, проектування та експлуатації інформаційних систем, продуктів, сервісів інформаційних технологій, інших об'єктів професійної діяльності.
ПНс8-4	Знання моделей систем масового обслуговування, мереж Петрі; методології ймовірнісного та імітаційного моделювання об'єктів, процесів і систем; планування та проведення експериментів з моделями, прийняття рішень щодо досягнення мети за результатами моделювання.
ПНс8-6	Знання методів та алгоритмів оперативної аналітичної обробки та інтелектуального аналізу даних для задач класифікації, прогнозування, кластерного аналізу, пошуку асоціативних правил з використанням програмних інструментів підтримки багатовимірного аналізу даних.

¹ Профіль програми освітнього ступенів бакалавра. Спеціальність 122 Комп'ютерні науки та інформаційні технології, спеціалізація 122-08 Системи штучного інтелекту

ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ ТА РЕЗУЛЬТАТИ НАВЧАННЯ МОДУЛЬ «МУЛЬТИАГЕНТНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ В ІГРОВИХ ДОДАТКАХ»

Спеціальні (фахові) компетентності	ПК-7, ПК-8, ПКс8-3, ПКс8-4, ПКс8-6
Професійна підготовка	РН-7, РН-8, ПНс8-3, ПНс8-4, ПНс8-6

4 МІЖДІСЦИПЛІНАРНІ ЗВ'ЯЗКИ

Для засвоєння матеріалу обрано наступні дисципліни:

- алгоритмізація та програмування;
- алгоритми та структура даних;
- об'єктно-орієнтоване програмування;
- технологія створення програмних продуктів;
- методи та системи штучного інтелекту.

5 МЕТА ТА ПЕРЕДБАЧУВАНІ РЕЗУЛЬТАТИ ВИВЧАННЯ МОДУЛЯ

5.1 МЕТА МОДУЛЯ

Мета модуля – формування у студентів системи знань про ігрові програмні системи, Отримання знань про ігрові правила та проектування ігрової механіки, застосування отриманих знань та практичних навичок у комп'ютерних ігрових додатках (КІД).

5.2 РЕЗУЛЬТАТИ НАВЧАННЯ

ЗНАННЯ ТА ЇХ ВИКОРИСТАННЯ

У разі успішного оволодіння матеріалами модуля студент буде вміти використовувати сучасні технології побудови КІД та їх компонентів, методи та алгоритми розробки ігрової механіки. Отримані знання дадуть студентам можливість будувати сучасні програмні ігрові системи.

ДОСЛІДНИЦЬКІ НАВИЧКИ

У разі успішного вивчення модуля студент буде вміти виконувати постановку завдань для побудови КІД, пропонувати варіанти їх вирішення, вибирати засоби розробки.

СПЕЦІАЛЬНІ ВМІННЯ

У разі успішного вивчення модуля студент буде вміти:

- визначати підходи та методи для створення інтелектуальних компонентів

комп'ютерних ігор;

- використовувати методи, моделі та алгоритми для створення програмних ігрових додатків;
- брати участь у командній розробці комп'ютерних ігрових додатків.

СОЦІАЛЬНІ ВМІННЯ

У разі успішного вивчення модуля студент буде вміти брати участь у командній роботі: обговорювати в групі шляхи побудови комп'ютерних ігрових додатків.

ОСОБИСТІ ЯКОСТІ

У разі успішного вивчення модуля студент буде вміти:

- самостійно вирішувати питання, що стосуються побудови та створення комп'ютерних ігрових додатків;
- аналізувати сучасну науково-технічну літературу з питань створення комп'ютерних ігрових додатків.

6 КАЛЕНДАРНИЙ ПЛАН СЕМЕСТРУ ТА СТРУКТУРА МОДУЛЯ

ІНФОРМАЦІЙНЕ НАПОВНЕННЯ ЗМІСТОВНОГО МОДУЛЯ

Номер	Номер тижня	Тема змістовного модуля
1	11	Вступ. Основні поняття та означення. Основні поняття, роль гри у розвитку особистості. Історія розвитку комп'ютерних ігор, підходи до класифікації КІД. Вивчення інтерфейсу Clickteam Fusion
2	12	Комп'ютерні ігри та ігрові професії. Історія розвитку комп'ютерних ігор, етапи розробки ігор та ігрові професії. Класифікація КІД та їх складові. Особливості ігрових професій. Розміщення фону і об'єктів в додатку Clickteam Fusion
3	13-14	Розроботка комп'ютерної гри. Етапами розробки комп'ютерної гри, ігрові ресурси, особливості ведення ігрової документації. Анімація персонажів у додатку Clickteam Fusion.
4	15	Менеджмент game-проектів. Види компаній, що розробляють КІД, характеристики та обов'язки спеціалістів команд з розробки ігор, характеристики методологій розробки game-проектів. Створення подій і дій у додатку Clickteam Fusion. Звукове оформлення і завершення гри в додатку Clickteam Fusion

7 ФОРМИ НАВЧАННЯ

Навчальне навантаження модуля складається з аудиторної та самостійної роботи. Аудиторна робота включає 15 лекцій та 5 лабораторних робіт. Самостійна робота студентів передбачає підготовку до аудиторних зайнять та лабораторних робіт, а також підготовку до тесту та оформлення залікового звіту з лабораторних робіт.

Підготовкою до поточкових аудиторних зайнять є аналіз літератури, інтернет-матеріалів по темах лекцій й лабораторних робіт, підготовка до тестів.

Контактні години передбачають індивідуальні консультації та контроль студентів в он-лайн режимі.

Заліковий звіт – опис та презентація за результатами виконання лабораторних робіт.

8 ПОРЯДОК ПРОВЕДЕННЯ АТЕСТАЦІЇ

Загальний принцип оцінювання підсумкових знань студентів полягає в опитуванні студентів на лекціях, оцінюванні поточної практичної роботи студентів у навчальному семестрі на лабораторних роботах та на заході контрольної оцінки у формі іспіту, у результаті яких студент отримує сумарну оцінку в балах. За результатами вивчення кожної теми змістовного модуля передбачено отримання оцінки за виконання залікового завдання. Оцінка включає: результати тестування студентів на лекціях, результати поточного опитування при виконанні лабораторних робіт модуля, оцінку за виконання залікового завдання. Сумарно оцінка кожного змістовного модуля не может бути більше, ніж 10 балів. Максимальна оцінка іспіту 40 балів.

ОЦІНКА РЕЗУЛЬТАТІВ ВИВЧЕННЯ ДИСЦИПЛІНИ В ЦІЛОМУ

Поточне тестування	Балів
Модуль 1	до 20
Модуль 2	до 30
Модуль 3	до 30
Модуль 4	до 20
разом	до 100

ГРАФІК ПРОВЕДЕННЯ ПОТОЧНОГО ОЦІНЮВАННЯ МОДУЛЯ

Номер тижня	Оцінювання
12	Оцінка виконання лабораторної роботи 1
13	Оцінка виконання лабораторної роботи 2
14	Оцінка виконання лабораторної роботи 3
15	Оцінка виконання лабораторної роботи 4

Номер тижня	Оцінювання
16	Оцінка виконання лабораторної роботи 5
16	Оцінка залікового звіту

ПРЕДСТАВЛЕННЯ ЗВІТУ ВІДНОСНО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

При представленні звіту з лабораторних робіт та захисту виконуються наступні дії. При умові виконання кожної окремої лабораторної роботи єдиний звіт надається студентом на п'ятнадцятому тижні семестру. Продовження терміну можливе лише при наявності поважної причини, передбаченої порядком навчання студентів у вищій школі. При цьому:

електронна версія єдиного звіту за результатами проходження модуля надається викладачеві через систему Інтернет на 12, 13, 14, 15, 16 тижні; за кожний день затримки представлення та здачі єдиного звіту з модуля знімається 1 бал.

МЕТОД ОЦІНКИ ЗМІСТОВОГО МОДУЛЯ

Кількість балів в загальній оцінці змістовного модуля така:

Тестування з теми	максимально 3 бала
Виконання лабораторної роботи 1	максимально 3 бала
Тестування з теми	максимально 3 бала
Виконання лабораторної роботи 2	максимально 3 бала
Тестування з теми	максимально 3 бала
Виконання лабораторної роботи 3	максимально 3 бала
Тестування з теми	максимально 3 бала
Виконання лабораторної роботи 4	максимально 3 бала
Тестування з теми	максимально 3 бала
Виконання лабораторної роботи 5	максимально 3 бала
Оцінка залікового звіту	максимально 30 балів

Усі набрані бали підсумовуються (максимально 30 балів), штрафні бали за запізнення в представленні єдиного звіту (максимально мінус 1 бал) віднімаються. Сумарна оцінка (від 0 до 30 балів) є індивідуальною оцінка студента засвоєння змістовного модуля.

МЕТОД ОЦІНКИ ДИСЦИПЛІНИ В ЦІЛОМУ

Оцінки студентів за результатами вивчення змістовних модулів 1 – 4 підсумовуються. Оцінка іспіту (максимально 40 балів) додається. Таким чином розраховується сумарна оцінка студента в балах за дисципліною.

Сумарна оцінка в балах переводиться за нижченаведеною шкалою оцінювання в національну та ЄКТС- оцінку:

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 – 100	A	Відмінно
82 – 89	B	Дуже добре
74 – 81	C	Добре
64 – 73	D	Задовільно
60 – 63	E	Посередньо
35 – 59	FX	Не зараховано з можливістю повторного складання
0 – 34	F	Не зараховано з обов'язковим повторним вивченням дисципліни

КРИТЕРІЇ ОЦІНЮВАННЯ ПРЕЗЕНТАЦІЇ РЕЗУЛЬТАТІВ РОБОТИ

	Високий	Середній		Низький
	4	3	2	1
організація	Організаційна структура (конкретне введення і висновок, послідовність викладу матеріалу всередині тіла і переходи) чітко і послідовно спостерігалася, вміло представлена і робить зміст презентації цілісним.	Організаційна структура (конкретне введення і висновок, послідовність викладу матеріалу всередині тіла і переходи) чітко і послідовно проглядається в презентації.	Організаційна структура (специфічне введення і висновок, послідовність викладу матеріалу всередині тіла і переходи) періодично проглядається в презентації.	Організаційна структура (специфічне введення і висновок, послідовність викладу матеріалу всередині тіла і переходи) не спостерігається в презентації.
Мова	Вибір мови є вражаючим, що запам'ятовується, є привабливим і підвищує ефективність презентації. Мова в презентації підходить аудиторії.	Вибір мови є продуманим і в цілому підтримує ефективність презентації. Мова в презентації підходить аудиторії.	Вибір мови є повсякденним і звичайним і частково підтримує ефективність презентації. Мова в презентації підходить аудиторії.	Вибір мови незрозумілий і мінімально підтримує ефективність презентації. Мова в презентації не підходить аудиторії.
Подача	Вміння подачі (поза, жест, зоровий контакт і голосова виразність) роблять презентацію переконливою, а доповідач виглядає підготовленим, впевненим.	Вміння подачі (поза, жест, зоровий контакт і голосова виразність) роблять презентацію цікавою, і динаміка доповідача виглядає впевнено.	Вміння подачі (поза, жест, зоровий контакт і голосова виразність) роблять презентацію зрозумілою, а доповідач виглядає не дуже впевнено.	Вміння подачі (поза, жест, зоровий контакт і голосова виразність) роблять презентацію незрозумілою, а доповідь виглядає невпевнено.

	Високий	Середній		Низький
	4	3	2	1
Допоміжні матеріали	Різноманітні типи допоміжних матеріалів (пояснення, приклади, ілюстрації, статистика, аналоги, цитати з відповідних джерел) дають посилання на інформацію або аналіз, які в значній мірі підтримують презентацію і підтверджують авторитет доповідача в представленій темі.	Допоміжні матеріали (пояснення, приклади, ілюстрації, статистика, аналоги, цитати з відповідних джерел) містять відповідні посилання на інформацію або аналіз, які в цілому підтримують презентацію і підтверджують авторитет доповідача в представленій темі.	Допоміжні матеріали (пояснення, приклади, ілюстрації, статистичні дані, аналоги, цитати з відповідних джерел) містять відповідні посилання на інформацію або аналіз, які частково підтримують презентацію і авторитет доповідача в представленій темі.	Недостатні допоміжні матеріали (пояснення, приклади, ілюстрації, статистика, аналоги, цитати з відповідних джерел) містять відповідні посилання на інформацію або аналіз, які мінімально підтримують презентацію і авторитет доповідача в представленій темі.
Головна думка	Головна думка є переконливою (точно сформульована, відповідним чином повторюється, запам'ятовується і міцно підтримується).	Головна думка є ясною і узгоджується з допоміжним матеріалом.	Головна думка є в принципі зрозумілою, але не часто повторюється і не запам'ятовується.	Головна думка може бути виведена, але явно не вказана в презентації.

9 ЗВОРОТНИЙ ЗВ'ЯЗОК

Інформація відносно результатів тестування, виконання лабораторних робіт та загальна оцінка змістовного модулю надається кожному студенту як індивідуально, так і групі в цілому. Інформація щодо результатів тестування надається студентам по завершенні тестування. Загальні результати надаються на шістнадцятому тижні навчання. Інформація щодо оцінки виконання лабораторної роботи надається студентіві під час заняття. Інформація щодо оцінки змістовного модуля в цілому надається студентам на 12, 13, 14, 15, 16 тижні навчання.

Контактні дані для on-line допомоги та консультування:

викладач: Никітін С.О., e-mail: sergnick.bk@gmail.com

Касілов О.В., e-mail: o.kasilov@hotmail.com

Никітіна Л.О., e-mail: nikitinalud@gmail.com

10 ВИКЛАДАЦЬКИЙ СКЛАД ТА ДОПОМІЖНІ ДЖЕРЕЛА

ОБОВ'ЯЗКИ ВИКЛАДАЧІВ

- подання матеріалів модуля згідно з програмою;
- оцінка результатів тестування та виконання лабораторних робіт.

ОБОВ'ЯЗКИ КООРДИНАТОРА ДИСЦИПЛІНИ

- планування та внесення змін до модуля;
- координація та керування професорсько-викладацьким складом;
- координація проведення тестування, лабораторних робіт та іспіту.
- обов'язки допоміжного персоналу

Допоміжний персонал Здійснює підготовку комп'ютерної техніки до виконання лабораторних робіт студентами та надає технічну підтримку студентам під час виконання лабораторних робіт.

КОНТАКТНІ ДАНІ ВИКЛАДАЧІВ

к.т.н. доцент, Нікітін С.О., e-mail: sergnick.bk@gmail.com

к.т.н., доцент, Нікітіна Л.О., e-mail: nikitinalud@gmail.com

КОНТАКТНІ ДАНІ КУРАТОРА

к.т.н., доцент, Касілов О.В., e-mail: o.kasilov@gmail.com

11 НАВЧАЛЬНА ПРОГРАМА І МАТЕРІАЛИ

11.1 ТЕМА 1: ВСТУП. ОСНОВНІ ПОНЯТТЯ ТА ОЗНАЧЕННЯ

АНОТАЦІЯ

Лекція знайомить з навчальною програмою, метою та завданнями дисципліни. Даються означення основних понять, визначається роль гри у розвитку особистості. Наводиться історія розвитку комп'ютерних ігор, підходи до класифікації КІД.

МЕТА ЛЕКЦІЇ

Ознайомити студентів з темами та змістом модуля, знаннями та вміннями, якими ними передбачено оволодіти в результаті його вивчення. Надати формулювання основних понять комп'ютерних ігор та КІД, визначити сутність інтелектуального КІД та інтелектуальної поведінки. Пояснити призначення та основи сфери застосування КІД.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

Студент має усвідомити призначення дисципліни та можливості застосування отриманих знань та вмінь при розробці КІД. Студент має ознайомитися з основними поняття КІД. Має отримати розуміння сфери застосування КІД, методів та моделей ігрових технологій.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Дайте визначення відомих вам понять КІД.
- Дайте визначення інтелектуальних компонентів КІД.
- Поясніть призначення та сферу застосування КІД.
- Наведіть приклади КІД.
- Назвіть та охарактеризуйте підходи до класифікації КІД.
- Дайте визначення інтелектуальної поведінки.

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)

11.2 ТЕМА 2: КОМП'ЮТЕРНІ ІГРИ ТА ІГРОВІ ПРОФЕСІЇ

АНОТАЦІЯ

Ознайомлення з історією розвитку комп'ютерних ігор, етапами розробки ігор та ігровими професіями. Лекція знайомить з класифікацією КІД та їх складовими. Пояснюються особливості ігрових професій.

МЕТА ЛЕКЦІЇ

Ознайомити студентів з основними особливостями КІД. Надати загальну характеристику підходів до класифікації КІД. Охарактеризувати основні професії розробників КІД.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

Студент має зрозуміти сутність поняття КІД. Повинен знати підходи до класифікації КІД та їх характеристики. Студент має вміти визначити приналежність КІД до певного класу.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Дайте визначення КІД.
- Дайте визначення основних класів КІД.
- Назвіть характеристики класів КІД.
- Назвіть та охарактеризуйте професії розробників КІД.
- Назвіть та охарактеризуйте основні ігрові платформи.
- Наведіть характеристики ігор для мобільних пристроїв.
- Назвіть основні етапи розвитку КІД.

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)

11.3 ТЕМА 3: РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ

АНОТАЦІЯ

Лекція знайомить з основними етапами розробки комп'ютерної гри, ігровими ресурсами, особливостями веденням ігрової документації.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

Студент має зрозуміти послідовність та зміст етапів розробки КІД. Ознайомитися з характеристиками основних ігрових ресурсів. Вивчити принципи розробки ігрової документації.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Дайте визначення ігрової платформи.
- Назвіть основні елементи комп'ютерної гри
- Охарактеризуйте ресурси типу «зображення».
- Охарактеризуйте ресурси типу «звуки».
- Охарактеризуйте етапи розробки комп'ютерної гри.
- Що представляє собою ігрова документація?
- Дайте визначення концепт-документа та поясніть його зміст.

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)

11.4 ТЕМА 4: МЕНЕДЖМЕНТ GAME-ПРОЕКТІВ

АНОТАЦІЯ

Лекція знайомить з видами компаній, що розробляють КІД, наводить характеристики та обов'язки спеціалістів команд з розробки ігор, характеристики методологій розробки game-проектів.

МЕТА ЛЕКЦІЇ

Ознайомити студентів з особливостями формування команд розробників КІД та організації їх роботи. Навести характеристики методологій розробки КІД.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

Сформувати у студентів знання основних підходів до організації роботи спеціалістів під час розробки КІД. Сформувати вміння вибрати відповідну методологію для розробки game-проектів.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Охарактеризуйте види компаній в ігровій індустрії.
- Назвіть та охарактеризуйте обов'язки основних спеціалістів команди розробників.
- Назвіть та охарактеризуйте обов'язки неосновних спеціалістів команди розробників.
- Поясніть, як визначається оптимальний склад команди розробників.
- Поясніть, як виконується масштабування складу команди розробників.
- Охарактеризуйте принципи управління командою розробників.
- Поясніть суть методології Agile.
- Поясніть суть методології Waterfall.
- Назвіть та охарактеризуйте основні методології у рамках Agile.

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)

11.5 ЛАБОРАТОРНА РОБОТА № 1. ЗНАЙОМСТВО З СЕРЕДОВИЩЕМ РОЗРОБКИ

АНОТАЦІЯ

Лабораторна робота орієнтована на Ознайомлення студентів з основними характеристиками середовища розробки Clickteam Fusion.

МЕТА ЛАБОРАТОРНОЇ РОБОТИ

- знайомство з інтерфейсом додатка Clickteam Fusion;
- вивчення основних можливостей Clickteam Fusion;
- створення першої міні-гри.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

Успішне виконання лабораторної роботи надає студенту вміння орієнтуватися в інтерфейсі середовища розробки Clickteam Fusion та використовувати редактор, можливість публікації на різних платформах.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Охарактеризуйте створення ігрових додатків на Clickteam Fusion.
- Поясніть можливості редактора гри.
- Охарактеризуйте можливості роботи з ресурсами.
- Як виконується підтримка імпортування файлів?
- Поясніть, що таке платформа та активний об'єкт.
- Назвіть властивості інструментів розробки.

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)

11.6 ЛАБОРАТОРНА РОБОТА №2.

РАЗМІЩЕННЯ ФОНУ ТА ОБ'ЄКТІВ У ДОДАТКАХ CLICKTEAM FUSION

АНОТАЦІЯ

Лабораторна робота орієнтована на оволодіння студентами навичок аналізу та подання концепції КІД.

МЕТА ЛАБОРАТОРНОЇ РОБОТИ

- вивчення прийомів розміщення та блокування фону в додатках Clickteam Fusion;
- вивчення основних можливостей розміщення об'єктів в ігровій області Clickteam Fusion;
- налаштування виду ігрового персонажу та інших ігрових об'єктів.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

У разі успішного виконання лабораторної роботи студент має вміти створити компоненти гри та налаштувати їх властивості.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Як можна описати КІД?
- Назвіть та охарактеризуйте можливі способи представлення КІД.
- За якими принципами формується назва гри?
- Як визначити ключові відмінності гри, що розробляється?
- За якими правилами вибирають платформу?
- За якими правилами вибирають цільову аудиторію?
- Чим характеризуються шари КІД?

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)

11.7 ЛАБОРАТОРНА РОБОТА №3. АНІМАЦІЯ ПЕРСОНАЖІВ У ДОДАТКАХ CLICKTEAM FUSION

АНОТАЦІЯ

Лабораторна робота орієнтована на оволодіння студентами навичок моделювання якісної ігрової сцени, шляхів вирішення проблем моделювання ігрової сцени.

МЕТА ЛАБОРАТОРНОЇ РОБОТИ

- вивчити можливості редактора анімації;
- дослідити варіанти руху активних об'єктів;
- виконати налаштування анімації ігрових персонажів.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

У разі успішного виконання лабораторної роботи студент має вміти виконати анімацію персонажів ігрової сцени у КІД.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Охарактеризуйте функціональні можливості редактора анімації.
- Охарактеризуйте інструменти редактора анімації.
- Які основні об'єкти можна додати на фрейм?
- Як виконати завантаження зовнішніх файлів для анімації об'єктів?
- Як можна налаштувати рух анімованих об'єктів?
- Як можна Імпортувати анімаційні файли?

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)

11.8 ЛАБОРАТОРНА РОБОТА №4. СВОРЕННЯ ПОДІЙ ТА ДІЙ В ДОДАТКИ CLICKTEAM FUSION

АНОТАЦІЯ

Лабораторна робота орієнтована на оволодіння студентами навичок розробки рівнів гри та керування поведінкою ігрових персонажів, пов'язаною з переміщенням об'єктів сцени у просторі.

МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Розробка програмного забезпечення для відображення набору статичних об'єктів (основна геометрія рівня) та набору динамічних (інтерактивних) ігрових об'єктів та розробка реакції об'єктів на події.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

У разі успішного виконання лабораторної роботи студент має вміти подати в ігровому додатку статичні та динамічні об'єкти.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Назвіть характеристики подій у КІД.
- Назвіть характеристики умов у КІД.
- Назвіть характеристики функцій у КІД.
- Опишіть процес створення ігрового рівня.
- Як виконується формування реакції об'єктів на події?
- Назвіть етапи обробки даних.
- Як виконується впорядкування дій у рамках події?
- Які інструменти використовують при редагуванні події?

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)

11.9 ЛАБОРАТОРНА РОБОТА №5.

ЗВУКОВЕ ОФОРМЛЕННЯ ТА ЗАВЕРШЕННЯ ГРИ У ДОДАТКУ SLICKTEAM FUSION

АНОТАЦІЯ

Лабораторна робота орієнтована на оволодіння студентами навичок застосування звукового оформлення та компіляції КІД, налаштування середовища розробки на різних платформах, збирання додатка та встановлення його на цільову платформу.

МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Розробка звукового оформлення та компіляція збирання розробки програмного забезпечення КІД на різних платформах.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

У разі успішного виконання лабораторної роботи студент має вміти розробити та застосувати звукове оформлення та здійснити компіляцію КІД на різних платформах, що задає порядок виконання методів та виклику зовнішніх модулів.

КОНТРОЛЬНІ ЗАПИТАННЯ

- Як створити та застосувати ресурси звуків у КІД?
- Що являє собою процес збирання КІД та який є його кінцевий результат?
- Які платформи підтримують середовище розробки?
- Назвіть особливості зборки КІД для мобільних платформ.
- Назвіть кроки встановлення КІД на мобільну платформу.
- Охарактеризуйте параметри налаштування зборки.

МЕТОДИЧНІ МАТЕРІАЛИ ТА ВКАЗІВКИ

[HTTP://BLOGS.KPI.KHARKOV.UA/V2/GAMEHUB/DOWNLOAD](http://blogs.kpi.kharkov.ua/v2/gamehub/download)



Лекції

12 СПИСОК ДЖЕРЕЛ ІНФОРМАЦІИ

1. A Quick Intro to Physics in ClickTeam Fusion 2.5 [Электронный ресурс]. — Mode access: <http://download.clickteam.com/tutorials/en/quick-physics.pdf> (reference date: 10.01.2017).
2. Ahearn L. Awesome Game Creation: No Programming Required / L. Ahearn, C. E. Crooks. — 2nd ed. — Game Development Series, 2002. — 450 p.
3. Champandard A. I. AI Game Development / A. I. Champandard. — New Riders Games, 2003. — 731 p.
4. Clickteam Fusion 2.5 tutorials – Game development for beginners [Electronic resource]. — Mode access: <http://library.clickteam.com/videos/clickteam-fusion-2.5-tutorials-beginners/> (reference date: 25.05.2017).
5. Donnelly R. Artificial Intelligence in Gaming [Electronic resource] / Ryan Donnelly. — Mode access: <https://ru.scribd.com/document/92736987/Ryan-Donnelly-Artificial-Intelligence-in-Gaming> (reference date: 21.01.2017).
6. Free ClickTeam Fusion 2.5 tutorials [Electronic resource]. — Mode access: <https://clickfusion.academy/free-clickteam-fusion-2-5-tutorials/> (reference date: 10.01.2017).
7. Fusion 2.5 Game Engine Tutorial, part 2 [Electronic resource]. — Mode access <https://www.youtube.com/watch?v=ePX5YC78A2g> (reference date: 10.01.2017).
8. Novak J. Game Development Essentials: An Introduction / J. Novak. — 3rd ed. — Delmar Cengage Learning, 2011. — 483 p.

ЛЕКЦІЯ 1 РОЗРОБКА КОМП'ЮТЕРНИХ ІГОР

Мета лекції: Ознайомлення з історією розвитку комп'ютерних ігор, етапами розробки ігор та ігровими професіями

1.1 ІСТОРІЯ РОЗВИТКУ КОМП'ЮТЕРНИХ ІГОР

Історія комп'ютерних ігор починається у п'ятдесятих роках минулого століття. Тоді з'явилися перші пристрої, які пізніше перетворилися в ігрові приставки, портативні ігрові консолі та персональні комп'ютери, які сьогодні є найстарішою ігровою платформою.

Термін «відеоігри» (video games) сьогодні використовується для позначення всіх видів ігор, взаємодію з якими організовано за допомогою різних пристроїв для відображення відеоінформації. У цю категорію потрапляють комп'ютерні ігри, ігри для ігрових приставок та ігри для мобільних пристроїв.

Етапи розвитку ігрової індустрії і характеристики деяких комп'ютерних ігор наведено в таблиці нижче.

Етап	Назва, рік виходу	Коментарі
1940-і роки	Ракетний симулятор 1947 р	Гра реалізована за допомогою катодного-променевої трубки. Гравець керував світловою плямою–ракетою для ураження цілі. Ціль було намальовано і прикріплено до поверхні екрану.
1950-ті роки	OXO 1952 р	Комп'ютерний варіант гри в хрестики-нулики. Програма містила модуль штучного інтелекту, що дозволяв комп'ютеру грати з людиною.
	Tennis for Two 1958 р	Простий симулятор гри в теніс. Перша відеогра. Для управління використовувалися контролери, устатковані джойстиком та кнопкою. Пристрій відображення інформації – осцилограф.
1960-і роки	Spacewar! 1961 р	Гра працювала на комп'ютері PDP-1. Мета — вразити космічний корабель супротивника і при цьому не потрапити під його вогонь і не зіткнутися із зіркою.
	Chase 1966 р	Вперше використовувала у ролі пристрою для виведення інформації звичайний телевізор.
	Space Travel 1969 р	Гра для ОС MULTICS. Пізніше гру було перенесено на інші ОС, в тому числі на ОС Unix.
1970-ті роки	Galaxy 1971 р	Засновано на концепції Spacewar! і реалізовано для першого ігрового автомата.
	Computer Space 1971 р	Ігровий автомат, що заснований на Spacewar!.

Етап	Назва, рік виходу	Коментарі
	Baseball 1971 р	Перша комп'ютерна реалізація бейсболу на комп'ютері DEC PDP-10.
	Star Trek 1971 р	Класична комп'ютерна гра на мові BASIC з текстовим інтерфейсом користувача.
	Atari PONG 1972 р	Продукт компанії Atari з ігровим пристроєм. Аркадна відеогра. Тенісна спортивна гра з використанням простої двомірної графіки.
	Maze War і Spasim 1974 р	Перші спроби створення багатокористувацьких тримірних FPS.
	Airflight 1974 р	Авіасимулятор. Стала прототипом для Microsoft Flight Simulator.
	Adventure 1975 р	Комп'ютерна гра з текстовим інтерфейсом в жанрі квест. Стала прототипом багатокористувацьких текстових ігор MUD-ігор.
	Space Invaders 1978 р	Продукт компанії Taito для аркадних ігрових автоматів. Жанр – «стрілялки». Мета – знищення зі звуковими ефектами рухомих рядів інопланетян за допомогою безлічі патронів.
	Pac Man 1979 р	Продукт компанії Namco. Жанр – «аркада». Кругла жовта істота має з'їсти все білі точки, уникаючи зіткнень з привидами.
1980-і роки	тетріс 1985 р	Укладання падаючих фігур в стакан без пустот.
	Final Fantasy 1987 р	Гра для ігрової консолі в жанрі JRPG (рольових ігор). Гравець створює команду героїв, які повинні знищити стихійних ворогів.
	1989 р	Кишенькова ігрова консоль Game Boy від компанії Nintendo з іграми Pokemon, Tetris та ін.
1990-і роки	The Lost Vikings 1992 р	Аркада від Blizzard Entertainment (Silicon & Synapse). Вибір трьох персонажів, викрадених інопланетянами, які повинні пройти 37 рівнів.
	Civilization I 1991 р. і Civilization II 1996 р.	«Цивілізаційна» стратегія, одна з найбільш довготривалих франшиз в ігровій індустрії.
	Dune II 1992 р	Жанр – стратегія в реальному часі. Продукт компанії Weswood Studios. Перша гра даного жанру для ПК.
	Doom 1993р. і Doom II 1994 р	Жанр – шутер від першої особи. Продукт компанії idSoftware. Гравець виступає в ролі бійця без умені в складі космічного спецназу, який повинен дослідити рівні різної складності, розгадати просторові і (рідко) логічні завдання, знайти схованки.

Етап	Назва, рік виходу	Коментарі
	Duke Nukem 3D 1996 р	Шутер від першої особи. Спочатку створено компанією 3D Realms / Apogee Software як серія відео-ігор для PC. Гравець бореться з прибульцями і запобігає інопланетного вторгнення.
	Quake 1996 р	Жанр – шутер від першої особи. Продукт компанії idSoftware. Солдат повинен очолити операцію відплати «Контрудар» і знищити ворога Quake.
	Counter Strike 1999 р	Серія комп'ютерних ігор в жанрі командного шутера від першої особи, засновано на движку GoldSrc. Ідея гри – протистояння двох команд – терористів і контр-терористів. Мета гри – виконати завдання карти або знищити сусупротивників.
	Star Craft 1998 р	Жанр – стратегія в реальному часі. Продукт компанії Blizzard Entertainment. Конфлікт трьох ворожих рас рівних силами. Кожна сторона відрізняється не тільки зовнішнім виглядом, але й характеристиками своїх представників, а також загальною стратегією бою.

В наші дні ігри продовжують розвиватися. Особлива увага сьогодні приділяється розробці нових ігрових інтерфейсів (таких, як Microsoft Kinect, Nintendo Wii U) і мобільних ігор.

Показовим є те, що Kinect для Xbox 360 потрапив в Книгу рекордів Гіннеса, як споживчий електронний пристрій, що швидко продається. У перші 60 днів після початку продажів Kinect було продано 8 мільйонів екземплярів. Існує чимало ігор, сумісних з Kinect, але потенціал цього сенсора ще належить розкрити. Цілком можливо, що саме ви станете одним з винахідників світу ігор нового покоління.

У середовищі ігрових консолей так само спостерігається поживлення – в 2011 побачили світ Nintendo 3DS і Sony PlayStation Vita.

Традиційні ігри – для стаціонарних ігрових консолей і ПК, зі звичним управлінням, звичайно ж, не залишаються без уваги. Але на тлі нових технологічних рішень вони виглядають досить спокійно, не привертають до себе підвищеної уваги.

На сьогоднішній день створення комп'ютерних ігор – це величезна індустрія. Бюджети ігрових проектів досягають десятків мільйонів доларів, а обсяг ринків сучасних ігор – комп'ютерних, консольних, мобільних, – оцінюється десятками мільярдів доларів.

ІГРИ ДЛЯ МОБІЛЬНИХ ТЕЛЕФОНІВ

За інформацією Ericsson, до четвертого кварталу 2011 року в світі було близько 6 мільярдів абонентів стільникового зв'язку. Навіть з огляду на те, що багато хто з них підключені до декількох операторів, в результаті виходить, що близько 60% населення Землі користуються мобільними телефонами. І кількість ця щороку зростає. На цьому

тлі не є дивними оптимістичні прогнози Gartner, відповідно до яких до 2014 року ринок мобільних ігор становитиме близько 11 мільярдів доларів.

Мобільний телефон спочатку було створено, як пристрій зв'язку, але до цієї його функції поступово додавалися інші, серед яких – і мобільні ігри.

Перші мобільні ігри з'явилися в середині 1990-х років. З огляду на графічні та обчислювальні можливості телефонів тих часів, ігри ці були досить прості. Наприклад – всім відомі Змійка (Snake), Тетріс, Лабіринт та подібні.

Першим мобільним телефоном, у який було вбудовано гру, прийнято вважати, Nokia 6110, випущений в 1998 році.

До цього ігри в телефони вбудовували (наприклад, тетріс в Siemens S1), але ця функціональність ставилася ледве не до розряду недокументованих можливостей. В ті часи стільниковий був занадто серйозним пристроєм і реакція ринку на подібне нововведення була непередбачуваною. Як виявилось, ігри припали ринку до смаку – і з кінця 1990-х практично всі телефони мали на борту кілька розважальних програм. Із зростанням можливостей пристроїв росла і якість ігор.

Перші ігри було вбудовано в прошивку телефону. Згодом, а саме з 2001 року, з появою телефонів, що підтримують технологію J2ME, ігри перетворилися на окремі програми, користувач мав змогу встановлювати їх на мобільний телефон. Ігри можна було купувати і завантажувати в телефон, використовуючи інтернет-з'єднання (поширення ігор відбувалося за допомогою WAP-сайтів). Ще через деякий час з'явилася можливість закачувати дистрибутиви ігор на комп'ютер, підключати телефон до комп'ютера за допомогою дата-кабелю, і, використовуючи спеціальне програмне забезпечення, завантажувати ігри на телефон. Крім мобільної версії Java поширення набули й інші платформи, що підходять для створення ігор. Ігровий ринок неухильно зростає і розвивався.

Сьогодні серед мобільних ігрових платформ виділяються Android, iOS і Windows Phone 7, розробкам ігор для яких присвячений наш курс.

Сьогоднішні мобільні телефони відрізняють серйозні обчислювальні можливості, в тому числі – апаратна підтримка тривимірної графіки, великої роздільної здатності дисплеїв, наявність сенсорних екранів (як правило – з підтримкою технологій розпізнавання множинних дотиків), наявність датчиків руху та інших технічних засобів. Не можна забувати і про те, що сучасний мобільний телефон – це пристрій, орієнтований, окрім іншого, на постійне з'єднання з Інтернетом – або за допомогою 3G-мережі, або через Wi-Fi. Крім того, всі згадані платформи передбачають комерціалізацію додатків, поширення їх через спеціальні сервіси. Все це відкриває величезне поле діяльності для розробників мобільного ПЗ, і зокрема мобільних ігор.

Можна помітити, що розробкою комп'ютерних ігор на самому початку їх виникнення займалися одинаки. Пізніше, на початку 80-х років, почали виникати компанії-розробники і компанії-видавці ігор. У 1980-90-х роках помітні ігрові проекти зазвичай створювалися працею великих компаній. Однак останні кілька років роль невеликих груп розробників і навіть одинаків зростає. Цей висновок можна зробити, наприклад,

якщо проаналізувати такий феномен, як аматорські модифікації (або, як прийнято казати, моди) популярних ігор. Наприклад, популярна гра Counter Strike з'явилася у вигляді аматорського мода до Half Life.

1.2 КЛАСИФІКАЦІЯ КОМП'ЮТЕРНИХ ІГОР

1 Класифікація за жанрами

1.1 Action

1.1.1 3D-шутери, «дії-стрілялки»

1.1.1.1 Шутери від першої та від третьої особи

1.1.1.2 «Криваві / М'ясисті» шутери

1.1.1.3 Тактичні шутери

1.1.2 Стрілялки

1.1.2.1 Побий їх всіх

1.1.2.2 Слешер

1.1.3 Аркада

1.1.4 Стелс-екшен

1.2 Симулятор / Менеджери

1.2.1 Технічні (реалістичні)

1.2.2 Аркадні

1.2.3 Спортивні симулятори

1.2.4 Спортивні менеджери

1.2.5 Економічні

1.3 Стратегії

1.3.1 Стратегії за схемою ігрового процесу

1.3.1.1 Стратегії в реальному часі

1.3.1.2 Покрокові стратегії

1.3.1.3 Карткові стратегії

1.3.2 Стратегії за масштабом ігрового процесу

1.3.2.1 War-games

1.3.2.2 Глобальні стратегії

1.3.2.3 Симулятори бога

1.4 Пригоди

1.4.1 Текстова пригодницька гра

1.4.2 Графічний квест

1.4.3 Пазли

1.4.4 Пригодницький бойовик

1.4.5 Симулятор побачень

1.4.6 Візуальна новела

1.5 Музичні гри

1.5.1 Ритмічні ігри

1.6 Рольові ігри

1.6.1 Тактичні RPG

1.7 Логічні

1.8 Традиційні та настільні

1.9 Текстові

1.9.1 Ігри в псевдографіці

2 Класифікація за кількістю гравців

2.1 Поодинокі

2.2 Стратегії

2.3 Стратегії на одному комп'ютері

2.4 Стратегії офлайн-гри

2.5 Масові онлайнкові

3 Класифікація за стилістикою

4 Класифікація по платформах

ACTION

3D-ШУТЕРИ, «ДІЇ-СТРІЛЯЛКИ»

3D-шутер (3D Shooter) – в іграх даного типу гравець, як правило, діючи самотужки, повинен знищувати ворогів за допомогою зброї ближнього бою (як правило холодної) і стрілецької зброї (найчастіше вогнепальної зброї та енергетичної) для досягнення певних цілей на даному рівні. Зазвичай після досягнення заданих цілей гравець переходить на наступний рівень. Ворогами часто є: бандити (напр. Max Payne), нацисти (напр. В Return to Castle Wolfenstein) та інші «погані хлопці», а також всілякі інопланетяни, мутанти і монстри (напр. Doom, Half-life, Duke Nukem 3D, Serious Sam).

ШУТЕРИ ВІД ПЕРШОЇ І ВІД ТРЕТЬОЇ ОСОБИ

У шутерах від першої особи (First person shooter, FPS) гравець не бачить персонажа з боку – він спостерігає за тим, що відбувається, від імені персонажа – «очима персонажа» (First person look), і картина, що спостерігається гравцем збігається з тим, що «бачить» персонаж. У шутерах від третьої особи (Third person shooter, TPS) гравець бачить персонажа з боку з фіксованою (зазвичай зі спини) або довільної точки огляду (англ. Third person look). У ряді ігор реалізовано можливість перемикання першої / третьої особи та фіксована / довільна камера.

Приклади шутерів від першої особи: Doom, Quake і ін.

Приклади шутерів від третьої особи: Tomb Raider, Max Payne і ін.

«КРИВАВІ / М'ЯСИСТІ» ШУТЕРИ

Суть полягає в знищенні величезної кількості ворогів, найчастіше комп'ютерних ботів, що насуваються великими групами на гравця. При цьому гравець повинен мати простір для маневру.

Приклади: Serious Sam, Painkiller і ін.

ТАКТИЧНІ ШУТЕРИ

Принципова відмінність тактичних шутерів від класичних полягає в тому, що персонаж не зображує героя-одинака, а діє у складі команди. В тактичному шутері зазвичай відтворюється діяльність загонів – взаємодія між бійцями, маневрування і вибір напрямку атаки, підбір команди та її озброєння. В одиночному режимі ці можливості реалізуються ботами, в мережевому – через взаємодію живих гравців.

Приклади: Battlefield, Counter-Strike і ін.

ФАЙТИНГ

Геймплей складається виключно з поєдинків двох або більше сусупротивників із застосуванням рукопашного бою.

Приклади: Mortal Kombat, Street Fighter і ін.

ПОБИЙ ЇХ ВСІХ

Різновид файтингів, бій в яких відбувається за межами арени і з безліччю сусупротивників водночас.

Приклади: Oni, Enter the Matrix і ін.

СЛЕШЕР

Ігри з видом від третьої особи; основною частиною ігрового процесу в них є фехтувальні поєдинки із застосуванням холодної та іншої зброї.

Приклади: God of War, Devil May Cry і ін.

АРКАДА

Жанр, який характеризується коротким за часом, але інтенсивним ігровим процесом. Кажучи точніше, аркадною вважається гра для аркадних ігрових автоматів.

СТЕЛС-ЕКШЕН

Ігри, в яких метою є не битися з більшістю зустрінутих супротивників, а всіляко уникати можливого контакту з ними, водночас виконуючи поставлені завдання. Стелс-елементи (наприклад, можливість виглядати з-за рогу, притулившись до стіни) часто зустрічаються в іграх різних жанрів.

Приклади: Metal Gear, Tom Clancy's Splinter Cell і ін.

СИМУЛЯТОР / МЕНЕДЖЕРИ

Ігри, що надають можливість симуляції і управління тим чи іншим процесом з реального життя.

ТЕХНІЧНІ (РЕАЛІСТИЧНІ)

За допомогою комп'ютера якомога повніше імітується фізична поведінка і управління будь-яким складним об'єктом технічної системи (наприклад: бойовим винищувачем, автомобілем і т. д.). Якщо аркадні симулятори прагнуть розважити гравця за допомогою різних неможливих явищ, трюків і гостроти сюжету, то головний критерій якості

технічних симуляторів – повнота та реалістичність моделювання об'єкта (автомобіля, повітряного судна і т. ін.).

Приклади: Автосимулятори Gran Turismo, Live for Speed і ін; авіасимулятори Microsoft Flight Simulator, War Thunder і ін; космічні пригоди Orbiter; залізничні симулятори Railworks, Trainz; симулятор водного транспорту Ship Simulator.

АРКАДНІ

Спрощена версія технічних симуляторів, нерідко з альтернативною фізикою. Принципова відмінність від власне аркад – наявність хоч і спрощеної, але все ж фізичної моделі. Найчастіше з подібною фізикою робляться симулятори космічних кораблів та автомобілів.

Приклади: Wing Commander, Need for Speed і ін.

СПОРТИВНІ СИМУЛЯТОРИ

Імітація будь-якої спортивної гри. Найбільшого поширення набули імітації футболу, хокею, баскетболу, та ін.

СПОРТИВНІ МЕНЕДЖЕРИ

Поєднують у собі елементи спортивного та економічного симулятора. Відмінною рисою є те, що в спортивному симуляторі гравець спостерігає безпосередньо за ігровим процесом і може впливати на хід матчу прямо під час його проведення, а в менеджері налаштування тактики, стратегії трансферів і фінансових операцій вибираються заздалегідь, і гравець переглядає результати вже після матчу.

У спортивному менеджері гравець виступає в ролі керуючого власною спортивною командою (спортсменом). Завданням гравця стає не тільки перемога в матчах, але й грамотне і успішне управління інфраструктурою свого клубу.

Приклади: FIFA Manager, Football Manager

ЕКОНОМІЧНІ

Економічні ігри нерідко зараховують також до жанру стратегій, присвячених відображенню економічних, ринкових процесів – частіше за все мова йде про підприємництво; метою гравця, що керує якимось підприємством, є отримання віртуального прибутку. У «чистих» економічних симуляторах відсутні елементи будівництва; гравець повинен керувати комерційним підприємством, що вже існує; ринкові процеси і поведінка конкурентів наближені до реальності.

Приклади: Монополія, Віртономіка.

Елементи економічної симуляції, досить значні, включено в багато стратегічних ігор, де однією з цілей гравця також є отримання прибутку.

СТРАТЕГІЇ

Стратегії – ігри, що вимагають планування та вироблення певної стратегії для досягнення якоїсь конкретної мети – наприклад, перемоги у військовій операції. Гравець управляє не одним персонажем, а цілим підрозділом, підприємством або навіть всесвітом. Розрізняють походові або покрокові стратегічні ігри (Turn-Based

Strategy, TBS), де гравці по черзі роблять ходи, і кожному гравцеві відводиться необмежений або обмежений (в залежності від типу і складності гри) час на свій хід, і стратегічні ігри в реальному часі (Real Time Strategy, RTS), в яких всі гравці виконують свої дії одночасно, і плин часу не переривається.

СТРАТЕГІЇ ПО СХЕМІ ІГРОВОГО ПРОЦЕСУ

СТРАТЕГІЇ В РЕАЛЬНОМУ ЧАСІ

В цих стратегіях в реальному часі (Real-time strategy, RTS) гравці проводять свої дії одночасно. RTS з'явилися дещо пізніше покрокових, першою грою цього жанру, що стала популярною, була *Dune II* (1992), сюжет якої заснований на однойменному творі Френка Герберта. Вже тоді сформувалися загальні принципи стратегій в реальному часі.

Більшість «класичних» стратегій в реальному часі припускають наступний ігровий процес: збір деяких ресурсів; будівництво та зміцнення бази або табору; створення на цій базі бойових одиниць (наймання солдатів, будівництво техніки); об'єднання їх у групи, штурм і знищення цими групами ворожої бази.

Приклади: Command & Conquer, Warcraft і ін.

ПОКРОКОВІ СТРАТЕГІЇ

Покрокові стратегії (Turn-based strategy, TBS) – ігри, в яких гравці роблять свої дії по черзі. Покрокові стратегії з'явилися раніше RTS і значно відрізняються від них. Поділ ігрового процесу на ходи відриває його від реального життя і позбавляє гру динамізму, в результаті чого ці ігри не так і популярні, як стратегії в реальному часі. З іншого боку, в TBS у гравця набагато більше часу на роздуми, під час здійснення ходу його ніщо не квапить, що дає можливість набагато більш глибокого і ґрунтовного планування.

Приклади: X-COM, Civilization і ін.

КАРТКОВІ СТРАТЕГІЇ

Карткові стратегії – це комп'ютерні версії настільних колекційних карткових ігор.

Приклади: Magic The Gathering, Спектрмансер і ін.

СТРАТЕГІЇ ЗА МАСШТАБОМ ІГРОВОГО ПРОЦЕСУ

ВАРГЕЙМИ (WAR-GAMES)

У Варгеймі, на відміну від інших видів стратегій, гравець повинен не створювати армію, його мета – перемогти сусупротивника в бою, використовуючи ті сили, які у нього є в розпорядженні на початок бою. У варгеймах, як правило, робиться наголос на автентичності, реалістичності і історичності.

Приклади: Steel Panthers, Panzer General та ін.

ГЛОБАЛЬНІ СТРАТЕГІЇ

Глобальні стратегії – стратегії, де гравець управляє державою. В його руках не тільки війна

та економіка, але й науковий прогрес, освоєння нових земель і дипломатія. У деяких з них поряд з глобальною картою існують місцеві, на яких проходять тактичні битви.

Приклади: Master of Orion, Civilization і ін.

СИМУЛЯТОР БОГА

Симулятор бога – стратегічні ігри, в яких гравцеві доводиться виступати в ролі «бога» – такої собі надприродної сутності, що піклується про цілий невеликий народ. Подібні ігри характеризує, як правило, непрямий контроль над окремими ігровими персонажами – ними керує комп'ютер, а роль гравця визначається «надприродним» втручанням в їхнє життя, будівництвом будівель, підтримкою оптимального стану опікуваного суспільства і тому подібне. Багато симуляторів бога не ставлять перед гравцем ніяких конкретних завдань, надаючи йому можливість вільно і необмежено розвивати опікуване суспільство.

Приклади: Black & White, Dungeon Keeper і ін.

ПРИГОДИ

Пригода (Adventure) – гра-розповідь, в якій керований гравцем герой просувається по сюжету і взаємодіє з ігровим світом за допомогою застосування предметів, спілкування з іншими персонажами та вирішення логічних задач.

ТЕКСТОВА ПРИГОДНИЦЬКА ГРА

Спочатку, через мале поширення графічних пристроїв відображення і нестачу ресурсів (пам'яті і потужності процесора), пригодницькі ігри були текстовими. Пізніше цей жанр було названо текстовим квестом. Відмінність від графічних квестів полягає в тому, що гравець взаємодіє з ігровим світом за допомогою командного рядка, а інформація про світ виводиться у вигляді текстів і малюнків з друкованих знаків. У текстових квестах зовсім не обов'язково відсутня графічна складова – це яскраво виражено в більш пізніх текстових квестах (Superhero league of Hoboken і т. п.)

ГРАФІЧНИЙ КВЕСТ

Графічні квести (пригодницькі ігри) еволюціонували з текстових пригод. Перші графічні квести з'явилися ще на початку 1980-х. Це такі популярні серії, як King's Quest, Space Quest, Leisure Suit Larry. Перші ігри цього жанру поєднували в собі текстове введення з графічним відображенням того, що відбувається. З виходом гри Maniac Mansion набув поширення інтерфейс, що використовує замість текстового введення управління за допомогою комп'ютерної миші («point and click»).

ГОЛОВОЛОМКИ

Окрім збирання предметів та їх використання, в цих іграх вирішуються різні головоломки, в тій чи іншій мірі інтегровані в сюжет, і на вирішення головоломок робиться основний упор. Зазвичай може знадобитися збірка різних, нерідко абсурдних як за виглядом, так і за функціональністю, механізмів. Найпопулярнішими представниками жанру є Myst, Neverhood.

ПРИГОДНИЦЬКИЙ БОЙОВИК

Головним чином засновано на реакції та на рефlekсах гравця, але є й елементи класичних квестів – предмети і взаємодія з навколишнім оточенням. Відомими представниками є ігри серії Legend of Zelda, Tomb Raider, Last of Us і ін.

СИМУЛЯТОР ПОБАЧЕНЬ

Симулятор романтичних відносин. Так само відомі як романтичні пригоди (romantic adventure). За діями (ходом ігри) деякі з них близькі до RPG, інші – до квестів.

ВІЗУАЛЬНА НОВЕЛА

Інша назва – візуальний роман. Жанр комп'ютерних ігор, в якому глядачеві демонструється історія за допомогою виведення на екран тексту, статичних (або анімованих) зображень, а також звукового та/або музичного супроводу.

МУЗИЧНІ ГРИ

У музичних іграх геймплей будується на взаємодії гравця з музикою – жанр її може бути будь-яким, від головоломок до ритм-ігор.

Приклади: SingStar, Mad Maestro! Та ін.

РИТМІЧНІ ІГРИ

Піджанр музичних ігор, який останнім часом набув великої популярності завдяки аркадним автоматам і спеціальним контролерам, що часто використовуються в подібних іграх. Через брак перших і дорожнечу других жанр поки не дуже поширений в Україні. Основною ідеєю є правильне натискання кнопок, що відображаються на екрані під ритм музики.

Приклади: osu !, Guitar Hero і ін.

РОЛЬОВІ ІГРИ

Характерні ознаки рольової гри (RPG – Role playing game):

- у головного героя (героїв), і інших персонажів і ворогів (частіше в меншій мірі) присутня деяка кількість параметрів (умінь, характеристик, навичок), які визначають їх силу та здібності. Зазвичай головна характеристика персонажів і ворогів – це рівень, що визначає загальну силу персонажа і визначає доступні навички і предмети екіпірування. Всі ці параметри треба вдосконалювати шляхом вбивства інших персонажів і ворогів, виконанням завдань і використанням цих самих навичок;
- присутній пророблений і великий світ, сильна сюжетна лінія, розгалужені діалоги з різними варіантами відповідей, безліч різних персонажів зі своїми цілями і характерами;
- велика кількість різних предметів: одяг, зброя, зілля, артефакти і т. ін.

ПІДЖАНРИ

- **action RPG** – рольова гра з шутерною або слешерною бойовою системою, включає до себе складні квести і розвинену систему прокачування.

Приклади: *Mass Effect, Deus Ex, Gothic, The Elder Scrolls*.

- **Hack'n'Slash RPG** – рольова гра з акцентом на винищення ворогів, збирання найбільш потужних речей, прокачування характеристик. Зазвичай має спрощений світ і сюжет, невелику кількість діалогів і варіантів вирішення квестів та інших завдань.

Приклади: *Diablo, Dungeon Siege, Titan Quest*.

- **true RPG** (Також просто RPG або CRPG) – рольова гра з великою кількістю діалогів, глибокою рольовою системою, свободою у виборі шляхів вирішення різних завдань, з проробленим світом і сюжетом.

Приклади: *Fallout, Planescape: Torment, Baldur's Gate, Dragon Age: Origins*.

- **JRPG** (Японська РПГ) рольова гра з проробленим світом і діалогами, але з меншою свободою вибору, найчастіше має дуже захоплюючий, але лінійний сюжет, відсутність вибору в прокачуванні характеристик персонажа, а також дуже опрацьованих і красивих персонажів. Гра чимось схожа на інтерактивну книгу. Найчастіше створюється японськими розробниками, – і існує дуже мало ігор такого роду, створених за межами Японії.

Приклади: *Final Fantasy, Dragon Quest*.

ТАКТИЧНІ RPG

Жанр рольових ігор, який є сумішшю з покроковою стратегією. Гравець управляє невеликою групою воїнів, хоча в деяких тактичних RPG їх число може доходити до кількох десятків. Перші тактичні RPG з'явилися на консолях в Японії, однак сьогодні існує багато західних тактичних рольових ігор.

Приклади: *Knights of the Old Republic, Fallout Tactics* і ін.

ГОЛОВОЛОМКИ, ЛОГІЧНІ, ПАЗЛИ

Пазли (Puzzle) та інші. У некомп'ютерній головоломці роль арбітра, що стежить за дотриманням правил, виконує або сам гравець (пасьянс), або деякий механічний пристрій (кубик Рубика). З появою комп'ютерів можливості головоломок розширилися, оскільки написати комп'ютерну програму простіше, ніж сконструювати механічний пристрій.

Головоломки, як правило, не вимагають реакції від гравця (проте багато хто веде відлік часу, витраченого на вирішення).

Приклади: *Сапер, Sokoban* і ін.

ТРАДИЦІЙНІ ТА НАСТІЛЬНІ

Комп'ютерна реалізація настільних ігор (Board), наприклад шахи, карти, шашки, «Монополія», серія ігор Warhammer.

Приклади: *CGoban* – інтернет-клієнт для азійської настільної гри ГО.

ТЕКСТОВІ

Interactive fiction – жанр комп'ютерних ігор, в якому спілкування з гравцем здійснюється за допомогою текстової інформації. Розвиток цього жанру, в зв'язку з низькою вимогою до ресурсів, почався досить давно, і не припинявся навіть з появою графічних ігор.

Існують два види інтерфейсу – інтерфейс, в якому текст вводиться з клавіатури або інтерфейс у вигляді меню, де гравець вибирає дію з декількох запропонованих (CYOA, Choose Your Own Adventure).

ІГРИ В ПСЕВДОГРАФІЦІ

Різновид текстових ігор, в яких є графічна картинка у вигляді псевдографіки – мозаїки, побудованої з ASCII-символів.

Приклади: Roguelike

КЛАСИФІКАЦІЯ ЗА КІЛЬКІСТЮ ГРАВЦІВ

Одиночні (singleplayer) – такі, що розраховані на гру самотужки, проти комп'ютера, та такі, що розраховані на багато користувачів (multiplayer) є ігри, розраховані на кількох людей (зазвичай до 64) у локальній мережі або в Інтернеті. Часто зустрічаються як додатковий режим в іграх, що мають в основі одиночне проходження.

РОЗРАХОВАНІ НА БАГАТО КОРИСТУВАЧІВ НА ОДНОМУ КОМП'ЮТЕРІ – (*HOTSEAT, SPLITSCREEN*)

На сучасних персональних комп'ютерах зустрічаються рідко, проте ще мають місце на старих ПК і приставках. Hotseat – гравці по черзі роблять необхідні ігрові дії, використовуючи один і той же комп'ютер / приставку. Splitscreen – екран поділяється на частини (частіше дві, рідше чотири), гравцям вже не потрібно чекати своєї черги, дія гри розгортається в реальному часі, кожен гравець взаємодіє з ігровим світом у своїй частині екрана.

РОЗРАХОВАНІ НА БАГАТО КОРИСТУВАЧІВ ОФЛАЙН-ІГРИ

Деякі жанри (спортивні менеджери, покрокові стратегії і т. д.) можуть працювати в такому форматі: гравці роблять ходи і відправляють результат через веб або електронну пошту. Незалежно від методу зв'язку (Фідонет, електронна пошта, веб ...), ці ігри:

вимагають періодичної появи в мережі, але при цьому потрібна лише періодичність, час підключення вибирає сам гравець, не погоджуючи його з провідними і суперниками.

МАСОВІ ОНЛАЙНОВІ (*MASSIVELY MULTIPLAYER ONLINE GAME, MASSIVELY MULTIPLAYER ONLINE RPG*)

Масові ігри, що використовують підключення до мережі Інтернет. У більшості таких ігор відсутня одиночна гра (в офлайн). Найбільш часто зустрічаються жанри – настільні та рольові ігри (MMORPG). Серед них розрізняють також ігри, які запускаються в браузері і не потребують скачування та/або установки чогось на комп'ютер. Зі звичайних багатокористувацьких ігор виділяється жанр MUD – текстові онлайн ігри.

Приклади (MMORPG): World of Warcraft, Star Wars: The Old Republic і ін.

КЛАСИФІКАЦІЯ ЗА СТИЛІСТИКОЮ

- Вестерн – Дія в вестернах в основному відбувається в другій половині XIX століття на Дикому Заході – в майбутніх західних штатах США, а також у Західній Канаді і Мексиці.

- Кіберпанк – занепад людської культури на тлі технологічного прогресу в комп'ютерну епоху. Описують антиутопічний світ майбутнього, в якому високотехнологічний розвиток, інформаційні технології та кібернетика поєднуються з глибоким занепадом або радикальними змінами в соціальному устрої.
- Космічні.
- Постапокаліптичні – стиль, що несе настрій пустельності, самотності і страху в образах постарілої і покинутої техніки або будівель.
- Стімпанк – моделюють цивілізацію, де досконало освоїли механіку і технології парових машин.
- Сучасні.
- Фентезі – засновані на використанні міфологічних і казкових мотивів. У сучасному вигляді сформувалися на початку ХХ століття.

КЛАСИФІКАЦІЯ ПО ПЛАТФОРМАХ

Виділяються наступні типи ігрових платформ:

- персональні комп'ютери;
- ігрові консолі / приставки;
- мобільні телефони і КПК.

Відносно кількості платформ, на яких працює гра, існує також поділ на:

- **Мультиплатформені / Ігри для всіх платформ** (такі, що вийшли на двох і більше платформах).
- **Одноплатформені ігри** (такі, що вийшли тільки на одній платформі – платформенні ексклюзиви).

1.3 ІГРОВІ ПРОФЕСІЇ

У процесі виробництва гри в справу вступають представники багатьох «ігрових» професій – поговоримо про основні з них. До речі, при розробці невеликого проекту в рамках обмеженого бюджету одна людина може поєднувати в своїй особі обов'язки цілої команди розробників.

ПРОГРАМІСТИ

Програмісти зайняті роботою з написання програмного коду гри. Їхніми зусиллями реалізується ігрова фізика, штучний інтелект, з яким належить битися гравцеві при грі «проти комп'ютера» та багато іншого. Що цікаво, багато ігрових програмістів стали такими після того, як почали програмувати самотійно, у вигляді хобі.

Якщо говорити про інструменти програміста – то практично всі комерційні ігри написані на мові C++ або C; деякі особливо відповідальні частини гри пишуть на мові Assembler. Останнім часом набуває популярності мова програмування C#. Строго кажучи, гру реально написати практично на будь-якій мові програмування – наприклад, прості ігри можна створювати в Microsoft Word або Microsoft Excel, використовуючи вбудований в ці продукти Visual Basic For Applications.

ХУДОЖНИКИ

Роль художників і взагалі всіх, хто працює з графікою, в сучасному створюванні ігор важко переоцінити. У всі часи одним з критеріїв оцінки гри була її графічна складова – а сучасні засоби роботи з графікою дозволяють створювати красиві ігрові світи, в основі яких лежить копітка робота художників в усіх її проявах.

До того ж, аніматори оснащені сьогодні передовим апаратним і програмним забезпеченням (зокрема – технологією Motion Capture) яке, наприклад, дозволяє записувати реальні рухи людини і переносити їх потім на ігрових персонажів.

МУЗИКАНТИ

Композитори, музиканти, актори, звукорежисери працюють над звуковим оформленням гри. Вони пишуть і виконують музику, читають тексти персонажів. Без гідної музики і якісного озвучування, як і без гарної графіки, сучасна гра навряд чи буде успішною. Хоча звуки і музика, звичайно, не головне в більшості ігор, але, наприклад, пограйте в Need For Speed Undercover для Windows Phone 7 без звуків і музики – чимала частина чарівності гри зникне без сліду.

ПИСЬМЕННИКИ

Як правило, будь-яка сучасна гра побудована навколо певного сценарію. Особливо сильною є роль сценаріїв у рольових, пригодницьких іграх і в іграх змішаних жанрів, однак навіть якщо робота ведеться над черговою FPS-грою – якісні тексти їй не завадять. Всім цим займаються сценаристи, режисери, письменники.

ДИЗАЙНЕРИ РІВНІВ

Коли програмісти, художники, музиканти виконують основну роботу зі створення персонажів гри, ігрових інтер'єрів, звуків, за справу беруться дизайнери рівнів. Якщо навіть гра представляє собою один великий «рівень» без явного розбиття на частини, дизайнерам доводиться чимало попрацювати над створенням ігрового всесвіту.

ТЕСТЕРИ

Гра – це величезна програма, яка перед виходом у світ повинна бути протестована. Тестування відіграє дуже важливу роль в процесі створення ігор – іноді тестерів набирають з числа добровольців. Охочих взяти участь в попередньому тестуванні гри, як правило, більше, ніж потрібно – багато хто хоче спробувати новинку першими. Якщо в ході тестування виникають помилки, тестер сповіщає про них розробників.

Багато початківців-розробників, що хочуть самі розробляти ігри, задаються питанням: «Чи може одна людина, або, принаймні, команда з декількох ентузіастів створити сьогодні гру, яка принесе розробникам славу, багатство і стане бажаним гостем на ігрових пристроях гравців усього світу»? Це складне питання.

Історія знає безліч прикладів, коли біля витоків будь-якої гри, що стала шалено популярною у всьому світі, стояла невелика група ентузіастів. Наприклад – це всім відомий Tetris, який в практично незмінному вигляді існує вже не одне десятиліття.

Або ігрова серія Final Fantasy – перша версія цієї гри з'явилася наприкінці 1980-х років, а тепер – це цілий ігровий всесвіт, над новими версіями якого працює величезна кількість людей.

Якщо говорити про мобільні ігри, то вони з одного боку мають усі ознаки сучасних ігор. Серйозний мобільний проект вимагає не менше зусиль, ніж ігри для інших платформ. Однак, особливості мобільних ігор, зокрема висока популярність досить простих за своїм устроєм ігор, в які грають, щоб скоротати час (так званих казуальних ігор), відкриває дорогу в світ мобільної розробки невеликим командам розробників. Такі ігри порівняно прості в розробці. Візьмемо, наприклад, таку популярну гру Tetris, з розробкою власної реалізації якої впорається і один програміст, різні варіанти гри Lines, пасьянси, незліченні реалізації ігор Snake, Pacman, Breakout. Ці ігрові ідеї живуть уже багато років (а то й десятиліття) і не втрачають актуальності, постійно перевтілюючись, але зберігаючи сутність вихідних ідей. І якщо розробнику вдасться втілити в життя цікаву ігрову ідею, що залучає користувачів, успіх йому забезпечений.

Потрібно визнати, що поодиноці зараз набагато складніше, ніж, скажімо, 20 років тому, створити значну гру. Але ігровий бізнес в найбільшому ступені серед інших заснований на інтуїції і фантазії окремих людей – і якщо у вашій голові народиться геніальна ігрова ідея і ви зможете донести її до інших людей – цілком можливо, що світ стане свідком народження чергової великої гри.

ЛЕКЦІЯ 2 РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ

Розробник комп'ютерної гри має знати всі аспекти створення ігор – технічні (особливості художнього і звукового оформлення, програмування і інтерфейсу) і ділові (маркетинг, управління розробкою проекту, бюджетування). Розробник повинен знати особливості цільової аудиторії, розуміти мету гри, орієнтуватися в ігрових жанрах.

Цільова аудиторія – це люди, на яких розраховано гру. Цільовою аудиторією можуть бути діти, чоловіки, жінки, що належать до різних вікових і соціальних груп.

2.1 ОСНОВНІ ЕЛЕМЕНТИ КОМП'ЮТЕРНОЇ ГРИ

Основними елементами комп'ютерної гри є зображення, звук, інтерфейс.

2.1.1 ЗОБРАЖЕННЯ

Зображення – це те, що гравець бачить на екрані. Для створення зображень використовують програмні інструменти і методи. Перед створенням зображення художнику-розробнику необхідно розуміти, як буде використовуватися зображення.

Етапи створення зображення:

- 1) створення ескізу на папері;
- 2) підготовка двомірного зображення з використанням 2D-програм;
- 3) конструювання та візуалізація тримірних об'єктів за допомогою 3D-програм.

Комп'ютерна графіка буває 2D (двомірна, об'єкти є плоскими) і 3D (тримірна). У 3D зображуються об'ємні об'єкти в осях прямокутної системи координат, вісь X – горизонтальна, вісь Y – вертикальна, вісь Z спрямована на спостерігача або від нього.

Мінімальним елементом зображення є піксель – точка. Пікселі зображення розташовано по рядках і стовпцях. При значному збільшенні зображення пікселі перетворюються на великі зерна. Ступінчастість зображення згладжують за допомогою методу інтерполяції.

Роздільна здатність (resolution) – це кількість пікселів на одиницю розміру зображення по горизонталі і вертикалі. Якість зображення вимірюється в одиницях dpi – кількість пікселів на дюйм зображення.

Важливим показником роздільної здатності є відношення ширини пікселя до його висоти (aspect ratio), тому що не всі пікселі мають форму квадрата.

Пікселі зображення характеризуються значеннями координат на екрані і кольором. Колір задається номером і являє собою в палітрі RGB суміш червоного (Red), зеленого (Green) і синього (Blue) кольорів. Кожен піксель може приймати один колір з невеликої (від 2 до 256) кількості кольорів. З урахуванням глибини кольору кількість кольорів розширюється до мільйонів. Глибина кольору – кількість біт, що використовується для зберігання і представлення кольору при кодуванні одного пікселя растрової графіки або відеозображення.

У ході розробки проекту необхідно маніпулювати зображеннями – вирізати, копіювати, обертати і т.д.:

- Cut – виділений фрагмент зображення видаляється зі сцени;
- Copy – виділений фрагмент зображення копіюється в буферну пам'ять і може бути вставлений в іншому місці;
- Paste – фрагмент зображення вставляється з буферної пам'яті в інше місце;
- Skew – перекид, нахил, викривлення, деформація зображення;
- Rotate – обертання зображення або поворот на заданий кут;
- Resize – зміна розміру, масштабу зображення;
- Crop – обрізання зображення відповідно до заданого контуру;
- Flip – дзеркальне відображення зображення щодо горизонтальної або вертикальної осі.

Крім названих простих операцій можна застосовувати і більш складні.

- спрайт – невеликі зображення (персонажі, предмети і ін.), що можна переміщати по екрану. До спрайту можна застосовувати анімацію. Невеликі спрайти можуть рухатися по великому зображенню. При анімації створюється ряд шаблонних зображень об'єкта, які показують на екрані в певній послідовності, що створює ефект руху;
- маска – зображення, що як трафарет накладається на інше (фонове) зображення. Маска може робити частину зображення невидимою.

У ряді ігор для визначення прозорого кольору використовують певний елемент (зазвичай це останній елемент) з таблиці кольорів. В цьому випадку при візуалізації для імітації прозорості областей, що зафарбовані певним кольором, використовується довільний колір – останній з таблиці кольорів.

Для створення частково прозорого зображення з початкового кожному пікселю результуючого зображення привласнюють деяке значення з проміжку між кольором фону і накладеної маски.

При згладжуванні (anti-aliasing) нерівностей контура намагаються забезпечити перехід кольору від краю зображення до кольору фону, при цьому ступінчастість стає менш помітною.

Графічні зображення можуть зберігатися в файлах з різними форматами в залежності від апаратних можливостей, дизайну, аспектів безпеки. Основні фактори вибору формату збереження – якість і зручність використання зображення. Слід знаходити компроміс між суперечливими вимогами – якістю зображення та його розміром.

2.1.2 ЗВУКИ

Звукове оформлення має велике значення для гри. Звуковий супровід є компонентом, що визначає якість гри і професіоналізм розробника, задоволеність користувача процесом гри.

Звук має великий вплив на сприйняття користувача. Звукові ефекти і звуковий супровід можна створити самостійно, записавши відповідні звуки, або з інших джерел. Професійні розробники мають для цього спеціальні інструменти.

2.2 ЕТАПИ РОЗРОБКИ КОМП'ЮТЕРНОЇ ГРИ

Розробка комп'ютерних ігор – це досить чітко налагоджений процес, що має певні етапи, всі ці етапи проходять так чи інакше при створенні ігор. Однак життя зазвичай вносить свої корективи навіть у найчиткіший план.

Дуже часто розробники ігор не можуть встигнути доробити гру в скільки-небудь прийнятний термін – яскравий приклад – DukeNukem Forever, випуску якого весь ігровий чекав багато років. Практично завжди після виходу комп'ютерної гри за нею йдуть виправлення – вся справа в тому, що розробники, знову ж таки, не вкладаються у відведені їм терміни.

Тексти програм ігор нерідко «йдуть» в Інтернет і всі грають в новітню гру задовго до її офіційного релізу. Причому тут не можна однозначно сказати, чи шкідливо це для ігрових компаній. З одного боку – шкідливо – адже копії гри потрапляють до користувачів абсолютно безкоштовно (не рахуючи витрат на трафік в Інтернеті). Однак з іншого – часто «йдуть» коди, далекі від досконалості і «витік» лише розігріває інтерес до фінальної версії гри.

Треба врахувати, що в ігровому бізнесі існують два типи компаній – розробники (developer) і видавці (publisher). Якщо розробник і видавець збігаються – процес розробки гри лише виграє – розробнику немає потреби переконувати стороннього видавця в доцільності капіталовкладень у розробку.

Розглянемо етапи розробки типової комп'ютерної гри.

ПІДГОТОВКА ДО ВИРОБНИЦТВА

Підготовка до виробництва гри – це перший етап роботи над грою. Завдання розробників на цьому етапі – розробити концепцію гри, дизайн персонажів, вибрати засоби для реалізації проекту, створити прототип гри, підготувати план, за яким буде створюватися гра, і узгодити цей план з начальством, або – з компанією, яка планує видавати гру. Як правило, всі сучасні ігри пишуться під конкретного видавця, що вкладає в розробку чималі кошти.

Коли всі адміністративні питання вирішені, гра переходить на етап виробництва.

ВИРОБНИЦТВО

Виробництво – це ключовий етап у створенні гри. Розробники займаються реалізацією раніше створеного плану. Однак початковий план гри піддається змінам – іноді ці зміни відбуваються дуже часто – аж до щоденних коригувань.

В ході виробництва гри – особливо це стосується комерційних версій – періодично влаштовується розгляд поточних результатів розробки, до якого команда має представити проект, що досяг певного рівня розвитку. Тобто, наприклад, до одного з таких моментів має бути готовою демо-версія гри, що працює, до іншого – перший рівень і т. д. Як правило, ці проміжні результати служать відмінною рекламою нових ігрових проектів – демо-версії публікують на ігрових сайтах, геймери «приміряють» до цих версій можливості свого обладнання.

ВИПУСК

Після того, як гру створено, протестовано і налаштованою, настає час її випуску. Як правило, інтерес до цієї події посилено підігрівається видавцем гри – адже не варто забувати, що головна мета видавця – прибуток. Як правило, найбільш успішні ігри з надлишком виправдовують очікування видавців.

ПІДТРИМКА

Ігри для ПК часто виходять з помилками – вся справа в тому, що розробникам вічно не вистачає часу, щоб все як слід налагодити. На щастя, є можливість виправляти помилки на вже встановлених іграх, встановлюючи патчі (від англійського patch – латка). Цим користуються розробники, випускаючи сируваті ігри і, після цього, цілу низку латочок (патчів) для них. Така практика не поширена для консольних ігор – тут розробники змушені відповідальніше підходити до своєї роботи і випускати повністю робочу гру, яка не потребує втручань.

2.3 ІГРОВІ РЕСУРСИ

2.3.1. ІГРОВІ РЕСУРСИ

Будь-яка гра складається з багатьох частин. Але серед них можна виділити дві основні. Перша – це ігрові ресурси, і друга – це програмний код.

Ігрові ресурси – це графічні, музичні та інші ресурси, що використовуються для оформлення гри. Розглянемо різні формати, в яких зберігаються ігрові ресурси.

Графічні файли

Графічні файли використовуються для зберігання зображень. Як правило, це – графічні зображення ігрових об'єктів, які застосовуються в двовірних або в псевдотримірних іграх, коли тримірний вигляд ігрових об'єктів досягається лише художніми засобами, без використання технологій тримірної графіки. Так само ці файли можуть бути використані для створення елементів оформлення гри і як текстури для накладення на тримірні моделі.

JPEG

JPEG – це стандарт зберігання файлів зображень, розроблений спеціально для зберігання цифрових фотографій. *JPEG* розшифровується як Joint Photographic Experts Group – саме так називалася робоча група, що розробила цей стандарт. Особливістю *JPEG* є можливість стиснення зображення з втратами якості. Як правило, в *JPEG* виділяють 10 рівнів стиснення (від 1 до 10), проте ці зображення можна стискати і з більш точним зазначенням рівня стиснення. Чим сильніше стиснення – тим більше втрати якості. На невисоких рівнях стиснення *JPEG*-файли практично не містять так званих артефактів стиснення. У випадку з *JPEG* це проявляється в помітному спотворенні зображення, особливо в тому, що містить чіткі межі між кольорами, чіткі лінії. Справа в тому, що при стисненні за алгоритмом *JPEG*-зображення розбивається на фрагменти 8x8 пікселів.

Як правило, в JPEG зберігають різні двомірні зображення, які можна використовувати для організації елементів інтерфейсу користувача, для створення різних екранів гри (екран вітання, екрани з інформацією про набрані очки і так далі). В JPEG можна зберігати зображення (їх називають спрайтами), що використовуються в двомірних іграх в якості ігрових об'єктів (переважно – прямокутної форми), в якості фонових зображень.

Найкраще використовувати JPEG для зберігання кольорових зображень з плавними колірними переходами. Зокрема, це можуть бути фотографії, намальовані ігрові об'єкти та ін.

JPEG-файли мають розширення JPG або JPEG.

Для створення JPEG-файлів з успіхом можна використовувати популярні графічні редактори – такі, як Adobe Photoshop різних версій.

PNG

Формат PNG (Portable Network Graphics) відрізняється від формату JPG тим, що використовує алгоритми стиснення, що стискають зображення без втрати якості. В результаті виявляється, що зображення, які мають чіткі переходи кольорів (такі, як схеми, графіки, зображення для простих елементів управління) можуть бути досить сильно стиснуті в PNG-файлах. Якщо зображення має плавні переходи кольорів – кращим вибором для його стиснення буде JPG, хоча й PNG цілком можна використовувати. Як і JPG, даний формат підтримує повнокольорові зображення, проте зображення можна створити з використанням декількох фіксованих кольорів, що йде на користь розміру графічного файла, і, природно, при правильному підборі кольорів, не позначається на його якості.

Формат PNG було розроблено для заміни застарілого, але все ще популярного формату GIF. Основне призначення GIF-файлів – проста інтернет-графіка.

PNG-файли мають розширення .PNG. Для їх створення можна використовувати растрові графічні редактори, – такі, як Adobe Photoshop.

Ще одна важлива особливість формату PNG – підтримка прозорості. Тобто при створенні файлу можна вказати, які з пікселів зображення вважати прозорими. В результаті PNG ідеально підходить для зберігання графічного зображення двомірних ігрових об'єктів складної форми.

TGA

TGA(Truevision Graphic Adapter), TARGA – (Truevision Advanced Raster Graphics Adapter) – це графічний формат, що використовується переважно для зберігання ігрових текстур. Цей формат було спеціально створено для зберігання текстур, він використовується у багатьох існуючих іграх саме для цих цілей. Він підтримує повнокольорові зображення (8, 16, 24, 32-бітові), підтримує 8-ми бітний альфа-канал для зазначення прозорих ділянок зображення.

Текстура- це растрове (точкове) зображення, що накладається на поверхню полігону,

та є частиною тримірної моделі. Текстури дозволяють надавати тримірним моделям кольору, створювати видимість рельєфу, дрібних деталей моделі. При проектуванні 3D-моделі створення дрібних деталей зазвичай потребує дуже багато ресурсів, використання ж текстур дозволяє отримувати привабливий вигляд моделі, створеної з використанням порівняно невеликої кількості тримірних елементів. Сьогодні тримірні моделі створюються з високим рівнем деталізації – потужності сучасних комп'ютерів вистачає для реалістичної обробки таких моделей. В іграх минулих років моделі зазвичай виглядають схематично (наприклад, профіль колеса автомобіля в іграх, які використовують низькополігональних моделей, може виглядати як багатокутник, а не коло).

Текстури розрізняються за глибиною кольору і розподільною здатністю – чим вище те та інше, тим вищою є якість зображення. Відносно текстур існує таке поняття, як текстель – число пікселів, що припадає на мінімальну одиницю текстури.

TGA-файли мають розширення *TGA*

Для створення *TGA*-файлів можна застосовувати все той же Adobe Photoshop та інші редактори. Наприклад, безкоштовний редактор Paint.NET (<http://www.getpaint.net/>)

DDS

DDS (Direct Draw Surface) – це графічний формат, розроблений спеціально для використання в DirectX SDK. Його призначено переважно для зберігання текстур. Завдяки особливостям формату – це підтримка збереження зріджених і незжатих текстур, апаратна підтримка, цей формат є ідеальним для зберігання ігрових текстур.

DDS-файли мають розширення *DDS*. Для їх створення можна використовувати плагіни для звичайних графічних редакторів, для роботи з такими файлами створено спеціальні набори утиліт і плагінів.

BMP

BMP (Bitmap) – це графічний формат, який зберігає зображення у стисненому вигляді. Він підтримує алгоритм стиснення, який проте використовується досить рідко. Створювати *BMP*-файли можна практично у всіх растрових редакторах. Особливість *BMP* – широке поширення при створенні графічних інтерфейсів користувача в різних Windows-програмах. *BMP*-файли зазвичай мають порівняно великий розмір.

ФАЙЛИ ТРИМІРНИХ МОДЕЛЕЙ

Тримірні моделі використовуються в тримірних об'ємних іграх. Для додання реалістичного вигляду на такі моделі зазвичай накладають файли текстур.

FBX, X

FBX (Autodesk FBX) – це універсальний формат для зберігання тримірних моделей і супутньої інформації. Він використовується у багатьох тримірних додатках, зокрема, дуже популярний в комп'ютерних іграх.

Існують плагіни для популярних програм створення тримірної графіки, що дозволяють

конвертувати створені в них моделі в цей формат.

Формат X – це формат тримірних файлів, який використовується DirectX.

Файли опису шрифтів, SPRITEFONT

SPRITEFONT-файл – це xml-файл з настройками шрифту. Він містить інструкції, що стосуються візуалізації шрифту в ігровому вікні XNA-проекту. Файли цього типу мають розширення .SPRITEFONT.

ЗВУКОВІ ФАЙЛИ

MP3 – стиснений звуковий файл. Цей формат має величезну популярність за рахунок можливості сильного стиснення звуку з прийнятною якістю. *WAV* – зазвичай являє собою нестиснений файл звуковий файл, як правило, може використовуватися для відтворення коротких звукових фрагментів (звукових ефектів) високої якості.

Всі названі вище формати файлів – це ігрові ресурси, що можуть бути включені в ігровий проект. З огляду на тенденцію до зближення технологій XNA і Silverlight, що спостерігається сьогодні, не можна чітко розділити названі ігрові ресурси між двома технологіями.

2.3.2. ІГРОВА ТЕРМІНОЛОГІЯ

Ігрова термінологія складається під сильним впливом англійських назв. Часто в українській мові досить складно знайти відповідну аналогію прийнятим англійським термінам, тому в сфері ігрової термінології існує подвійність – досить часто творці ігор користуються англійськими термінами, в той же час як і україномовні назви так само знаходять застосування. Розглянемо деякі терміни, які використовуються в ігровій індустрії.

Sprite – цей термін часто замінюють україномовним неологізмом *спрайт* – словник Lingvo 12 визначає поняття «спрайт» як «елемент динамічного графічного відображення». В ігровій індустрії синонімами слова *спрайт* є такі слова, як «зображення», «картинка», іноді користуються словом «текстура», проте зазвичай це поняття має дещо інше смислове навантаження. Як правило, *спрайт* – це двовірне зображення, причому у вузькому сенсі слова – це лише зображення, а в широкому – це ігровий об'єкт, який має набагато більш широкий набір можливостей, ніж звичайне зображення. Спрайти мають прямокутну форму, проте в комп'ютерних іграх часто зустрічаються непрямокутні об'єкти. Це досягається за рахунок прозорих областей при малюванні спрайту.

Texture – текстура. Зазвичай текстурами називають двовірні зображення, які «накладають» на тримірні моделі. У термінології XNA поняття текстури і спрайту при розробці двовірних ігрових об'єктів збігається.

Background – фон. Так називається зображення, зазвичай відповідає розмірами ігровому полю, та є фоном для інших зображень. Фон може бути нерухомим і рухомим. Рухомий фон (*scrolling background*) використовується в іграх, які називаються Скролери (*scrollers*). Скролінг – це один з прийнятих ігрових термінів. Він означає прокрутку, переміщення вмісту вікна. Скролінгові ігри надзвичайно поширені серед двовірних ігор. Наприклад, використання скролінгового фону дозволяє створити ілюзію руху в двовірному гоночному симуляторі. Фон у двовірній грі може складатися з декількох

частин, що рухаються з різною швидкістю – це дозволяє створити ефект тримірності ігрового світу.

2D-game – двовимірна гра – гра, в якій використано двовимірні зображення.

3D-game – тримірна гра – використовує тримірні моделі та тримірний ігровий світ.

Tile- тайл – невелике зображення, яке використовується для конструювання рівнів в іграх. *Tile* можна перевести як «мозаїка» або «черепиця».

Polygone (полігон, багатокутник) – просторовий багатокутник, який використовується для створення тримірних об'єктів. Як правило, у комп'ютерній графіці використовуються трикутники.

Pixel (піксель) – найменший елемент растрового зображення, точка, яка відображається на екрані. Зазвичай в пікселях вимірюють роздільну здатність текстур (наприклад – 1024x768), екранну роздільну здатність монітора, розміри ігрових вікон. Слово *Pixel* – це аббревіатура від *Picture's ELe ment*.

Texel (тексель) – точка текстури в тримірному просторі. Слово *Texel* – це скорочення від *Texture Element*.

Voxel (воксель) – точка тримірного зображення. Це слово – аббревіатура від *Volumetric Pixel* – об'ємний піксель.

Texture Filtering (фільтрація текстур) – зменшення спотворень при накладенні текстур на тримірний об'єкт.

Camera (камера) – так називають точку в ігровому просторі, з якою гравець бачить ігровий світ. Відносно положення камери ігри можна поділити на ігри від першої особи (камеру розташовано так, що реалізується вид немов очима персонажа), ігри з видом зліва-зверху (стратегії), ігри з видом ззаду – камеру розташовують зазвичай позаду гравця і трохи вище за нього. Існують ігри, де камера може займати різні позиції, наприклад, реалізують вид з очей персонажа і вид ззаду.

Transparency (прозорість) – прозорими можуть бути частини двовимірних зображень – це дозволяє створювати зображення складної форми, які фактично обмежені прямокутником. Прозорими бувають і об'єкти в тримірному ігровому світі наприклад, – це може бути прозора вода або скло, певним чином заломлюють світло.

Light Model (модель освітлення) – способи моделювання освітлення об'єктів.

2.4. ІГРОВА ДОКУМЕНТАЦІЯ

Якщо ви починаєте розробляти серйозний ігровий проект вам знадобиться ігрова документація. Цю лекцію присвячено питанням розробки такої документації. Зокрема, ми поговоримо про концепт-документи, дизайн-документи і плани розробки гри.

2.4.1 КОНЦЕПТ-ДОКУМЕНТ

Розробка гри починається зі створення концепції гри. Власне кажучи, концепція – це та центральна ідея, навколо якої будується все інше. Концепція гри виражається у вигляді концепт-документа.

Концепт-документ може будуватися за такою схемою:

- вступ;
- жанр і аудиторія;
- основні особливості гри;
- опис гри;
- порівняння і передумови створення гри;
- платформа;
- контакти.

Документ concepttemplate.doc містить шаблон концепт-документа від 1С, документ conceptexample.doc – приклад концепт-документа.

ВСТУП

У вступі міститься коротке формулювання ідеї гри. Вступ має містити інформацію, що дозволить читачеві зрозуміти суть гри, її жанр, аудиторію. Фактично, вступ – це той мінімум інформації, який дозволяє читачеві – видавцеві, майбутньому члену команди розробників, майбутньому гравцеві – зрозуміти, чи цікава йому дана гра, чи хоче він мати до неї відношення.

ЖАНР І АУДИТОРІЯ

Відомості про жанр гри і про цільову аудиторію дуже важливі, тому що дозволяють зрозуміти особливості позиціонування гри, особливості майбутнього просування готового продукту. Як правило, в даний розділ є сенс включати інформацію про цільові групи користувачів, яким може бути цікава гра. Тут же корисно навести результати попередньо проведеного дослідження, наприклад – невеликого опитування потенційних гравців, а можливо – і гравців інших груп.

Таке дослідження можна провести наступним чином: скласти опитувальник, що складається з декількох питань, що призначені для з'ясування ставлення гравців до тих чи інших концепцій, покладених в основу гри. Питання має сенс складати у вигляді анкети, що містить варіанти відповіді на питання. Таку анкету можна використовувати як в електронному вигляді – наприклад, в інформаційних розсилках або на інтернет-ресурсах, так і в друкованому. Наприклад, опитувальник може містити питання за наступними напрямками:

- вік;
- стать;
- рід занять;
- як ви ставитеся до ігор жанру X?
- скільки часу ви зазвичай проводите, граючи в комп'ютерні ігри?
- як ви вважаєте, чого не вистачає іграм жанру X, які ви знаєте?
- визначте ваше ставлення до наступних нововведень, що можна зробити в грі жанру X:
 - нововведення 1
 - нововведення 2 ...
 - ...
 - нововведення n.

Це – лише шаблон для опитування потенційних гравців, для практичного застосування його має бути допрацьовано.

Цей розділ може містити не тільки відомості про жанр і про цільову аудиторію гравців, але й додаткові дані про позиціонування гри.

ОСНОВНІ ОСОБЛИВОСТІ ГРИ

Цей розділ має містити опис ключових особливостей гри, що відрізняють її від ігор того ж жанру, розрахованих на ту ж цільову групу. У цьому ж розділі слід вказати приблизний розмір гри. Розмір гри може бути виміряно в годинах або в інших одиницях.

ОПИС ГРИ

Тут слід детально описати суть гри, особливості дій гравця.

ПОРІВНЯННЯ І ПЕРЕДУМОВИ СТВОРЕННЯ

Цей розділ має містити порівняльний аналіз запропонованої гри з існуючими іграми, аналіз ринкових тенденцій. Тут же слід торкнутися питань ліцензування.

ПЛАТФОРМА

Цей розділ має містити інформацію про платформу, на якій планується створення гри. У ролі платформи може бути використано РС або ігрову приставку. Можливо, гра буде створюватися для декількох платформ. У випадку з РС слід вказати системні вимоги, виділивши мінімальні та рекомендовані вимоги. Зокрема, тут можна вказати вимоги до процесора, ОЗУ, відеокарти, звукової карти, мережевого обладнання і т.д.

КОНТАКТИ

У цьому розділі вказують контактні відомості розробників гри.

2.4.2 ДИЗАЙН-ДОКУМЕНТ

Концепт-документ є першим ігровим документом, що дозволяє отримати загальне уявлення про гру, фіксує її основні особливості. Дизайн-документ містить детальний опис гри, ігрових ресурсів, ігрової логіки. Дизайн-документ можна вважати розширеною версією концепт-документа.

Дизайн-документ може мати таку структуру:

1. Вступ

2. Концепція

2.1. Вступ

2.2. Жанр і аудиторія

2.3. Основні особливості гри

2.4. Опис гри

2.5. Передумови створення

2.6. Платформа

3. Функціональна специфікація

3.1. Принципи гри

3.1.1. Суть ігрового процесу

3.1.2. Хід гри і сюжет

- 3.2. Фізична модель
- 3.3. Персонаж гравця
- 3.4. Елементи гри
- 3.5. «Штучний інтелект»
- 3.6. Режим, що розрахований на багато користувачів
- 3.7. Інтерфейс користувача
 - 3.7.1. Блок-схема
 - 3.7.2. Функціональне опис і управління
 - 3.7.3. Об'єкти інтерфейсу користувача
- 3.8. Графіка і відео
 - 3.8.1. Загальний опис
 - 3.8.2. Двомірна графіка і анімація
 - 3.8.3. Тримірна графіка і анімація
 - 3.8.4. Анімаційні вставки
- 3.9. Звуки і музика
 - 3.9.1. Загальний опис
 - 3.9.2. Звук і звукові ефекти
 - 3.9.3. Музика
- 3.10. Опис рівнів
 - 3.10.1. Загальний опис дизайну рівнів
 - 3.10.2. Діаграма взаємного розташування рівнів
 - 3.10.3. Графік введення нових об'єктів
- Контакти

2.4.3 ПЛАН РОЗРОБКИ ГРИ

План розробки гри – це робочий документ, який спрямовано на деталізацію деяких положень, наявних в раніше складених документах і на планування створення гри. Він може мати наступну структуру:

1. Аналіз ринку

- 1.1. Цільова аудиторія
- 1.2. Хіти та порівняння

2. Технічний аналіз

3. Ресурси проекту

- 3.1. Персонал
- 3.2. Устаткування
- 3.3. Програмне забезпечення
- 3.4. Фінансові ресурси
- 3.5. Ризики проекту
- 3.6. Календарний план

ЛЕКЦІЯ 3 МЕНЕДЖМЕНТ У GAME-ПРОЕКТІВ

3.1 ВИДИ ПІДПРИЄМСТВ У ІГРОВІЙ ІНДУСТРІЇ

Розробник відеоігор — це розробник програмного забезпечення, що спеціалізується на розробці відеоігор — а саме на процесі і пов'язаних з ним дисциплінах з області створення відеоігор. Розробником ігор може бути як єдина людина, яка бере на себе всі завдання, так і велике підприємство з розподілом обов'язків між співробітниками по галузях діяльності — програмування, дизайн, художня складова, тестування і т.д. Більшість компаній-розробників відеоігор мають фінансову підтримку видавців і зазвичай маркетингову підтримку. Розробники, що існують за рахунок власних ресурсів, відомі як незалежні або інди-розробники і зазвичай створюють інди-гри.

Розробник може спеціалізуватися на певній консолі (як, наприклад, Nintendo's Wii U, Microsoft's Xbox One, Sony's Play Station 4), або може розвиватися на базі ряду систем (включаючи персональний комп'ютер). Розробники відеоігор спеціалізуються на певних жанрах ігор (наприклад, рольові відеоігри або шутери від першої особи). Деякі концентрують свою діяльність на перенесенні гри з однієї системи в іншу, деякі на перекладі гри з однієї мови на іншу. Рідше деякі компанії, окрім розробки ігор, займаються іншими видами робіт з розробки програмного забезпечення.

Більшість видавців відеоігор містять студії розробки (як, наприклад, Electronic Arts EA Canada, Square Enix's studios, Radical Entertainment, Nintendo EAD, що належать Sony Polyphony Digital b Naughty Dog). Однак, оскільки процес видання для них є основною діяльністю, вони, як правило, є швидше «видавцями», ніж «розробниками». Розробники можуть бути також і приватними (наприклад, Bungie – компанія, яка створила серію Halo ексклюзивно для Xbox).

Види підприємств в ігровій індустрії:

- розробник;
- видавець;
- змішаний тип;
- інди.

Всі ці типи не статичні і більшість компаній проходять еволюцію з одного типу в інший. Успішні компанії, як правило, в результаті приходять до змішаного типу. А інди-компанії можуть залишатися інди дуже довго.

Компанії-розробники

Компанії-розробники є найпоширенішим типом. Створюються, як правило, ентузіастами для того, щоб зробити гру мрії. Ентузіазм і бюджет не завжди збігаються, і якщо розробники не знаходять інвесторів, то припиняють своє існування. Якщо перший проект в якомусь вигляді вийшов, то у розробника є основний шлях – розширюватися, робити більш крупні проекти або рухатися до змішаного типу, шукати інші студії.

Є й інший шлях, – коли перший проект вийшов, гроші отримано, але на великий крок

їх не вистачає, тоді настає режим стагнації, коли компанія починає працювати або аутсорсерами або переходить на постійну роботу під видавця.

Для компаній-розробників такий шлях є швидше занепадницьким, тому що випадків, коли б на такому шляху розробники знову виростили до великої успішної компанії, практично немає.

ВИДАВЦІ

Тип, який в чистому вигляді, як правило, живе не дуже довго. Це компанії, які створюються професіоналами ігрової індустрії, інакше видавництво навряд чи буде успішним.

У таких компаніях спостерігається мінімум демократії, точковий підхід до топ-менеджерів, команда добре узгоджена. Робота в компанії-видавництві є хорошою школою для співробітника ігрової індустрії. У чистому вигляді команди-видавці зустрічаються досить рідко. Як правило, вони переходять в змішаний тип, створюючи постійні команди.

На відміну від компаній-розробників, створених з нуля, це компанії, що спочатку більш формалізовані, з бюджетом і командою професіоналів.

В першу чергу вони займаються пошуком розробників на ринку. Більшість видавців працює досить успішно. Якщо ви перебуваєте в пошуках роботи в ігровій індустрії, найкраще почати з компанії-видавця, тому що вона найкраще підходить для того, щоб навчитися, «прокачатися» і йти далі.

ЗМІШАНИЙ ТИП

У більшості випадків компанії-розробники і компанії-видавці переходять в змішаний тип. Це або успішні розробники, які вирішили працювати без видавця, або компанія, яку було створено як видавництво, але в процесі роботи для економії вона обзавелася внутрішніми студіями розробки.

Для успіху важливо знайти інвестора на старті і вчасно потрапити на ринок, заробити достатньо грошей на старті. У таких компаній з'являється можливість видання інших проектів – як своїх, так і інших розробників.

Більшість великих відомих компаній (EA, Blizzard, Ubisoft) – це компанії змішаного типу.

ІНДИ

Інди-розробники – це окремий тип. Вони відрізняються від попередніх типів в першу чергу своїм стихійним підходом. Компанії створюються під проект без будь-яких далекосяжних планів. У таких командах спостерігається слабе розуміння ринку, але, випадково потрапивши в тренд, вони можуть успішно просуватися, і при цьому цей ринок часто створюють з нічого. Інди можуть залишатися інди, не переходячи в інші типи.

3.2 КОМАНДА ФАХІВЦІВ В GAME-ПРОЕКТАХ

Ігрових фахівців можна розділити на дві категорії – обов'язкові і опціональні фахівці.

Обов'язкові фахівці:

- продюсер;
- менеджер проекту/Scrum Master;
- розробник;
- гейм-дизайнер;
- арт-директор/дизайн-лідер;
- маркетолог;
- ком'юніті-менеджер (КМ)/фахівець техпідтримки.

Опціональні фахівці:

- фахівець по звуку;
- аналітик;
- технічний директор;
- QA/QC;
- бізнес-девелопер;
- фахівець з PR;
- фахівець з HR.

Без обов'язкових фахівців навряд чи вдасться зробити проект. У маленьких командах можливе суміщення ролей, коли одна людина бере на себе кілька функцій.

Опціональні фахівці не завжди присутні в проекті і використовуються найчастіше тільки у великих компаніях або на специфічних проектах.

ПРОДЮСЕР

Якщо компанія маленька, то продюсер, швидше за все, буде або власником, або засновником, або директором. Якщо в компанії використовуються гнучкі методології Agile, то, швидше за все, продюсер буде і власником проекту (product owner), і основним представником проекту.

Продюсер знає все про команду і проект. Основна відмінність продюсера від менеджера в тому, що він бачить команду як людей, а менеджер проекту бачить їх як фахівців.

У великих європейських компаніях сьогодні продюсер – це людина, яка працює з командою з точки зору психології, а менеджер проекту (ПМ) – людина, яка працює з функціями і завданнями в команді.

У маленьких командах продюсер добре поєднує функції HR з питання пошуку та найму нових людей і біздева за поданням свого продукту на виставках і роботі з партнерами.

ПРОДЖЕКТ-МЕНЕДЖЕР (ПМ)

У маленьких командах зазвичай є тільки хтось один: продюсер або ПМ.

Якщо в команді є тільки продюсер, і при цьому використовується Agile, ні в якому разі не можна в особі продюсера поєднувати функції product owner і scrum master, цю функцію потрібно буде перевести на іншу людину.

Якщо в команді тільки ПМ, то він буде виконувати функції scrum master, а product owner і візіонером буде, швидше за все, геймдизайнер.

Ключові обов'язки ПМ на проектах – це постановка завдань і контроль за їх виконанням, відстеження термінів здачі проекту. ПМ в курсі всіх можливих проблем – як існуючих, так і потенційних.

РОЗРОБНИК

Найпростіша, але незамінна функція – розробник. Головний момент, який потрібно знати про розробників – вони дуже різні. Не тільки в сенсі мови, а й характеру, підходу до роботи.

Ключовий момент у побудові команди – підібрати всіх за психологічною сумісністю.

Якщо люди в управлінні і геймдизайні зазвичай досить гнучкі, то серед розробників зустрічаються люди зі складними характерами. Тому важливо знайти розробників, які будуть на хвилі з іншими членами команди.

ГЕЙМ-ДИЗАЙНЕР

Для маленької команди гейм-дизайнер – це незамінна людина, який буде другою людиною за ступенем впливу на проект після продюсера або ПМ. Геймдизайнер знає все про гру. Більшість геймдизайнерів багато грають і добре розбираються в іграх.

Важливо пам'ятати, що це не просто людина з фантазією, вона повинна вміти добре писати технічне завдання як для художників, так і для розробників. Часта проблема – коли геймдизайнером беруть людину, яка нічого не вміє.

Геймдизайнер – найкраща позиція у великих компаніях, з якої можна починати зростання. Це єдина, крім тестувальника, позиція, куди можна потрапити практично без будь-яких навичок.

АРТ-ДИРЕКТОР/ДИЗАЙН ЛІД

На проекті вони так само незамінні, як і програмісти. Добре, якщо ваш художник може не тільки малювати сам, але вміє працювати з аутсорсерами, писати ТЗ.

Проблеми з нестачі графічного контенту зазвичай вирішуються за допомогою аутсорсерів, і якщо ваш художник не вміє працювати з іншими художниками, ви втратите багато часу.

У художників часто виникають проблеми зі звітністю (трекерами та документообігом). Важливо з самого початку правильно побудувати роботу з ними.

МАРКЕТОЛОГ

Багато хто недооцінює значення маркетолога для успіху проекту, і працюють з підходом: «спочатку зробимо гру, а потім будемо її просувати». Це неправильно, починати просування гри потрібно якомога раніше, з моменту затвердження концепту.

Маркетологи бувають різними і з різним набором функцій. На ігровому ринку, особливо в мобільних іграх – це людина, яка будує трафік і вивчає аналітику з вашої тематики.

Сильний маркетолог зможе дати багато найцінніших порад ще на ранній стадії. Якщо ви працюєте з видавцем, маркетолог буде на боці видавця.

Значущість маркетолога дуже прив'язана до умов ринку. Чим ринок старіше, тим він більш конкурентний і тим важливішою є робота маркетолога для фінансових показників проекту. Якщо ви намагаєтеся вирватися на якийсь інноваційний ринок, де, як ви думаєте, конкурентів у вас поки немає, то, можливо, на першому етапі без маркетолога можна обійтися.

Якщо команда маленька і на повний спектр фахівців грошей не вистачає, то маркетолог дуже добре поєднується з піарником, а часто ще й з ком'юніті-менеджером.

КОМ'ЮНІТІ-МЕНЕДЖЕР/ФАХІВЕЦЬ ТЕХПІДТРИМКИ

Фахівці, яких часто недооцінюють. Насправді, якщо гра буде просуватися через Steam, то КМ – це, напевно, друга після геймдизайнера людина, від якої залежить успіх. КМ буде працювати зі спільнотою, і важливо почати формування співтовариства ще до того, як гра буде готова.

КМ при відносно невеликих витратах може дати величезний приріст аудиторії за рахунок роботи в соцмережах. Всі великі компанії на роботу зі спільнотою виділяють дуже великі гроші, тому що ці витрати набагато ефективніші, ніж витрати на залучення нових людей.

Ви створюєте співтовариство на одній грі, розвиваєте у ньому лояльність до вашої студії і отримуєте аудиторію для своєї іншої гри. Крім того, в спільноті перебувають найвідданіші ваші користувачі, які завжди беруть участь у тестуванні.

Єдиний випадок, коли не потрібен КМ – коли виконується розробка одноразової гри.

Якщо ви працюєте з видавцем, то на його боці, швидше за все, буде цілий відділ, який буде займатися роботою з спільнотою і розвитком вашої гри.

Опціональні фахівці є далеко не на кожному проекті. Вони бажані, але без них, у принципі, можна обійтися.

СПЕЦІАЛІСТ ПО ЗВУКУ

Для невеликих команд така людина – невиправдана і занадто велика витрата. Маленьким компаніям рекомендується використовувати очищену від прав на комерційне використання музику. Щодня ігри знімаються з доступу через використання музики.

Якщо ви купуєте музику і звуки, стежте дуже уважно за тим, яку ліцензію ви купуєте, і чи є там право на комерційне використання. Така ліцензія буде коштувати дорожче.

Якщо ви використовуєте звуки і музику, написані спеціально для вашого проекту, але на аутсорсингу, зверніть увагу на те, як оформлено передачу прав. Трапляється так, що аутсорсер продає музику відразу в кілька ігор, і, поки компанії судяться між собою, їхні ігри не продаються.

АНАЛІТИК

Бувають компанії, які вважають, що аналітики – це люди з відділу маркетингу, що займаються аналізом даних по вашій грі. Є люди, які вважають, що це, швидше за все, люди з відділу тестування, які дивляться, що відбувається на ринку.

Для маленьких компаній це дуже опціональний фахівець. Але він стає незамінним, якщо ви намагаєтеся зробити щось інноваційне і вийти на новий ринок. Тому перш ніж це робити, потрібно провести дослідження того, що на цьому ринку відбувається, які у вас конкуренти, потенційні проблеми і т. ін.

Аналітик – відмінний варіант для поєднання з маркетологом або тестувальником – в залежності від тієї сфери, де знадобиться аналітика на проекті.

Крім того, якщо у вас на проекті є людина, у якої сьогодні і завтра немає завдань, вона може не просто сидіти і байдикувати, а піти пограти в ігри ваших конкурентів і написати по них звіт. Найчастіше це буває дуже корисно і практично у всіх компаніях використовується такий підхід.

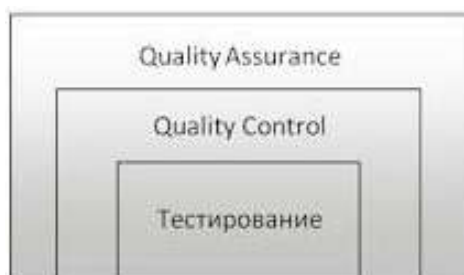
ТЕХНІЧНИЙ ДИРЕКТОР

Як правило, це дуже просунутий розробник, але при цьому технічний директор зовсім необов'язково повинен бути лідом. Йому досить визначити технології і стек розробки для проекту, і не потрібно займатися прямим керівництвом іншими програмістами.

Якщо в проекті є вже 3 програміста, виділіть одного з них в якості технічного директора, щоб він поставив стандарти для інших, тому що інакше всі будуть працювати як їм зручно.

QA/QC

Quality Assurance (QA) забезпечує правильність і передбачуваність процесу, в той час як Quality Control (QC) передбачає контроль дотримання вимог. Тестування ж, у свою чергу, забезпечує збір статистичних даних і внесення їх в документи, створені в рамках QC-процесу.



QA – превентивний процес, завданням якого є забезпечення якості продукту в майбутньому. QA орієнтований на процес. Охоплює всі етапи розробки, включаючи експлуатацію та реліз.

QA – Quality Control – Контроль якості.

QC – процес знаходження помилок у ПЗ з метою їх усунення. Завдання QC – підтримка якості продукту в поточний момент часу.

QC – це частина QA.

QA/QC-фахівці, про яких часто забувають до етапу, коли потрібно показувати гру людям і з'ясовується, що в ній дуже багато багів. У маленьких командах тестуванням зазвичай займаються всі, але більшою мірою – продюсер і геймдизайнер у вільний від основних завдань час. Якщо у вас є можливість залучити гравців із спільноти, це заощадить вам багато часу і грошей.

QC – це ще один з варіантів, з якого можна почати кар'єру без досвіду. Тестувальник може дійти у своїй кар'єрі до QA-ліда.

Часто використовують зовнішніх QA або QC, є досить багато студій, які на цьому спеціалізуються. Якщо у вас немає можливості тримати людину в штаті, ви можете скористатися послугами цих студій. Цінник у них не дуже високий, а виловивши і виправивши всі баги до релізу, ви точно заощадите набагато більше.

БІЗДЕВ

Business development, біздев (англ. – розробка/розвиток бізнесу) – це створення для організації довготривалої цінності, що формується клієнтами, ринками і зв'язками. Процес розробки бізнесу можна порівняти з вибором правильних технічних рішень при написанні ядра програмного продукту.

В основному біздев – це людина, що займається стратегією і роботою з партнерами. У невеликих компаніях зустрічається рідко. Виникає в тому випадку, коли у компанії є власник, який дає гроші, у нього є ідея, але при цьому у розробку він лізти не хоче. У цьому випадку його можна назвати біздевом.

PR

PR-фахівець – це менеджер зі зв'язків з громадськістю (саме так розшифровується термін PR – Public Relations). Це фахівець, що займається формуванням і підтриманням позитивної громадської думки про компанію, розроблені продукти, пов'язані з проектом події. Найчастіше поєднується з КМ. У багатьох компаніях піарників відправляють у відділ маркетингу.

У світових компаніях це люди, що працюють над іміджем проекту і команди. Це робота зі ЗМІ, сайтами, блогерами, виставками, конференціями і так далі.

PR як маркетинг: не потрібно чекати запуску, щоб почати роботу в цьому напрямку. Піарити проект потрібно якомога раніше, тому що маховик розкручується не за один день. Як тільки з'являються перші арти і концепт, можна починати роботу піарника.

HR

Human Resources (HR) – «людські ресурси». HR-менеджер – це професіонал, чиєю сферою діяльності є підбір персоналу, діловодство і управління кадровим потенціалом компанії.

HR-фахівці, що займаються пошуком нових людей в вашу команду, а також ведуть звітність і документацію по вже існуючих людях. Крім того, HR має займатися мотивацією і адаптацією співробітників.

Часта помилка: в команду шукають нову людину, наприклад, розробника. Проводить співбесіду ваш другий розробник і каже, що з навичками все відмінно. А на психологічну сумісність з командою його ніхто не перевіряв, і може вийти так, що ви витратите гроші і час на нову людину, яка не впишеться в колектив.

Якщо в команді є осіб десять, то HR вже необхідний. Хоча б зовнішній, щоб він вів договір і всю документацію. Але якщо приділяється велика увага атмосфері в команді, потрібен власний HR.

ПРИБЛИЗНЕ СПІВВІДНОШЕННЯ ЗАРПЛАТ ФАХІВЦІВ:

№	Спеціаліст	Середня зарплата (\$)	
		центр	регіони
1	Продюсер	2627	1313
2	Менеджер проекту / Scrum Master	2276	1 751
3	Розробник	2276	1313
4	Гейм-дизайнер	1401	876
5	Арт-директор / дизайн-лідер	1576	1051
6	Маркетолог	1576	700
7	Ком'юніті-менеджер (КМ) / спец. техпідтримки	1401	700
8	Фахівець по звуку	1576	1576
9	Аналітик	2627	1313
10	Технічний директор	2627	1576
11	Qa / qc	876	700
12	Бізнес-девелопер	2276	1 751
13	Фахівець з PR	1576	700
14	Фахівець з HR	1576	700

3.3 ОПТИМАЛЬНИЙ СКЛАД І МАСШТАБОВАНІСТЬ КОМАНДИ

ОПТИМАЛЬНИЙ СКЛАД КОМАНДИ

Проаналізуємо оптимальний склад маленької команди, яка зібралася для реалізації невеликого проекту.

Будь-яка команда починається з кістяка – без нього ви не зможете зробити проект. Як правило, це чотири людини. Управлінець: продюсер, ПМ або біздев, розробник, геймдизайнер і художник або дизайнер.

Бувають рідкісні винятки, коли геймдизайнер виконує ще й функцію керівника. Додаткових людей потрібно наймати, ґрунтуючись на вашому проектному плані. Якщо за вашим планом виходить, що завантаження у програміста на півроку, а у художника на рік, значить, потрібно найняти другого художника.

Для невеликої гри потрібно закладати на розробку 4 місяці з буфером у 2 місяці.

Якщо ви бачите, що ваш проект за складністю сильно виходить за ці рамки, ви можете повернути його в ці рамки за допомогою збільшення кількості людей.

Не всі фахівці потрібні на всіх етапах проекту, тому потрібно заздалегідь вирішити, що ви будете робити з цими людьми. Якщо ви не можете перевести співробітника на інший проект, подумайте: які завдання він буде вирішувати, коли виконає свої основні задачі. Можна або спочатку наймати людину під конкретні завдання: і після їх виконання, з ним розлучатися. А можна використовувати його для якихось інших завдань.

Важливий момент: як тільки якихось фахівців стає більше двох, потрібно відразу ж виділяти з них Лідера, інакше ви будете сильно втрачати в ефективності.

Коли їх тільки двоє, це робити не обов'язково, вони будуть прекрасно працювати в парі.

Дуже важливо стежити за балансом фахівців, щоб не вийшло так, що у вас вся команда чекає одну людину або навпаки – у когось просто немає завдань.

МАСШТАБУВАННЯ КОМАНДИ

Може виникнути ситуація, що ваша команда не справляється. Наприклад, змінилися вимоги і видавець попросив додати нові фічі. У цей момент виникає необхідність збільшити кількість людей в команді.

Практично неможливо знайти людину швидко, це займе якийсь час. Навіть якщо вам пощастить і ви знайдете її відразу, потрібен час на адаптацію. Тому починати пошук потрібно заздалегідь, за 2-3 місяці.

Якщо проект масштабується, все добре, і ви просто хочете робити більше контенту, то краще беріть молодих людей. Це дасть можливість заощадити і «прокачати» людей собі ж в команду. Якщо ж ви розширювали команду через те, що горять терміни, то брати потрібно тільки досвідчених людей, тому що молоді будуть довше входити в курс справи.

Нова людина має комфортно увійти у вашу вже сформовану команду.

ВИБІР ТИПУ КОМАНДИ

Що вибрати: In-House vs Розподілена команда

Є 2 основних типи команд: In-House і Розподілена команда.

Члени In-House-команди повинні сидіти разом в одному приміщенні. У розподіленій команді люди працюють над одним проектом з різних міст. Обидва ці підходи працюють успішно і не потрібно вважати, що, наприклад, розподілена команда – це погано. Але кожен варіант має свої плюси і мінуси, які обов'язково потрібно враховувати в роботі.

1. In-House

Основна перевага – швидкість комунікації. Можна підійти до іншої людини і щось з нею обговорити.

З цього плюса виростає неоднозначний пункт: в команді з'являється багато неформального спілкування. З одного боку, це добре, коли люди в команді дружать і

спілкуються, але з іншого, якщо в команді виникають якісь проблеми, то одна людина може заразити панікою всю команду буквально за півгодини.

У In-House-командах краще виконується контроль і управління командою і процесами.

Недоліки:

- потрібен офіс, де всі будуть збиратися – це додаткові витрати;
- потрібно всіх оформляти, це теж вимагає витрат.

2. Розподілена команда

Зазвичай люди, що працюють віддалено, мають індивідуальний простір і інструментарій, і витрати підвищуються не суттєво.

Крім того, люди, що працюють віддалено, більш точні в оцінках і прогнозах. Перша вимога до віддаленого фахівця – вміння оцінювати свою роботу. Вони отримують не фіксовану ставку, а за кількістю витрачених годин.

Для In-House команди важливіше психологічні вимоги, а для розподіленої команди – дотримання процесів.

РОБОТА З АУТСОРСИНГОМ

Більшість команд використовують аутсорс в будь-якому вигляді. Його можна використовувати в артi, в тестуванні, іноді можна отримати якусь частину коду за допомогою аутсорсерів.

Основні ризики в роботі з аутсорсерами:

- ви не особливо контролюєте людину, тому складно бути впевненим, що вона все зробить вчасно і з нормальною якістю;
- досить складно перевірити реальний рівень людини; у художника в портфоліо можуть лежати одні картинки, а для вашого проекту він буде малювати зовсім інші;
- складно перевірити, чи не малює він одночасно для декількох проектів;
- крадіжка інформації; уважно стежте за доступами; в маленьких студіях часто-густо зустрічається плутанина з документами; якщо людина забере всі ваші вихідні дані, є шанс, що ваша гра вийде раніше і не у вашій студії;
- авторське право; не потрібно закривати на це очі, оформлюйте все грамотно;
- всю оплату потрібно чітко обумовити і прописати; всі переробки і ризики мають бути зафіксовані;
- якщо у вас є якийсь позитивний досвід роботи з людиною, потрібно максимально його продовжувати, але при цьому не варто всю роботу замикаати тільки на ній, фахівець може виявитися зайнятим наступного разу.

3.4 УПРАВЛІННЯ КОМАНДОЮ

В ІТ використовуються два основні підходи до управління командою – Agile і Waterfall. Кожен з них має свої характеристики, особливості, переваги і недоліки.

МЕТОДОЛОГІЯ AGILE

Гнучка методологія розробки (Agile software development, agile-методи) – серія підходів

до розробки програмного забезпечення, орієнтованих на використання ітеративної розробки, динамічне формування вимог і забезпечення їх реалізації в результаті постійної взаємодії всередині самоорганізованих робочих груп, що складаються з фахівців різного профілю. Існує кілька методик, що належать до класу гнучких методологій розробки, зокрема екстремальне програмування, DSDM, Scrum, FDD.

Agile застосовується як ефективна практика організації праці невеликих груп (які роблять однорідну творчу роботу) в поєднанні з керуванням ними комбінованим (ліберальним і демократичним) методом.

Більшість гнучких методологій націлені на мінімізацію ризиків шляхом зведення розробки до серії коротких циклів, званих ітераціями, які зазвичай тривають два-три тижні. Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі і включає всі завдання, необхідні для видачі міні-проекту щодо функціональності: планування, аналіз вимог, проектування, програмування, тестування і документування. Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі, що гнучкий програмний проект є готовим до випуску по закінченню кожної ітерації. Після закінчення кожної ітерації команда виконує переоцінку пріоритетів розробки.

Agile-методи роблять упор на безпосереднє спілкування віч-на-віч. Більшість agile-команд розташовані в одному офісі, іноді званому bullpen. Як мінімум, вона включає і «замовників» (product owner – замовник або його повноважний представник, який визначає вимоги до продукту; цю роль може виконувати менеджер проекту, бізнес-аналітик або клієнт). Офіс може також включати тестувальників, дизайнерів інтерфейсу, технічних письменників і менеджерів.

Основною метрикою agile-методів є робочий продукт. Надаючи перевагу безпосередньому спілкуванню, agile-методи зменшують обсяг письмової документації в порівнянні з іншими методами. Це призвело до критики цих методів як недисциплінованих.

В рамках методології або філософії Agile існує багато конкретних методик управління. Вони трохи відрізняються, але всі об'єднані ключовими особливостями:

- постійна робота команди з замовником або з користувачами;
- отримання зворотнього зв'язку на кожному етапі;
- після кожної ітерації демонструється продукт, і після отримання зворотнього зв'язку вносяться зміни;
- результатом кожної ітерації обов'язково є продукт, що працює.

Agile як маніфест оформився в 2001 році, але на практиці існує приблизно з 70-х років. Коли сформувався «Маніфест Agile» було вироблено 4 цінності. Якщо вони близькі, вашій команді, ви можете працювати по Agile.

Цінності:

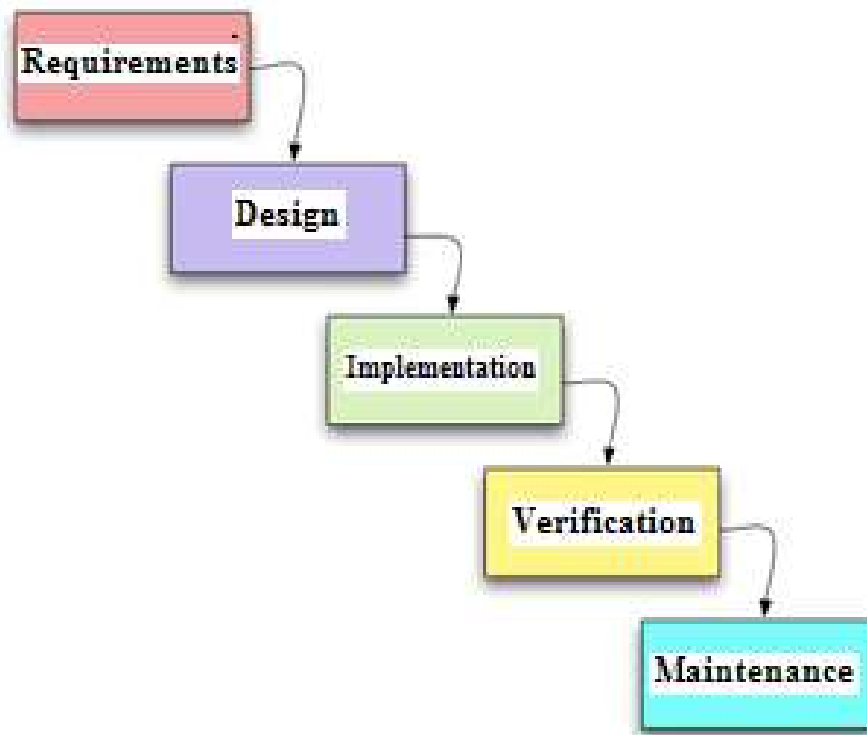
- індивідуальність і ітеративність завжди важливіше процесів та інструментів; під індивідуальністю мається на увазі рішення кожної людини;
- ПЗ, що працює, важливіше за документацію;
- постійна робота в контакт з замовником / користувачем;

- внесення змін «на льоту» важливіше проходження плану.
- Основна проблема в Agile – дотримання бюджету.

МЕТОДОЛОГІЯ WATERFALL

Каскадна модель (Waterfall model, модель «Водоспад») – модель процесу розробки ПЗ, у якій процес розробки виглядає як потік, що послідовно проходить фази аналізу вимог, проектування, реалізації, тестування, інтеграції та підтримки. Як джерело назви часто вказують статтю, опубліковану У.У.Ройсом (WW Royce) в 1970р. при тому, що сам Ройс використовував ітеративну модель розробки. Підхід Waterfall засновано на послідовному циклі розробки. Спочатку – якісне планування, потім – робота за планом.

У вихідній каскадній моделі наступні фази йшли у такому порядку:



- 1) визначення вимог;
- 2) проектування;
- 3) конструювання (реалізація або кодування);
- 4) втілення;
- 5) тестування та налагодження («верифікація»);
- 6) інсталяція;
- 7) підтримка.

Перехід від однієї фази до іншої відбувається тільки після повного і успішного завершення попередньої.

Дотримуючись каскадній моделі, розробник переходить від однієї стадії до іншої строго послідовно. Спочатку повністю завершується етап «визначення вимог», у результаті чого виходить список вимог до ПЗ. Після того як вимоги повністю визначено,

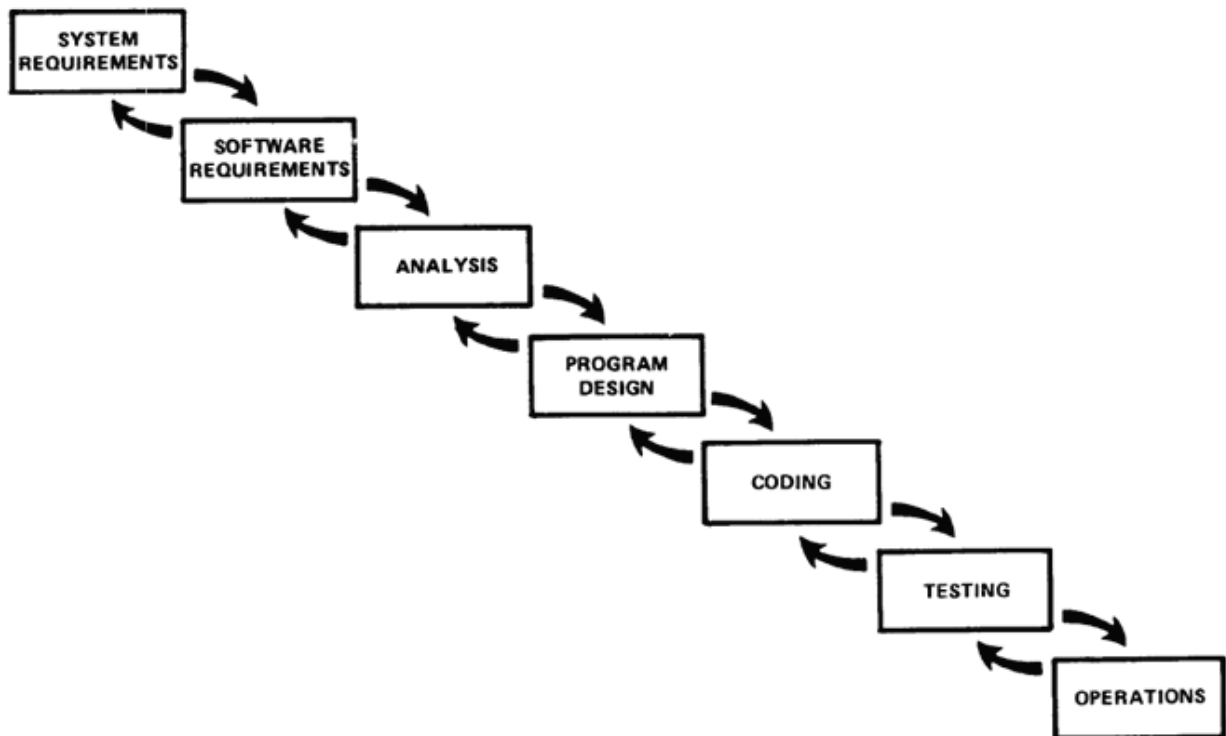
відбувається перехід до проектування, в ході якого створюються документи, що детально описують для програмістів спосіб і план реалізації зазначених вимог. Після того, як проектування повністю виконано, програмістами виконується реалізація отриманого проекту. На наступній стадії процесу відбувається інтеграція окремих компонентів, що розробляються різними командами програмістів. Після того, як реалізацію і інтеграцію завершено, проводиться тестування і налагодження продукту; на цій стадії усуваються всі недоліки, що з'явилися на попередніх стадіях розробки.

Таким чином, каскадна модель має на увазі, що перехід від однієї фази розробки до іншої відбувається тільки після повного і успішного завершення попередньої фази, і що переходів назад або вперед, або перекриття фаз не відбувається.

У 1970 році в своїй статті Ройс описав у вигляді концепції те, що зараз прийнято називати «каскадна модель», і обговорював недоліки цієї моделі. Там же він показав як цю модель може бути доопрацьовано до ітеративної моделі.

Проте існують модифіковані каскадні моделі (включаючи модель самого Ройса), що мають невеликі або навіть значні варіації описаного процесу. Ройс передбачав зворотні зв'язки між етапами на одному рівні (наприклад, дизайн-кодування, кодування-тестування і т.д.).

У другому варіанті, на відміну від першого, мова йде про паралельні роботи по двох послідовних етапах, що дає можливість на більш ранніх стадіях виявити помилки попереднього етапу життєвого циклу і усуває один з вагомих недоліків «класичного» водоспаду – неможливість повернення на попередній етап.



Наприклад, якщо розглянути паралельні роботи по 2 послідовних фазах – Coding і Testing, стає очевидним, що частина програми видається в тестування, в той час як

інша частина все ще перебуває на стадії розробки.

Тобто виходить, що мова дійсно йде про ітеративну методологію розробки ПЗ.

Зараз компанії, що працюють за цією системою, використовують ітераційний Waterfall. Всі ці стадії розподілені на якісь невеликі ітерації, наприклад, на 2 тижні.

Всередині команди у вас може бути будь-який підхід, але як тільки ви як продюсер починаєте працювати з видавцем або інвестором, без знання Waterfall обійтися ніяк не можна.

	Agile	Waterfall
переваги	гнучкість; легкість внесення змін; розвиток продукту під час розробки	послідовність; чіткі вимоги; жорсткі процеси
коли використовувати	немає чітких вимог до кінцевого продукту; часте внесення змін замовниками; ключовим завданням є швидке отримання працюючого продукту	сформульовані чіткі вимоги до кінцевого продукту; замовники чітко знають, чого хочуть, виконується робота над продуктом, який вже розроблявся раніше; робота з фіксованою оплатою за результатом і строго обмеженим бюджетом

SCRUM

Найвідоміша методологія в рамках Agile – це SCRUM. Вона використовує всі ключові особливості Agile і вводить кілька своїх речей.

Над проектом працює одна універсальна команда. У ній стільки різнопланових фахівців, скільки потрібно для вирішення будь-якої задачі проекту.

Оскільки команда самоорганізується, у фахівців scrum-команди немає формальної компетенції. Коли необхідно, тестувальник допомагає дизайнеру, а аналітик – розробнику.

У scrum-команді, крім власне фахівців, є дві ролі.

Scrum-майстер – людина, що організовує роботу. Це не управлінська посада, і фахівець не роздає вказівки. Завдання scrum-майстера:

- вести збори;
- усувати перешкоди в роботі (якщо команді заважає перфоратор в сусідньому офісі, майстер шукає вихід);
- помічати і витягувати на поверхню приховані проблеми;
- відповідати за дотримання методології;
- стежити за статусом завдань.

У вільний від цих завдань час скрам-майстер працює так само, як інші члени команди.

Власник продукту – product owner – визначає хід проекту, він може представляти зовнішнього замовника. Власник знає все про ринок і цільову аудиторію. Він ставить пріоритети завданням. Результат роботи команда представляє власнику продукту.

У SCRUM ітерації називаються спринтами і вводяться 4 обов'язкових церемонії.

- 1) планування спринту;
- 2) щоденний стендап;
- 3) демо за підсумками спринту;
- 4) ретроспектива за підсумками спринту.

Також SCRUM визначає ролі. Є дві ключові:

- Product Owner – визначає бачення продукту. З ним пов'язані основні ризики, це має бути порядна людина з правильним баченням;
- SCRUM Master – підтримує процеси і церемонії в команді.

SCRUM відмінно підходить для невеликих команд з 5-7 чоловік.

KANBAN

Kanban – японський термін, який почали використовувати стосовно виробництва у 60-х роках 20-го століття в компанії Toyota. В основу даного принципу покладено конвеєрний метод виробництва, а також різні швидкості виконання окремих технологічних операцій на виробництві. При будь-якому виробництві є основне виробництво («головний конвеєр») і додаткове виробництво («додаткові конвеєри»). Темп випуску кінцевих виробів задає головний конвеєр, в той час як додаткові конвеєри не можуть прискорити темп випуску виробу, але можуть сповільнити його, в разі несвоєчасного випуску необхідних деталей.

Додатково при виробництві може відбутися зміна пріоритетів. Наприклад, з'ясувалося що станція, яка виробляла ліві дзеркала зробила 20 шт., а станція, що виробляла праві дзеркала – 10 шт., В той час як на конвеєрі знаходяться 15 автомобілів і необхідно 15 штук дзеркал обох типів. У наявності конфлікт метрики – кількісно виробництво не впало (додаткові конвеєри випустили 30 виробів в термін), але виробництво все одно ризикує зупиниться. Kanban призначений допомогти з цією проблемою.

У спрощеному варіанті, Kanban включає в себе два прості правила:

- виробнича станція має план виробництва деталей («backlog»); план здійснюється для досягнення цього, і може змінюватися в будь-який час (наприклад станція виробляє занадто багато лівих дзеркал і повинна мати можливість переключитися на праві якомога швидше);
- кількість завдань, що виконуються на станції одночасно, обмежена (тобто виробляти не більше заданої кількості дзеркал, одночасно); це обмеження необхідно для управління швидкістю виробництва на станції, а також швидкістю реагування на зміни плану.

Останнім часом Kanban набирає велику популярність у виробництві програмного забезпечення. Деякі команди вважають цю методологію виключно корисною, деякі використовують за принципом «культу Карго». Чистий Kanban погано працює для продуктових команд (читай – «основний конвеєр»), але відмінно працює з командами підтримки, такими як:

- групи підтримки програмного забезпечення, де не важливий «план», але важлива швидкість реагування на зміни;

- групи тестування, що працюють окремо від груп розробки;
- служби підтримки;
- інші приклади «неосновних виробництв».

Канбан добре працює в стартапах, що не мають чіткого плану, але активно працюють над розробкою.

Приклад: використання Канбан в розробці невеликого проекту командою з одного розробника.

План розробки (backlog) відсортований в порядку пріоритету частин роботи, ліміт команди на завдання в процесі – 1 шт.

Для управління процесом керівник проекту може:

- 1) змінити ліміт на кількість завдань в роботі;
- 2) додати завдання з більш високим пріоритетом (наприклад р0) для того щоб воно було взято якомога швидше.

В процесі роботи може так статися, що роботу заблоковано (зламався хостинг, ще не викачаний потрібний framework і т.д.). У загальному випадку, заблокована робота повертається в backlog, і вибирається нове завдання, з максимальним пріоритетом. Залежно від характеру завдань і типу команди, ліміт може бути збільшений або зменшений. Наприклад, наш розробник може одночасно малювати форму реєстрації та дивитися за процесом розгортання нового сервера. Проте, якщо час завершення завдань буде менше необхідного, керівник проекту може зменшити ліміт, або збільшити команду. Таким чином, при грамотному керівництві, Канбан забезпечує максимально можливу для даної команди швидкість роботи, максимальну швидкість реагування на зміни і в той же час скорочує «витрати» на підтримку методології.

До обмежень Канбан при використанні його в продуктивних командах можна віднести наступні:

- дана методологія погано працює з великими командами (більше 5 осіб);
- Канбан в чистому вигляді погано працює з крос-функціональними командами, тобто на відміну від Scrum, важко поєднати тестування і розробку в одній команді: більш вдалою думкою є розбити процес на «станцію» розробки і «станцію» тестування з окремими керівниками і backlog-ами;
- зважаючи на свою історію і специфіку, Канбан не призначений для довгострокового планування.

Над завданням може працювати кілька вузькопрофільних команд. Наприклад, спочатку працюють аналітики, потім дизайнери малюють прототип, а на третьому етапі включаються розробники. При цьому універсальні команди не заборонені. У Канбан всередині команди немає ролей.

Переваги використання Канбан в Agile-командах:

- підвищується гнучкість;
- знижуються витрати;
- простота використання;

- економія часу;
- елементарна організація процесів.

Kanban – корисний інструмент, який використовується в будь-яких гнучких методологіях. Існує багато електронних варіантів. Це найпростіший спосіб почати організацію в невеликому проекті, що дає відчутний приріст ефективності.

3.5 РОБОТА НАД ПРОЕКТОМ

СКЛАДАННЯ СПИСКУ ЗАВДАНЬ

Для початку команда бере проект і ділить його на десятки, а то й сотні завдань поменше. Це частина філософії Agile, тому так роблять і в Kanban, і в Scrum.

Всі завдання проекту, які належить виконати, складають у загальний список – беклог. Беклог – це банк завдань проекту. Кожне завдання має бути актуальним. Якщо буде потрібно, їх можна додавати в беклог або видаляти з нього «на льоту».

Кожне завдання має вагу – зазвичай це час, що потрібний на рішення. Команда сама оцінює вагу всіх завдань, тому якщо проект не закінчено в задані терміни, винна команда.

У кожного завдання є пріоритет. У Kanban пріоритети розставляє команда, в Scrum – власник продукту. Пріоритети можна і навіть потрібно переглядати по ходу проекту – це один із стовпів гнучких методологій.

РОБОТА НАД ПРОЕКТОМ

Коли починається робота Scrum, стає зрозуміло, чому його називають набагато більш директивною методологією.

Scrum-команда розбиває час роботи над проектом на рівні відрізки – спринти. Спринт може тривати і день, і місяць, а в останні роки стандартом став спринт у 2 тижні.

Оскільки всі спринти однакові за тривалістю, в роботі команди з'являється ритм. Ритм – важливий аспект методології.

Спринт складається з чотирьох послідовних етапів.

- Планування. Команда перевіряє завдання з беклога і вибирає найбільш пріоритетні. На спринт беруть стільки завдань, скільки встигнуть зробити.
- Виконання. В ідеальній команді фахівці працюють паралельно: поки програміст створює код, тестувальник пише до нього тести, а технічний письменник – документацію.
- Реліз. Команда представляє результати своєї роботи світу. До моменту кожного релізу продукт повинен бути працездатним, корисним для користувача і більш досконалим, ніж до спринту.
- Ретроспектива. Команда обговорює спринт і проблеми, що виникли. Всі разом думають, як поліпшити роботу і зробити в наступному спринті більше.

У Scrum категорично не можна додавати завдання в поточний спринт, тому Scrum менш гнучкий. Навіть якщо з'явилася термінова і важлива задача, вона піде в роботу тільки з наступного спринту.

В кінці спринту недороблені завдання йдуть назад у беклог. Чи потрібно їх доробляти і коли – визначають на етапі планування наступного спринту.

У порівнянні зі Scrum-директивами Kanban – це лібералізм і хаос.

Спринтів як таких немає. Проект зазвичай ділять на ітерації, але вони можуть бути будь-якої довжини. Ритмічності Kanban не потребує.

Етапи scrum-спринту:

- планування;
- виконання;
- реліз;
- ретроспектива.

У Kanban одні етапи не зав'язані на інші. Настають вони, коли вирішить команда. Наприклад, реліз – по вівторках, планування – коли закінчилися завдання, ретроспективи – кожен останній четвер місяця, а на тлі всього цього безперервно йде розробка.

Виражених спринтів немає, з'являються особливості:

- нові завдання додають в будь-який час. Якщо потрібно терміново щось зробити, команда не чекає наступного спринту;
- завдання залишається в роботі скільки завгодно довго, поки команда не закінчить його або не скасує.

ВИКОРИСТАННЯ ДОЩОК

Дошка – це серце Kanban – і Scrum-розробки, єдиний візуальний атрибут методологій. Дошки насамперед вішають на стіну, коли хочуть показати, що працюють по Kanban або Scrum.

Дошку використовують, щоб зробити проект більш прозорим, розпланувати завдання і поставити обмеження.

Дошку розкреслюють на стовпці. Кожен стовпець – це стан завдання («Розробка», «Тестування», «Реліз»). Кількість стовпців залежить від проекту, але чим їх менше, тим краще.

Картки – це завдання. На кожній опис, вага і пріоритет. Коли завдання проходить черговий етап, її переклеюють у відповідний стовпець. При простому погляді на дошку зрозуміло, як просуваються справи з проектом в цілому і з кожним завданням.

Дошки бувають фізичні і електронні.

Фізична дошка. Часто дошка – це дійсно дошка з клітинами з малярного скотча. Завдання – липкі листочки, які зручно рухати по дошці.

Справжня дошка стоїть у кімнаті і кожен в команді бачить, як йдуть справи. Навколо цієї дошки добре збиратися під час нарад.

Електронна дошка. Електронна дошка під рукою на пляжі, на дачі, в машині і в метро. Якщо у вас є співробітники, які працюють віддалено, електронна дошка – єдиний вихід.

ДОТРИМАННЯ ОБМЕЖЕНЬ

У Scrum число завдань, які одночасно перебувають у роботі, обмежено їх загальною вагою. Якщо відомо, що команда робить за спринт 26 умовних одиниць, значить загальна вага завдань на наступний спринт не може перевищувати 26.

У Kanban число активних завдань обмежено їх вагою по кожному статусу окремо. При цьому неважливо, яка загальна вага завдань на дошці.

На kanban-дошці цифра поруч з назвою стовпця – це обмеження. Якщо для простоти припустити, що всі завдання приблизно рівні за трудомісткістю, то цифра позначає, скільки завдань може бути в колонці одночасно.

Колонки можна дробити на більш дрібні, щоб процес роботи був більш прозорим. При цьому обмеження накладають на батьківську колонку.

Обмеження колонки «Розробка» – не більше трьох завдань. Якщо колонка заповнена, то програмісти не можуть взяти нову роботу з беклога.

Якщо інтегратори завалені роботою, то не можуть взяти у розробників готові завдання. Якщо розробники візьмуть нову роботу, пробка стане ще щільніше. Тому в колонці «Розробка» обмеження для поточних і готових завдань загальне.

Kanban вказує, що в такій ситуації розробники переміщаються до інтеграторів і разом з ними розчищають завал. Тільки коли в «Інтеграції» з'явиться місце, розробники проштовхнуть туди завдання з «Готово» і візьмуть нове.

В результаті команда домагається не максимального числа незавершених завдань, а максимальної швидкості проходження завдання по дошці. Якщо говорити про продукт, швидкість появи нових можливостей важливіша за число можливостей в розробці.

Налаштування обмежень – найважливіша частина роботи по Kanban.

Припустимо, люди простоюють і не можуть взяти нові завдання, тому що колонка забита. В цьому випадку потрібно дивитися, що відбувається праворуч. Якщо справа заторів немає, значить, обмеження колонки занадто суворе.

Інший варіант – завдання простоюють, і середній час виконання знижується. Так буває, коли на дошці занадто багато завдань, і за них нікому взятися. Якщо помітили симптом, робіть суворіше обмеження колонок.

Kanban радить експериментувати. Поміняли обмеження на колонку – подивилися, що вийшло. І так весь час.

ВИМІРЮВАННЯ ПОКАЗНИКІВ

У Scrum вимірюють загальну вагу завдань, виконаних за спринт. Розділивши загальну вагу всіх завдань проекту на продуктивність за спринт, ми отримаємо приблизний термін закінчення проекту. Завдання команди в Scrum – підвищення продуктивності.

У Kanban вимірюють середній час проходження завдання по дошці. Цей час – показник ефективності команди. Показник дає цікавий ефект: команда не концентрується на виконанні конкретних завдань. Вона просто стежить, щоб середній час виконання був мінімальним.



Лабораторні роботи

ВПРОВАДЖЕННЯ AGILE

Перейти до Agile простіше через Kanban. У цій методології менше обмежень, можна не формувати крос-функціональні команди, а як і раніше поділяється на відділи. Scrum можна спробувати, коли відчуєш суть гнучкої методології.

- 1) Знайдіть настільну гру getkanban і пограйте в неї з колегами. Офіційна версія коштує \$ 450, але в інтернеті є й інші варіанти.
- 2) Якщо гра сподобалася, і ви відчули методологію, візьміть будь-який проект і поділіть його на дрібні завдання. Зручно, коли всі завдання приблизно рівні за часом виконання. Призначте пріоритет для кожного.
- 3) Поміркуйте, які стадії проходить будь-яке завдання. У найпростішому варіанті це може бути «Заплановано → В роботі → Завершено». Варіант складніше: «Заплановано → Дизайн → Розробка → Інтеграція → Готово». Статуси залежать від того, що саме ви робите.
- 4) Проставте обмеження на статус завдань так, як підказує логіка. Якщо у вас два дизайнери і три програмісти, нехай в колонці дизайну буде максимум 2 завдання, а на дошці «В розробці» – 3.
- 5) Проведіть першу нараду. Обговоріть, як організуєте роботу. Коли будуть релізи продукту, а коли збиратися на наради.
- 6) Перенесіть кілька завдань з беклога в «Заплановано» і запускайте процес.
- 7) Слідкуйте, щоб картки подорожували між колонками, коли завдання перейшло з одного статусу в інший.
- 8) Зміряйте середній час виконання завдання. Додаючи картку на дошку, пишiть на ній час початку роботи, а знімаючи – час завершення.
- 9) Експериментуйте з обмеженнями і робочим процесом так, щоб середній час виконання завдання зменшувався.
- 10) Збирайтеся на ретроспективні зібрання, де ви обговорюєте роботу над проектом: що пройшло добре, що виходить погано, як можна поліпшити роботу.

ЛАБОРАТОРНА РОБОТА №1 ВИВЧЕННЯ ІНТЕРФЕЙСУ CLICKTEAM FUSION

МЕТА РОБОТИ:

- знайомство з інтерфейсом додатка Clickteam Fusion;
- вивчення основних можливостей Clickteam Fusion;
- створення першої міні-ігри.

ПОРЯДОК РОБОТИ

1. Вивчити інструкцію.
2. Відкрити програми редактора Clickteam Fusion.
3. Створити новий проект.
4. Ознайомитися з інтерфейсом Clickteam Fusion. Виконати переходи у вікна наявних редакторів.
5. Зберегти новий порожній проект.
6. Створити об'єкти згідно з прикладом. Задати властивості об'єктів.
7. Задати параметри руху і поведінки об'єктів.
8. Перевірити працездатність проекту.
9. Дослідити можливі варіанти руху об'єктів, змінюючи відповідні їх властивості.
10. Зберегти створений проект.
11. Закрити додаток Clickteam Fusion.
12. Знову відкрити додаток Clickteam Fusion, завантажити свій проект і перевірити його. Продемонструвати працездатність проекту викладачеві.
13. Оформити звіт. Захистити роботу.

РОЗРОБКА ІГОР У ГРАФІЧНОМУ РЕДАКТОРІ CLICKTEAM FUSION

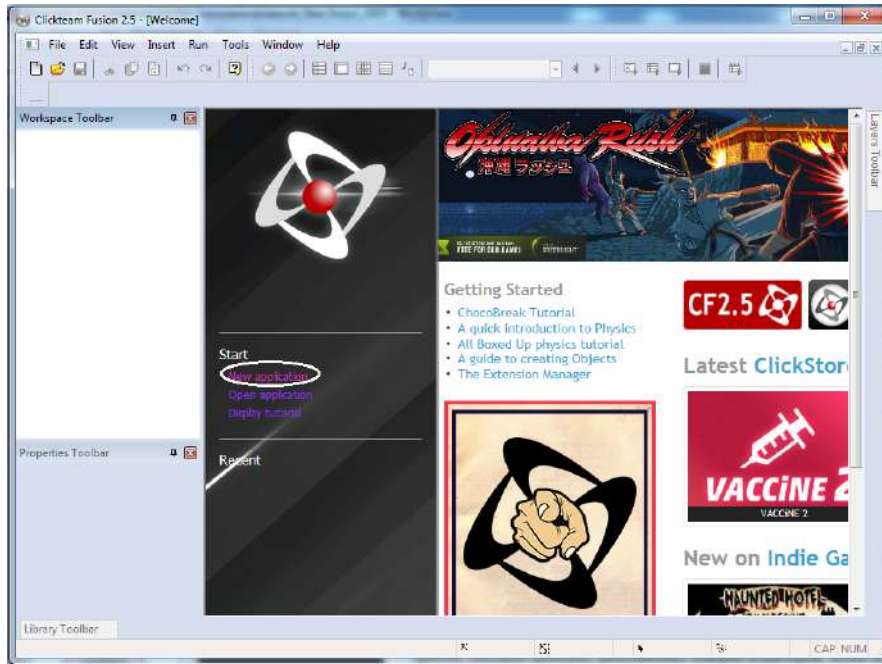
Редактор дозволяє швидко створити прототип гри, що може бути корисно сценаристам, художникам і геймдизайнерам, але головне – написати свою власну гру і опублікувати її в Steam, Google Play, Yandex.Store, Amazon.com, AppStore, Windows Phone Store та інших магазинах. У створеній грі можна здійснювати внутрішньоігрові платежі, а також розміщувати рекламу.

1 ІНТЕРФЕЙС ПРОГРАМИ CLICKTEAM FUSION

Запуск програми Clickteam Fusion виконується після виконання команд:

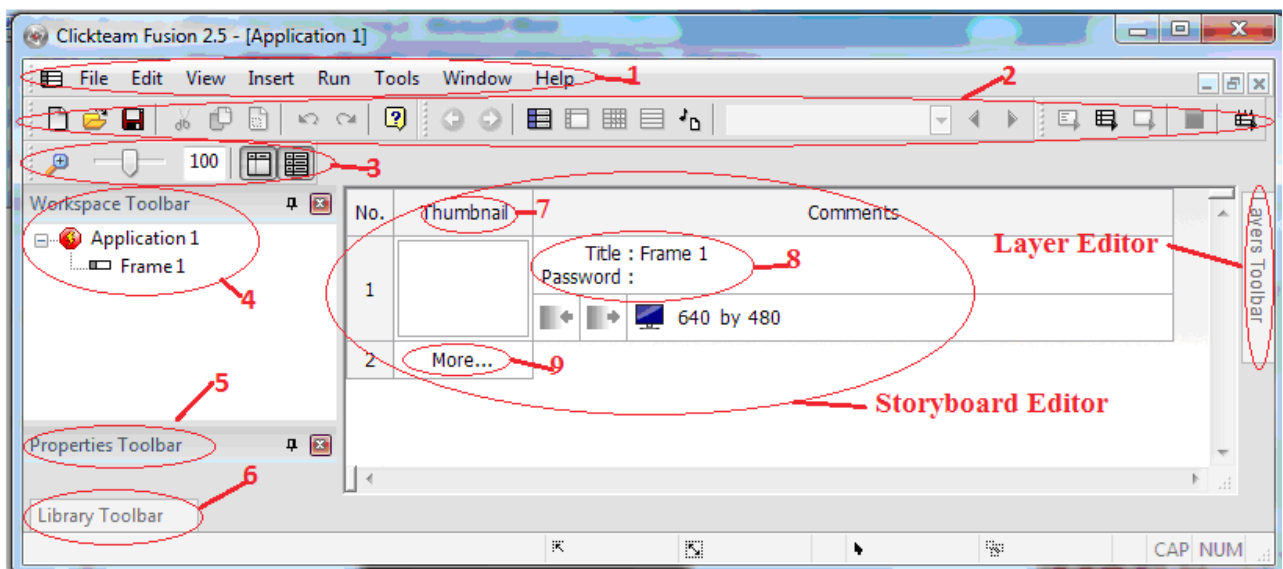
Меню Пуск ->  Clickteam Fusion 2.5 Free Edition

Для створення нового ігрового проекту виконати Start -> New application



або натиснути клавіші Ctrl + N.

Відкриється вікно нового додатка:



Вікно можна умовно розділити на кілька частин:

Головне меню додатка:

- 1) рядок меню, що знаходиться вгорі;
- 2) панель інструментів з іконками для швидкого доступу;

- 3) повзунок масштабу та іконки візуального представлення фрейма, що знаходяться в правій частині екрану;

Бокове меню:

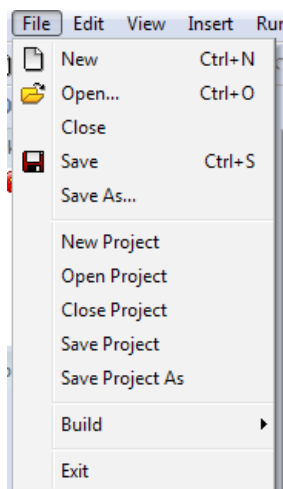
- 1) панель Workspace Toolbar, що надає швидкий доступ до налаштувань додатка, фреймів і елементів, що знаходяться в ньому.
- 2) панель Properties Toolbar (вона стане доступною після вибору елементів робочої панелі або елементів фрейму);
- 3) панель Library Toolbar – панель бібліотек;
- 4) робоче вікно Storyboard Editor, що дозволяє створювати нові рівні, копіювати їх і міняти порядок проходження; тут визначається розмір внутрішнього простору гри, додаються і коректуються ефекти, задаються паролі для кожного рівня доступу;
- 5) панель Thumbnail – зменшене зображення;
- 6) список фреймів (поки він один), дозволяє вибрати фрейм для редагування;
- 7) основні налаштування;
- 8) панель шарів Layer Editor – в правому краю екрану.

2 ГОЛОВНЕ МЕНЮ ПРОГРАМИ CLICKTEAM FUSION

Розглянемо головне меню додатка Clickteam Fusion:

У крайній правій частині знаходиться ікона властивостей самої програми (закрити, розгорнути та ін.). Її можна приховати, двічі клікнувши по ній. За нею йдуть елементи меню, здатні розкриватися, надаючи доступ до різних функцій.

МЕНЮ FILE – РОБОТА З ФАЙЛАМИ



New (Ctrl-N) – створити новий документ;

Open (Ctrl-O) – відкрити документ;

Close – закрити документ;

Save (Ctrl-S) – зберегти документ;

Save as ... – зберегти документ в зазначеному місці на диску із вказаним ім'ям;

New Project – створити новий проект;

Open Project – відкрити проект;

Close Project – закрити проект;

Save Project – зберегти проект;

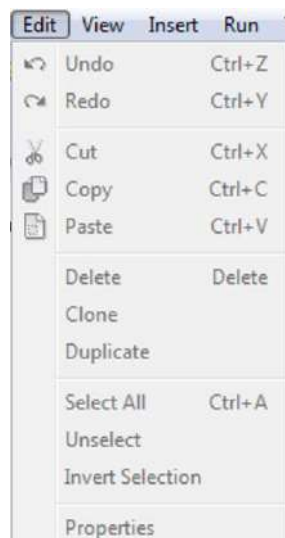
Save Project as ... – зберегти проект в зазначеному місці на диску із вказаним ім'ям;

Build – зібрати проект.

Exit – вихід з програми

Зібрати проект можна у вигляді готового додатка або програми-інсталлятора (а також і того й іншого). Залежно від наявних експортерів, ми можемо збирати проекти для різних операційних систем.

МЕНЮ EDIT – (РЕДАГУВАННЯ)



Undo (Ctrl-Z) – скасувати останню операцію;

Redo (Ctrl-Y) – повторити останню операцію;

Cut (Ctrl-X) – вирізати елемент з видаленням (з можливістю подальшої вставки);

Copy (Ctrl-C) – копіювати елемент без видалення (з можливістю подальшої вставки);

Paste (Ctrl-V) – вставити елемент;

Delete (Delete) – видалити елемент;

Clone – клонувати елемент: буде створено елемент з новим ім'ям (до найменування додається цифра), з яким можна проводити незалежні операції в редакторі подій;

Duplicate – дублювати елемент: на екрані створюється новий елемент з тим же ім'ям і параметрами, а в редакторі подій залишається один елемент;

Select All (Ctrl-A) – виділити все;

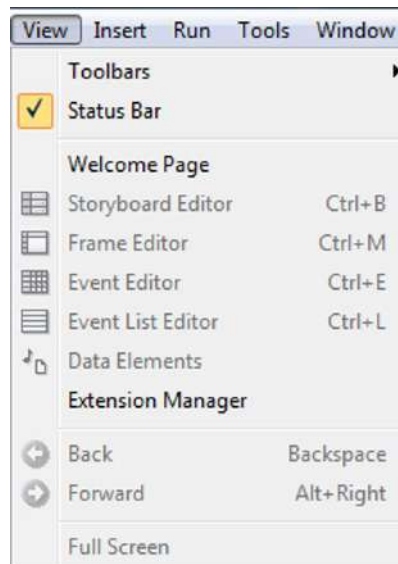
Unselect – скасувати виділення;

Invert Selection – інвертувати виділення;

Properties – відкрити вікно налаштувань.

На рисунку всі пункти неактивні, оскільки документ тільки-но відкрито, і не було зроблено ніяких дій.

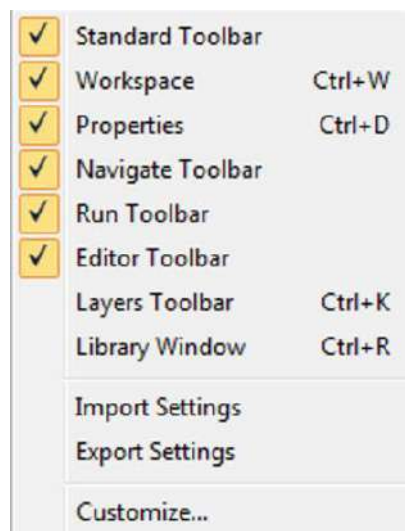
МЕНЮ VIEW – ПЕРЕГЛЯД



Інструменти меню Toolbars дозволяють нам відкривати або закривати на основному екрані панелі інструментів, що призначені для швидкого доступу шляхом зняття або встановлення прапорця. Розробник може налаштувати програму так, щоб на екрані було розташовано тільки ті панелі інструментів, з якими проходить активна робота.

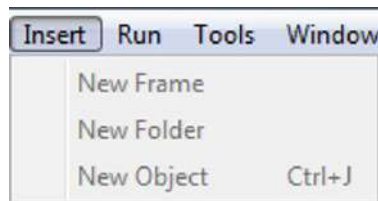
За допомогою команд меню Import Settings і Export Settings можна зберегти і за необхідності відновити зроблені налаштування.

МЕНЮ CUSTOMIZE



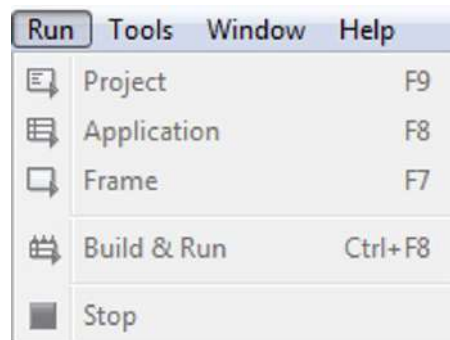
Дає нам можливість розширених налаштувань кожної панелі інструментів, аж до додавання і видалення будь-якого пункту.

МЕНЮ INSERT – ВСТАВКА



Меню дозволяє вставити новий фрейм (New Frame), нову папку (New Folder), новий об'єкт (New Object Ctrl-J) відповідно.

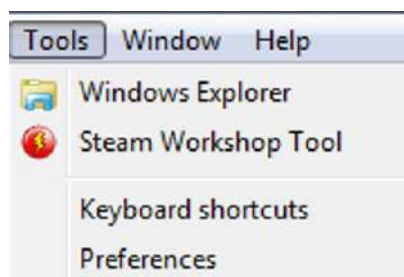
МЕНЮ RUN – ЗАПУСК



Меню Run дозволяє запускати весь проект, окремий додаток і окремий фрейм. При цьому треба зазначити, що при запуску одного фрейма кнопкою F7 буде запущено безпосередньо вибраний фрейм, а в разі наявності в ньому інструкцій щодо переходу на інший фрейм додаток буде просто закриватися.

Пункт Build & Run (Ctrl-F8) дозволяє зібрати і запустити додаток, а пункт Stop – зупинити його роботу.

МЕНЮ TOOLS – ІНСТРУМЕНТИ

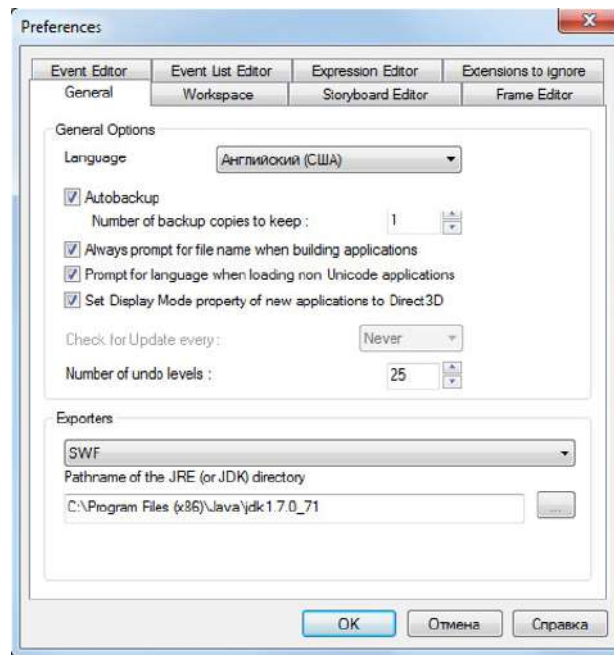


Windows Explorer – відкриває в провіднику Windows папку встановлення програми Clickteam Fusion;

Steam Workshop Tool – надає доступ до сховища різних додатків (як правило, в форматі *.mfa), які можна завантажувати і відкривати програмою Clickteam Fusion;

Keyboard shortcuts – ярлики клавіатури; відкриває вікно «Допомога по клавіатурі», де можна подивитися ті чи інші поєднання клавіш;

Preferences (переваги) – відкривається вікно налаштувань програми, що складається з декількох вкладок:



General – основні налаштування

Language – вибір мови зі списку, буде потрібно перезавантаження;

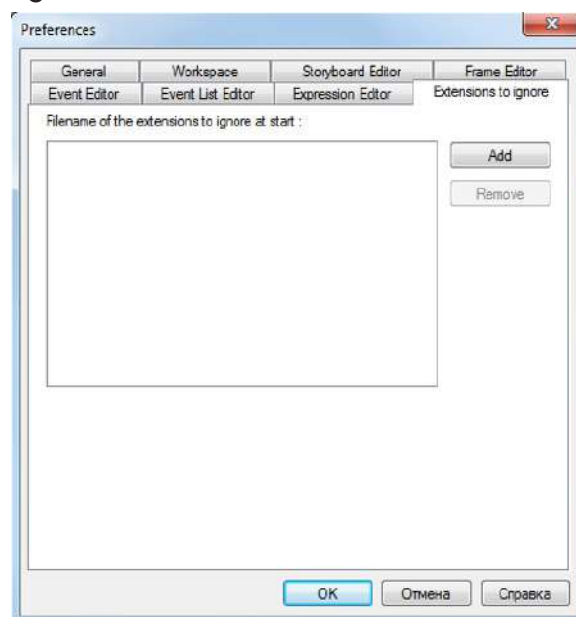
Exporters – налаштування експортерів.

Наприклад, для додатка, розробленого під Android, в першому полі необхідно встановити Android, в другому – вказати шлях до директорії JDK (скачати можна за посиланням www.oracle.com/technetwork/java/javase/downloads), а також Android SDK (developer.android.com/intl/ru/sdk/index.html).

При використанні BlackBerry Java потрібно встановити і вказати шлях до BlackBerry JDE.

При використанні HTML5 або SWF, необхідно вказати шлях до директорії JDK, так само як і для Android.

На вкладці Extensions to ignore



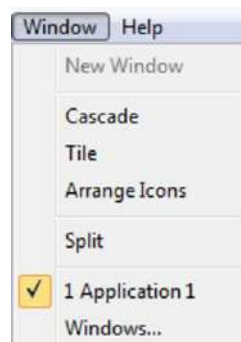
задаються розширення файлів, які слід ігнорувати під час запуску програми.

Наприклад, при відкритті завантаженого проекту з'являється таке вікно:



У даному випадку проект не може бути завантажено, поки не буде встановлено розширення Control X (шляхом натискання на кнопку Extension Manager і вибору розширення). Вийти з положення можна, додавши розширення ctrlx.mfx у вкладку Extensions to ignore. При цьому потрібно пам'ятати, що частину функціонала, що пов'язана з цим розширенням, буде втрачено.

МЕНЮ WINDOW – ВІКНО



New Window – відкрити нове вікно;

Cascade – розташувати вікна каскадом;

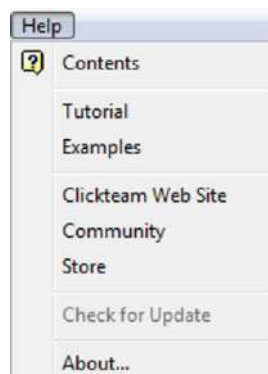
Tile – виділити вікно фреймів окремим вікном;

Arrange Icons – упорядкувати іконки;

Split – розділити фрейм на кілька частин.

Windows ... – вибір іншого активного документа зі списку.

МЕНЮ HELP



Contents – забезпечує доступ до довідкової системи;

Tutorial – навчальна система;

Examples – приклади;

Clickteam Web Site – буде відкрито вікно провідника Windows, виконано перехід на офіційний сайт;

Community – форум;

Store – магазин;

Check for Update – перевірка оновлень;

About – відомості про програму.

3 ПРИКЛАД СТВОРЕННЯ НАЙПРОСТІШОЇ ГРИ

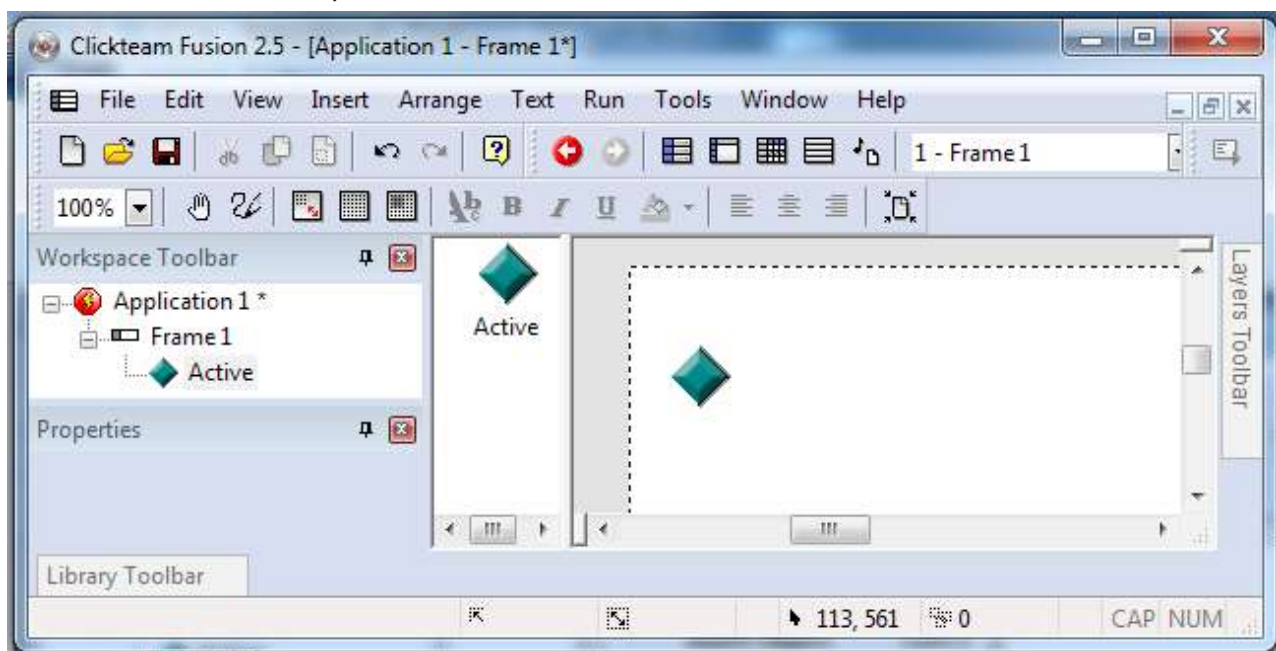
Створимо гру за сценарієм:

активний об'єкт «зелений ромб» рухається зверху до перешкоди, розташованої в центрі екрана. Керуючи об'єктом за допомогою стрілок або курсора, довести його до нижньої межі екрана.

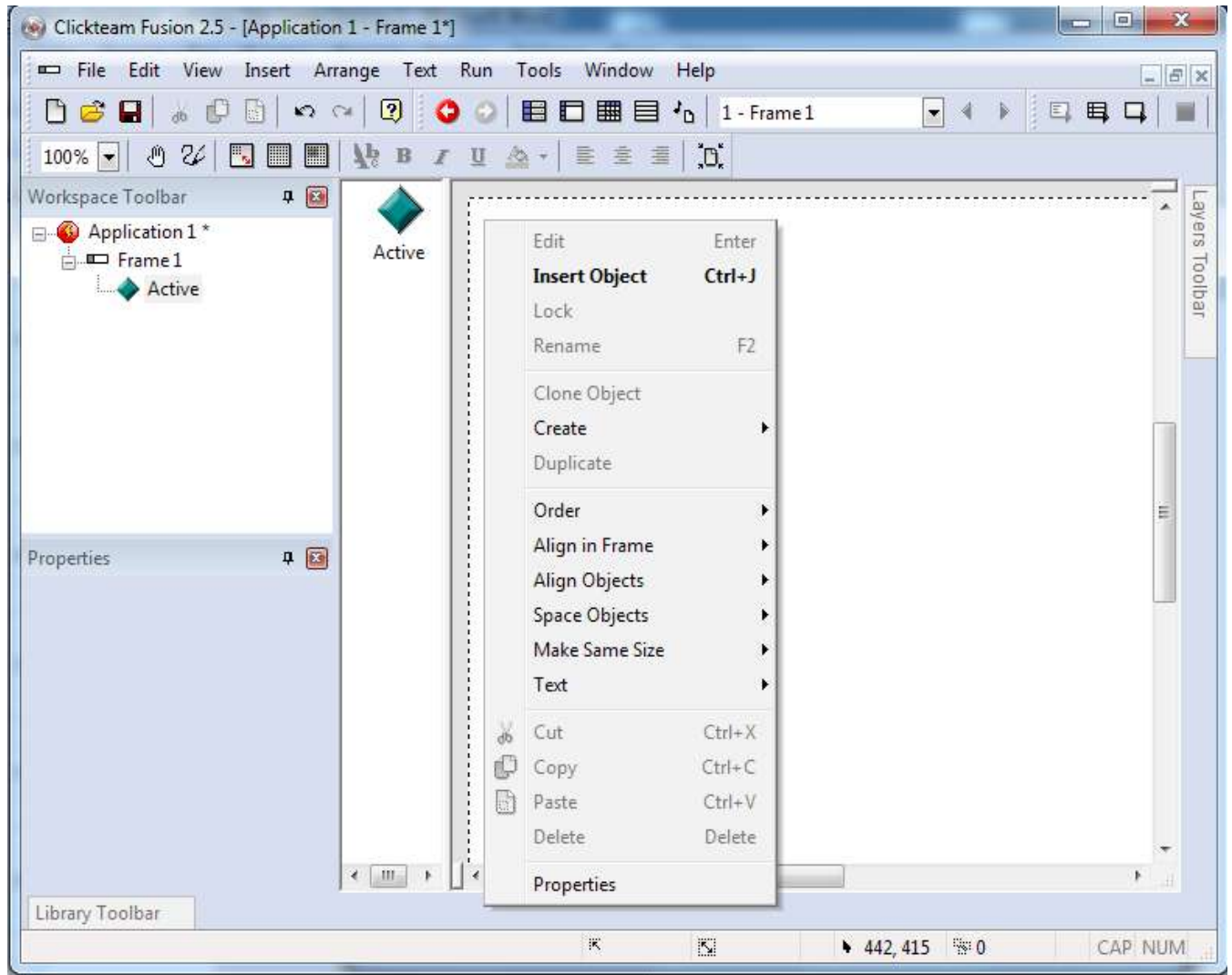
Реалізація проекту складається з кроків:

- 1) додати об'єкт «зелений ромб»;
- 2) налаштувати властивості об'єкта «зелений ромб»;
- 3) додати об'єкт «перешкода»;
- 4) налаштувати властивості об'єкта «перешкода»;
- 5) створити подію зіткнення об'єкта «зелений ромб» з об'єктом «перешкода»;
- 6) перевірити роботу проекту і зберегти його.

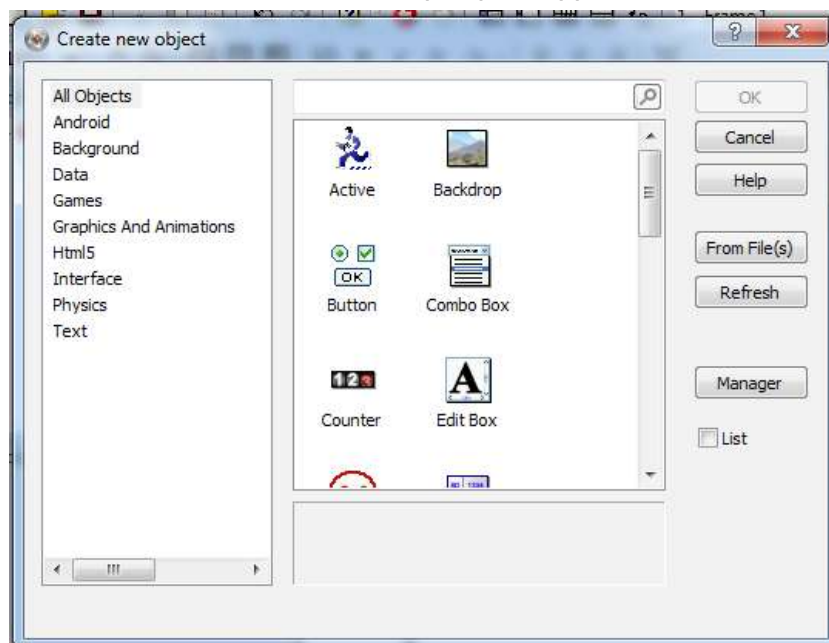
Як додати об'єкт на фрейм: двічі клікнути лівою кнопкою миші в білому прямокутнику в поле Thumbnail. Відкриється вікно Frame Editor:

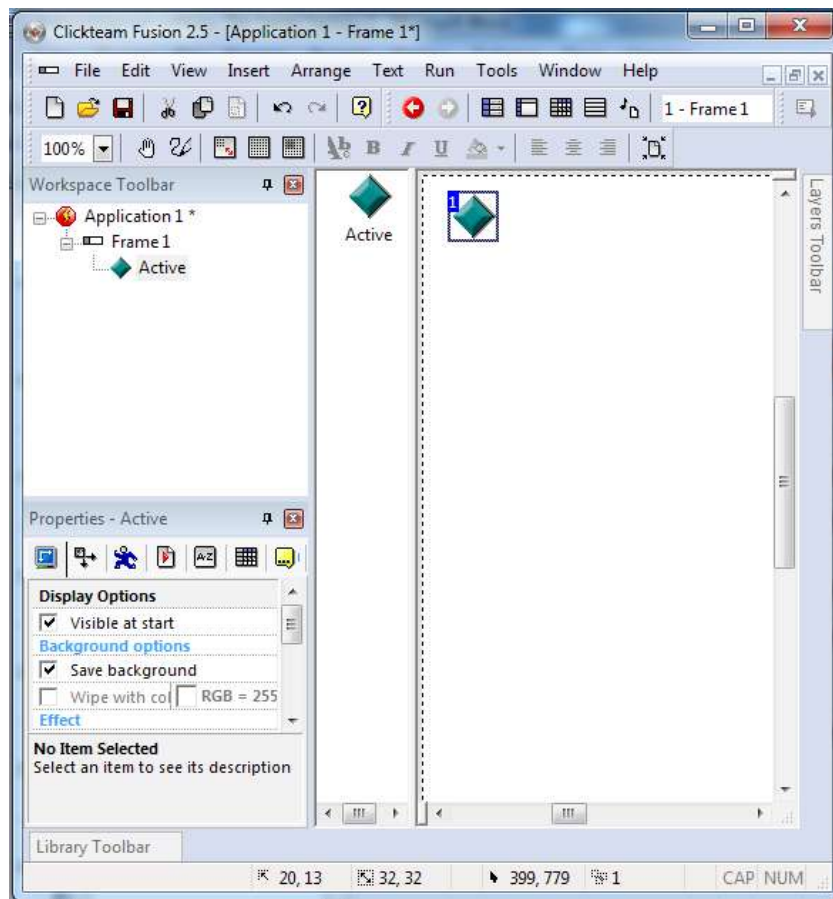


Як додати об'єкт на фрейм: RClick -> Insert Object:

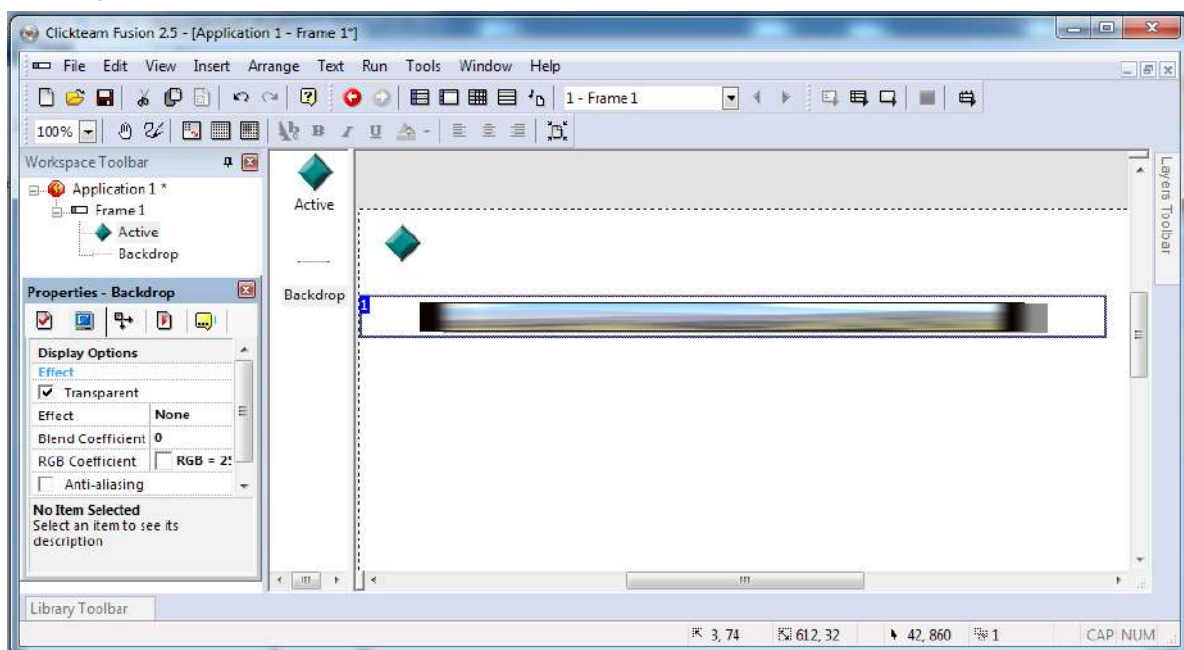


Вибрати тип об'єкта – Active, натиснути OK. Курсор набуде вигляду хрестика. В області редагування фрейма потягнути мишкою курсор на фреймі:



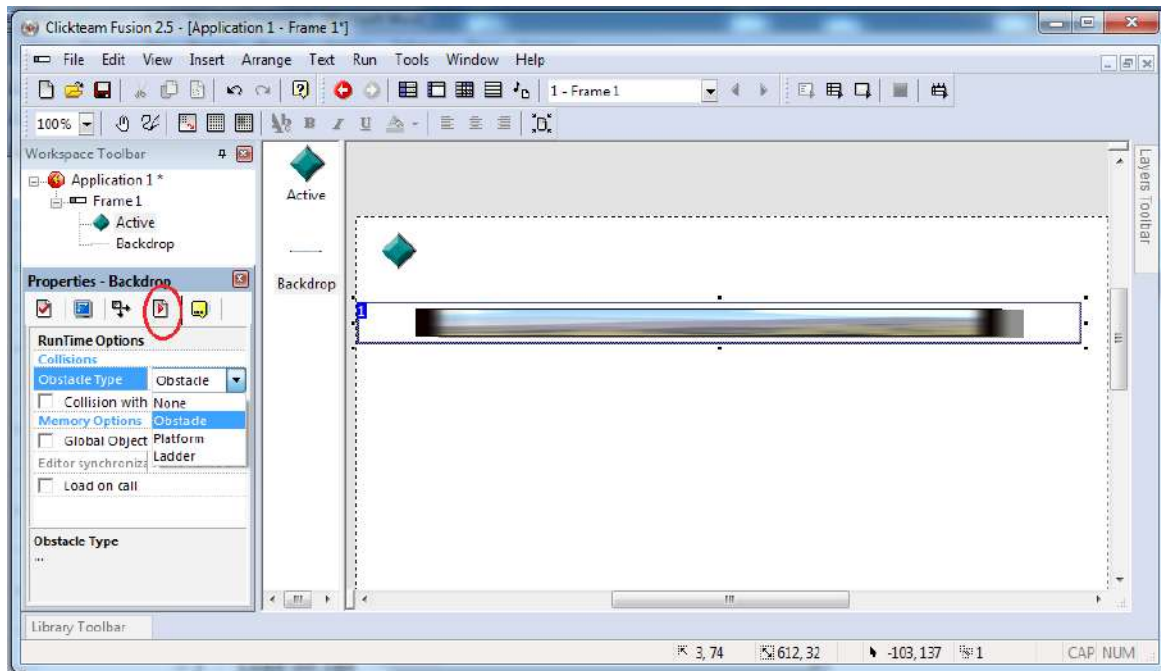


Розмістити на фреймі об'єкт Backdrop нижче об'єкта Active: Insert Object -> Background -> Backdrop:

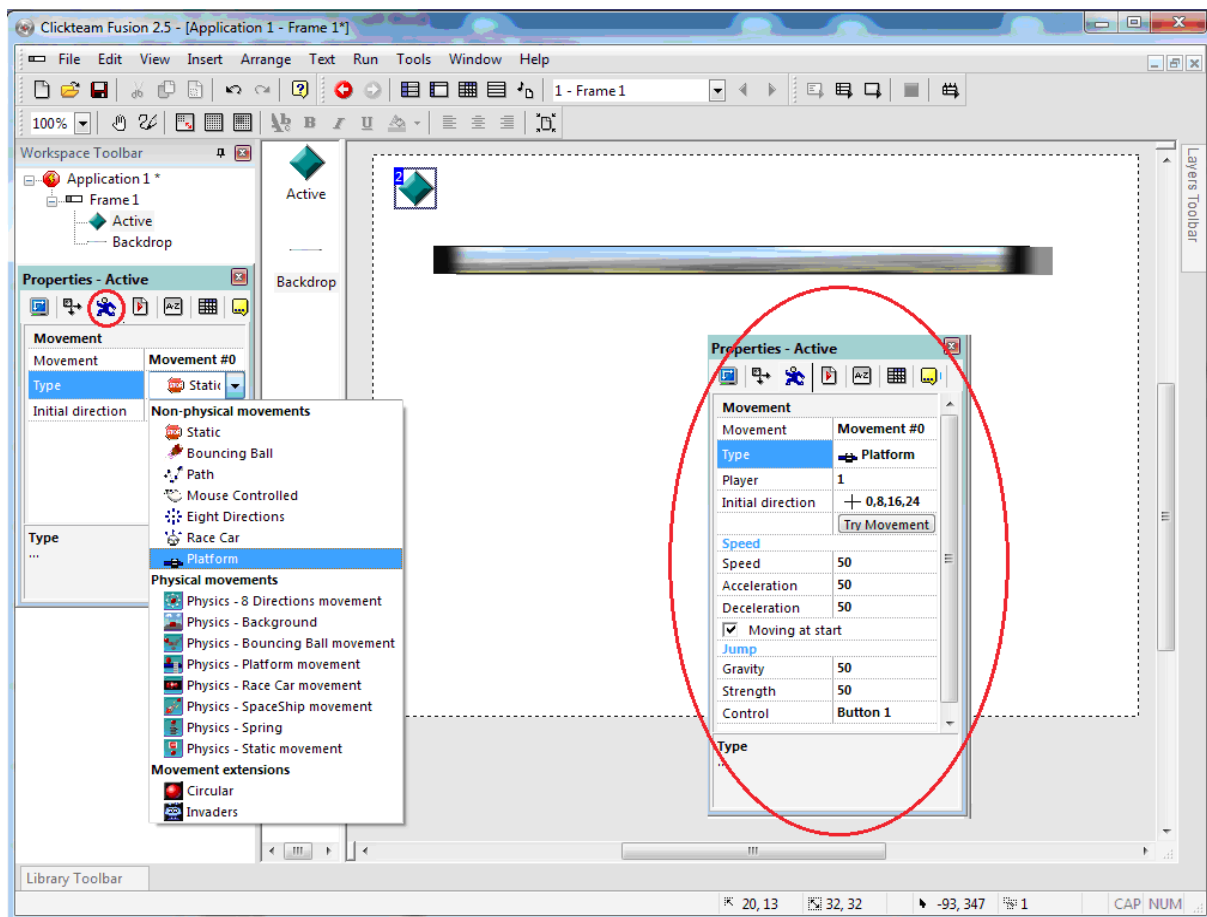


Клікнути лівою кнопкою миші по ньому один раз і потягнути вправо за праву точку, розтягуючи об'єкт на ширину фрейму.

У вікні налаштувань зліва показані властивості обраного об'єкта. Потрібно перейти в пункт RunTime Options (кнопка виділена червоним кружком) і в випадаючому меню Obstacle Type вибрати тип Obstacle.

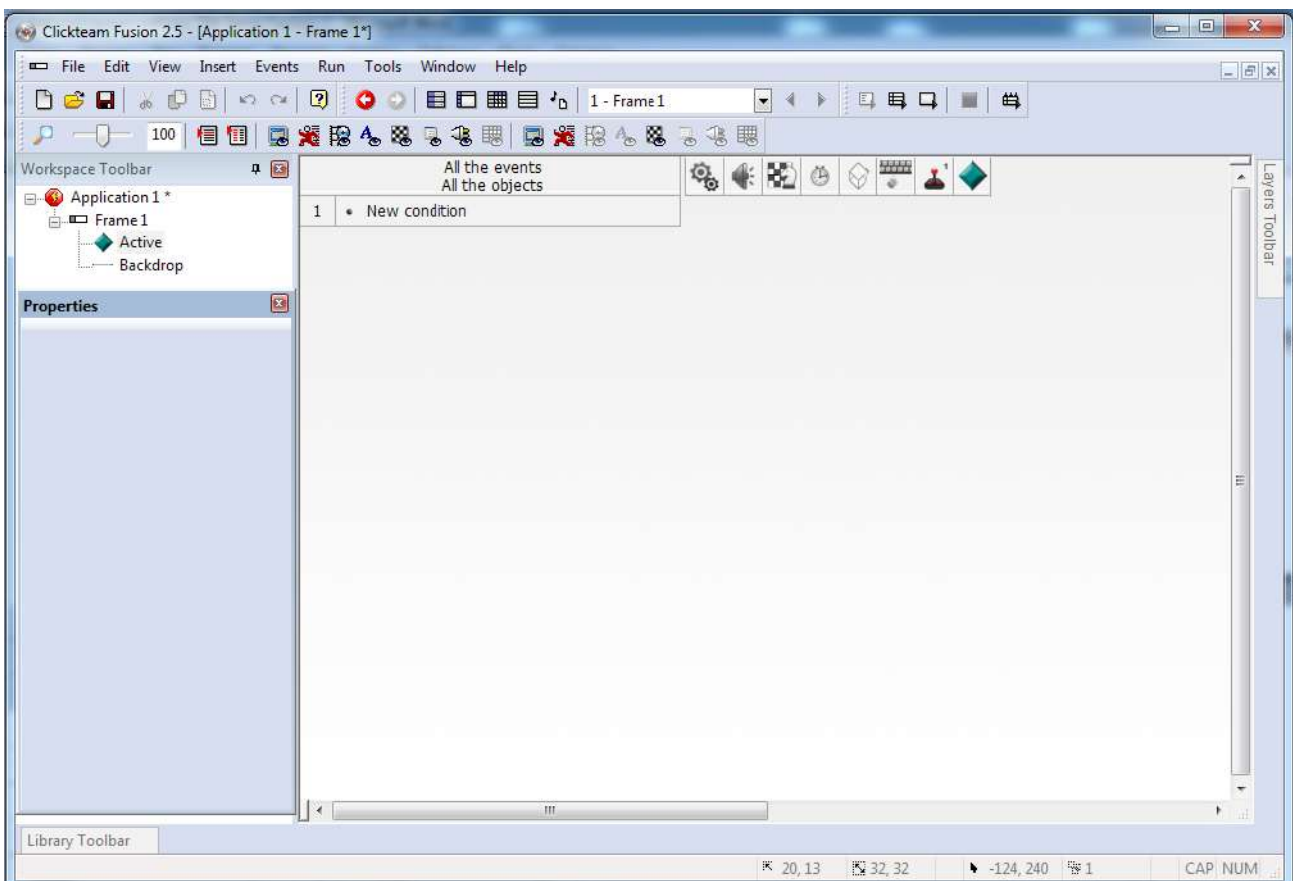
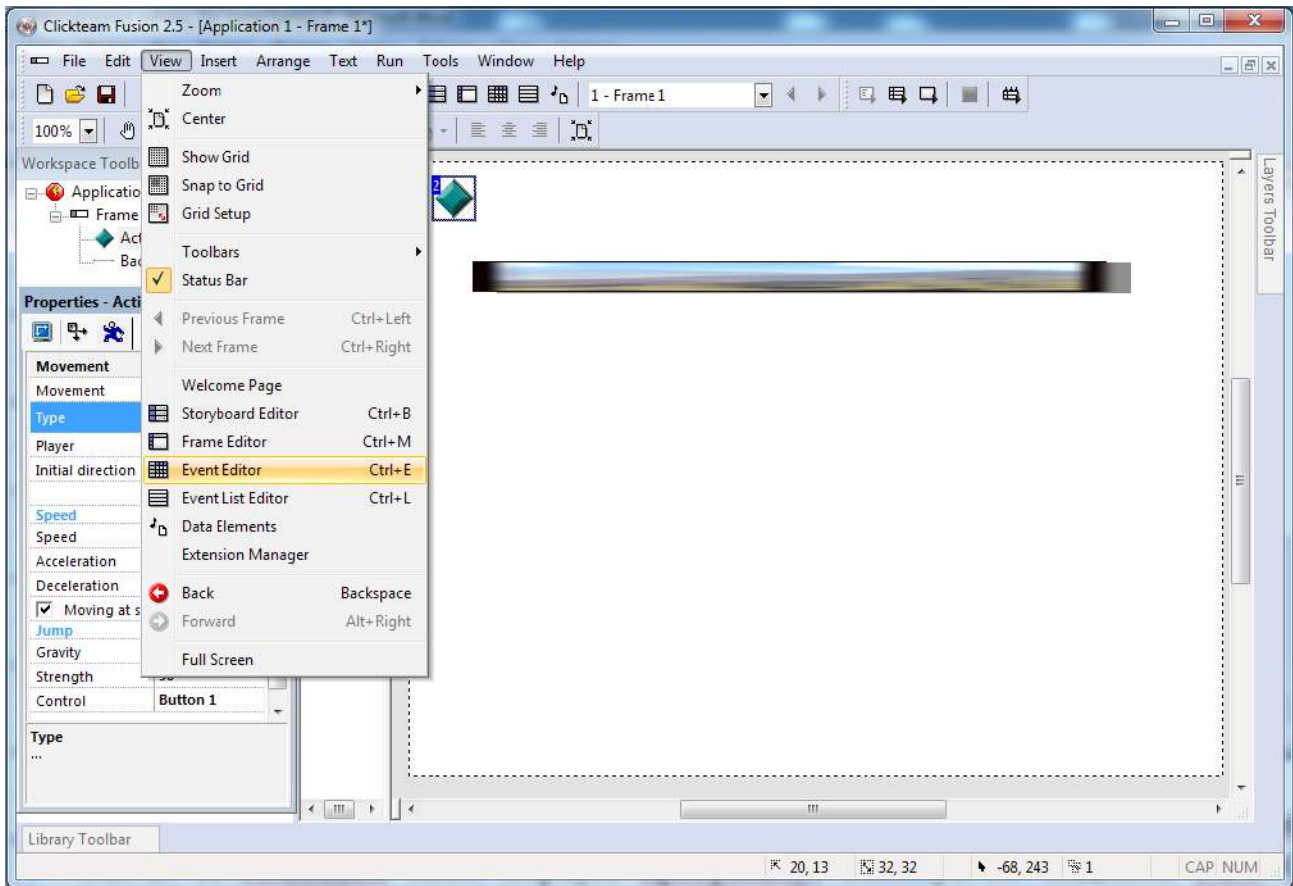


В області редагування фрейма клікнути мишкою на зеленому ромбоподібному об'єкті Active. У лівій нижній частині відобразяться його властивості. Вибрати вкладку Movement (кнопку виділено червоним колом) для визначення властивостей переміщення об'єкта. В полі Type вибрати зі списку тип переміщення Platform. В результаті властивості переміщення об'єкта Active приймуть такий вигляд, як показано в червоному овалі на малюнку:

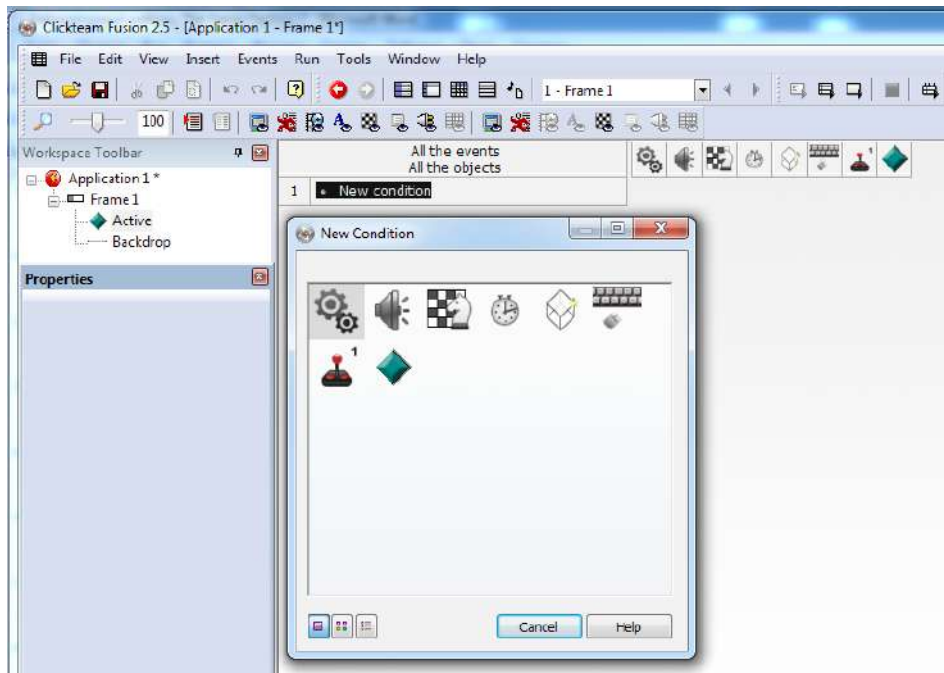


Лабораторні роботи

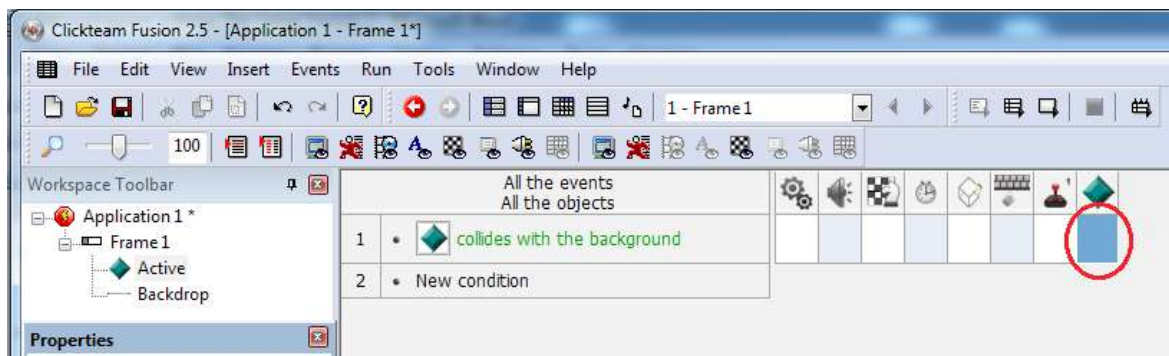
Перейти в редактор подій (Event Editor) за допомогою гарячої комбінації клавіш Ctrl-E або через меню:



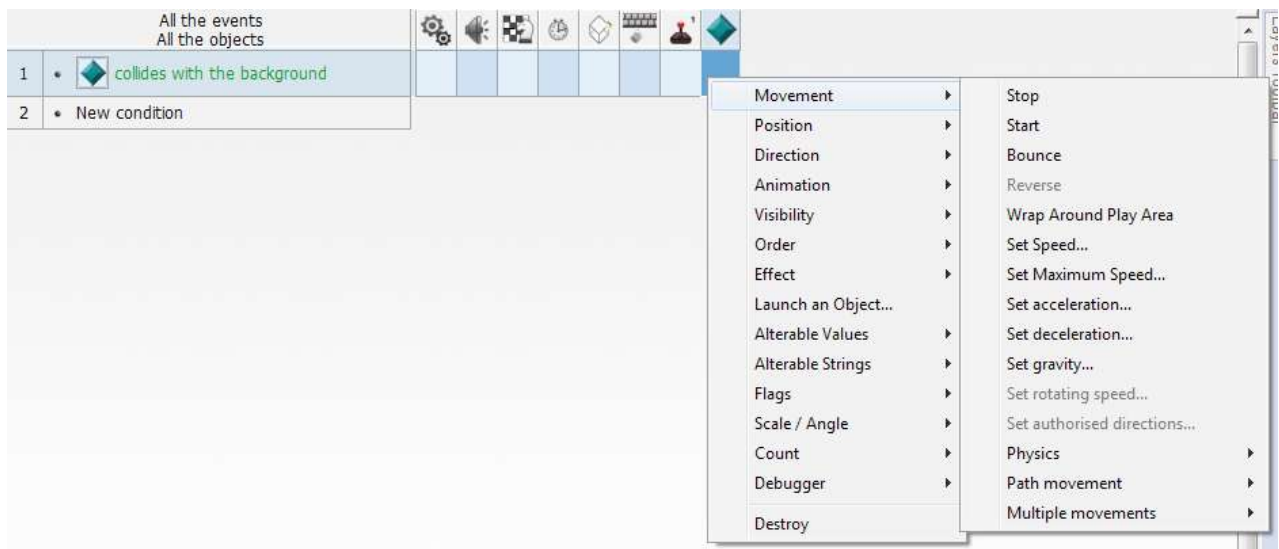
Для створення програми клікнути по напису New Condition правою клавішею миші:



Клікнути на ромбоподібному об'єкті Active, вибрати пункти Collision -> Backdrop і створити умову зіткнення об'єкта з «землею». Клікнути по квадрату під об'єктом Active (виділено червоним колом):



У правій колонці редактора клікнути на ньому правою кнопкою миші по клітці нижче ромба і вибрати Movement -> Stop:



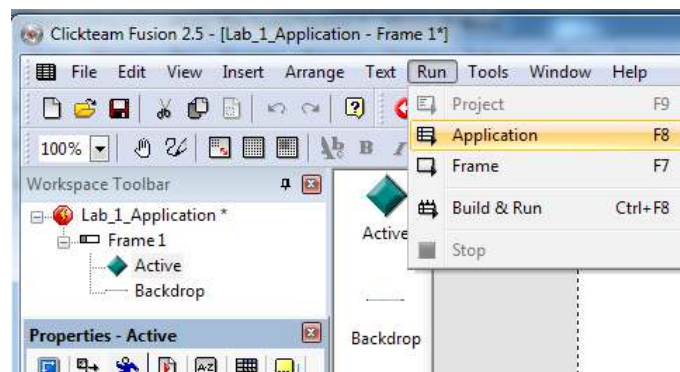
All the events All the objects									
1	•  collides with the background								
2	• New condition								

Перейти в Event List Editor (Ctrl-L):

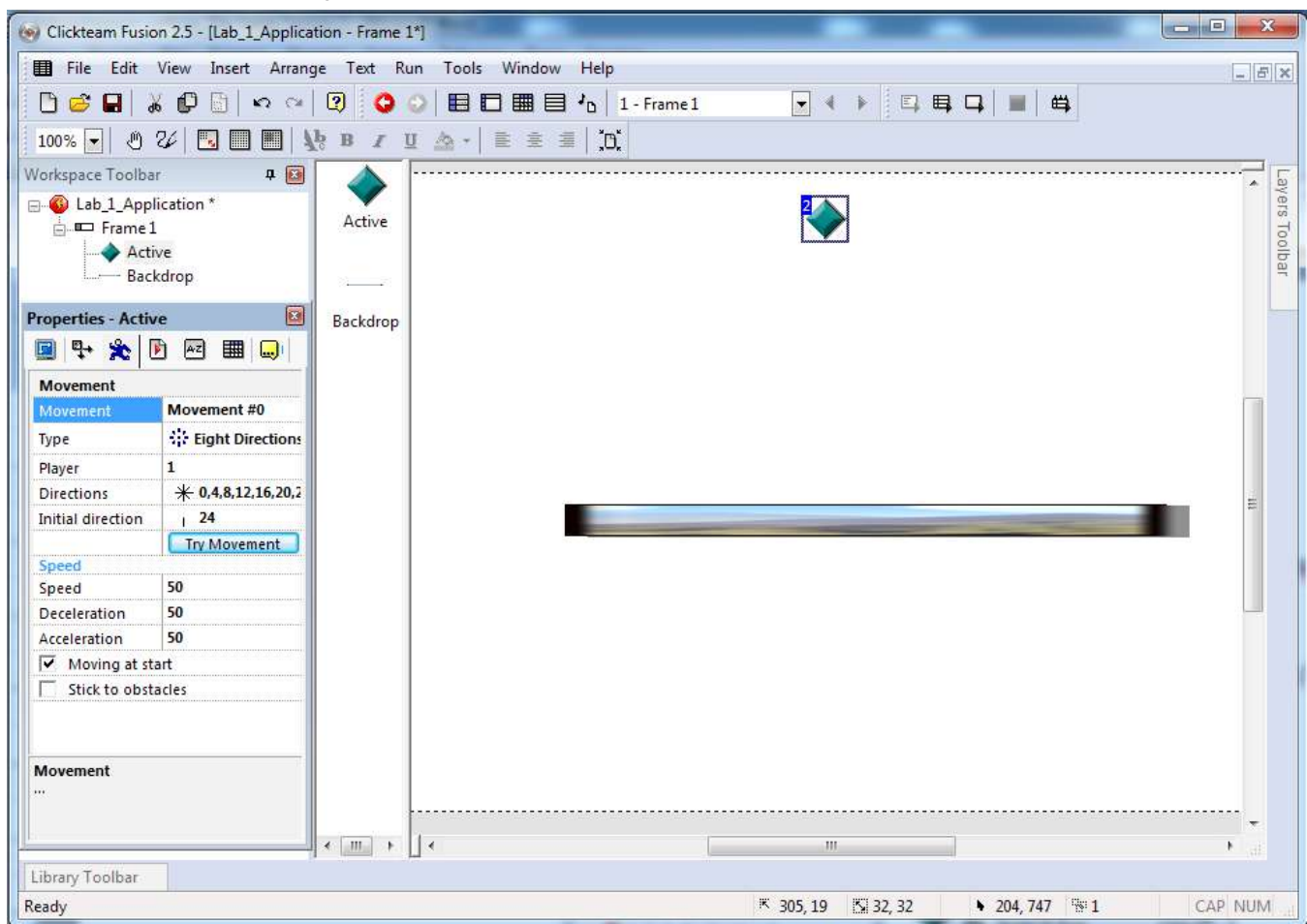
Якщо зображення на екрані комп'ютера збігається з малюнком вище, значить, програму створено правильно.

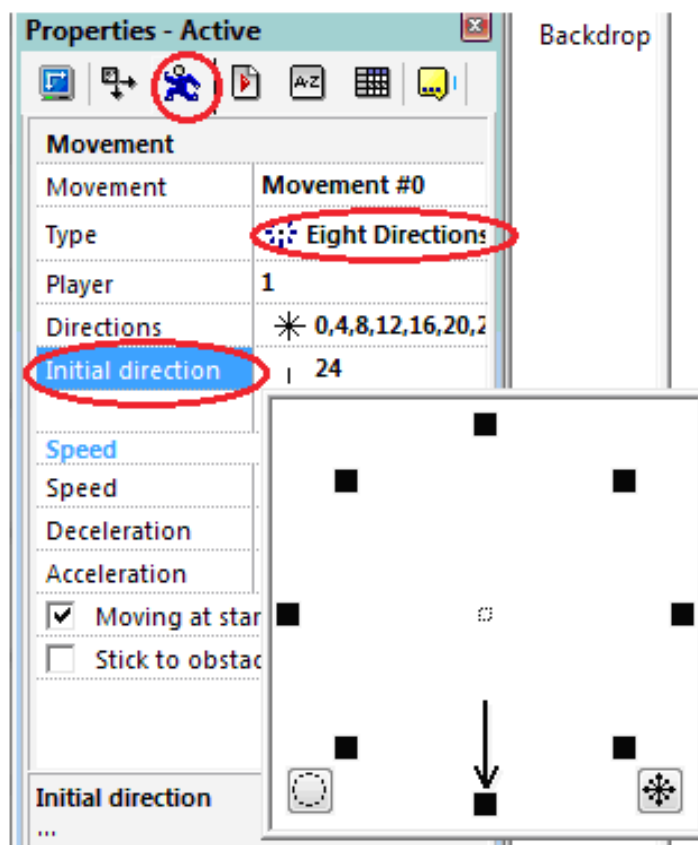
Зберегти додаток, виконавши команди File -> Save as

Перевірити роботу гри, натиснувши F8 або виконавши команди Run -> Application:



Змінити властивості руху об'єкта Active:





Перевірити роботу гри.

ЛАБОРАТОРНА РОБОТА №2

РОЗМІЩЕННЯ ФОНУ ТА ОБ'ЄКТІВ В ДОДАТКУ CLICKTEAM FUSION

МЕТА РОБОТИ:

- вивчення прийомів розміщення і блокування фону в додатку Clickteam Fusion;
- вивчення основних можливостей розміщення об'єктів в ігровій області Clickteam Fusion;
- визначення вигляду ігрового персонажа та інших ігрових об'єктів.

ПОРЯДОК РОБОТИ.

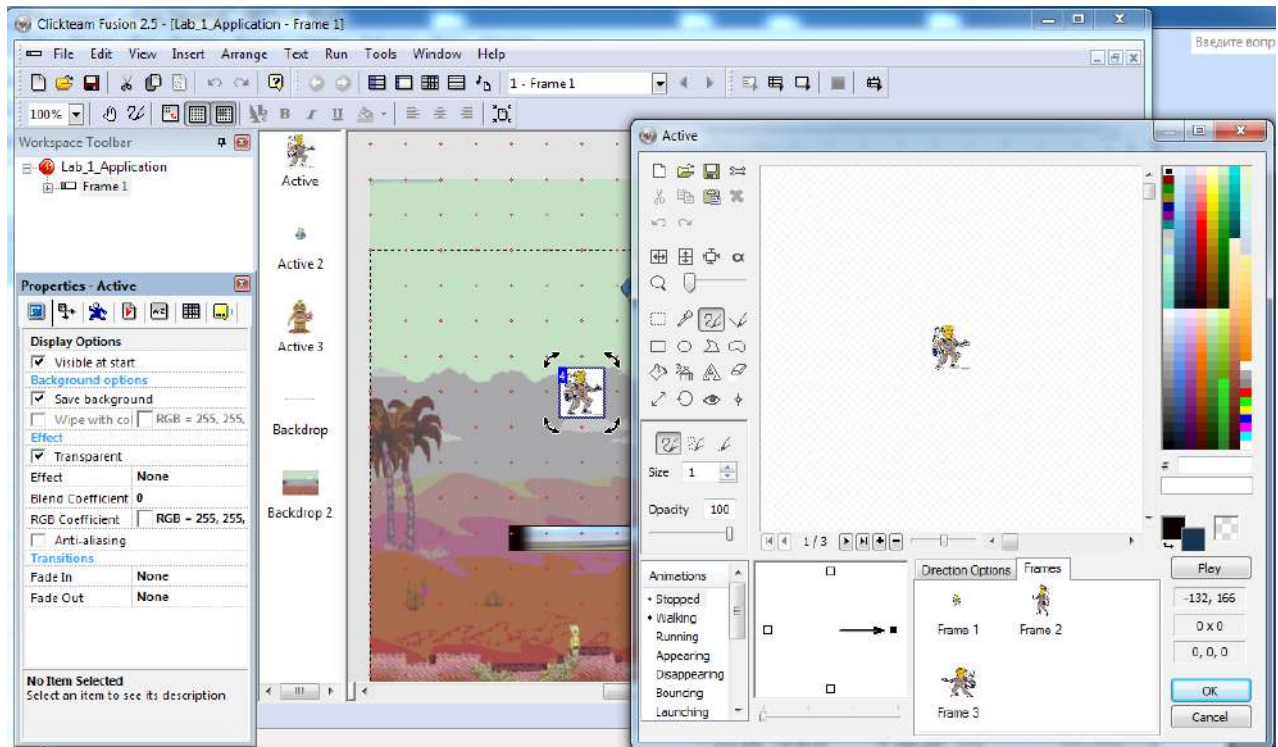
1. Вивчити інструкцію до лабораторної роботи.
2. Відкрити програми редактора Clickteam Fusion.
3. Відкрити проект, створений в лабораторній роботі №1.
4. Створити сценарій першої найпростішої гри.
5. Записати текст сценарію в звіті.
6. Створити об'єкти відповідно до сценарію. Задати властивості об'єктів.
7. Задати параметри руху і поведінки об'єктів.
8. Відредагувати зображення об'єктів.
9. Вибрати і помістити фонове зображення на фреймі. Заблокувати фонове зображення.
10. Перевірити працездатність проекту.
11. Дослідити можливі варіанти руху об'єктів, змінюючи відповідні їх властивості.
12. Зберегти створений проект.
13. Продемонструвати працездатність проекту викладачеві.
14. Закрити додаток Clickteam Fusion.
15. Оформити звіт. Захистити роботу.

РОЗМІЩЕННЯ ФОНУ І ВІЗУАЛІЗАЦІЯ ОБ'ЄКТІВ В ДОДАТКУ CLICKTEAM FUSION

1 РЕДАКТОР ЗОБРАЖЕНЬ

Редактор зображень є важливою частиною Clickteam Fusion 2.5: він дозволяє створювати і редагувати всі зображення і значки, що містяться в додатку. Це повний набір функцій редагування, при підтримці розширених функцій. Він містить всі необхідні інструменти для малювання, імпорту та редагування зображень.

Подвійний клік мишею по об'єкту відкриває вікно редактора зображень:



ОБЛАСТЬ РЕДАГУВАННЯ

Область редагування знаходиться в центрі діалогового вікна і містить зображення для редагування. Для малювання потрібно: вибрати інструмент з панелі інструментів малювання зліва, вибрати колір (а) малюнка в палітрі кольорів праворуч, а потім використовувати мишу, щоб намалювати зображення.

Якщо зображення більше області редагування, можна використовувати смуги прокрутки для перегляду прихованих областей. Також можна натиснути клавішу SPACE і, не відпускаючи її, перемістити зображення лівою кнопкою миші.

Для збільшення зображення використовують слайдер «Масштаб» в інструментах «Малювання» або коліщатко миші.

Інструменти малювання



Clear: Стирає поточне зображення.



Import: Відкриває діалогове вікно «Імпорт», що дозволяє імпортувати файл зображення в редактор. Дивись нижче.



Export: Зберігає зображення як зовнішній файл.



Options: Дозволяє вибрати функцію клікання правою кнопкою миші на зображенні, другий (фоновий) колір, або вибрати колір із зображення.



Cut: Копіює поточний виділений фрагмент в буфер обміну і видаляє його, з екрану



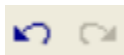
Copy: Копіює поточний виділений фрагмент в буфер обміну.



Paste: Вставляє вміст буфера обміну в зображення.



Delete: Видаляє виділений фрагмент.



Undo/Redo: Скасовує/повторює останні дії.



Flips horizontally: повертає зображення горизонтально. Натиснути SHIFT, щоб одночасно повернути альфа-канал.



Flips vertically: повертає зображення вертикально. Натиснути SHIFT, щоб одночасно повернути альфа-канал.



Crop: Змінює розмір зображення, усуваючи порожню область навколо нього. Можна використовувати комбінацію клавіш для обрізки декількох зображень одночасно, коли ви натискаєте кнопку Crop: SHIFT щоб обрізати всі зображення в заданому напрямку або зі списку зображень, SHIFT + CONTROL, щоб обрізати всі напрямки поточної анімації, SHIFT + CONTROL + ALT, щоб обрізати всі анімації поточного об'єкта.



Transparency mode: У цьому режимі можна вибрати перегляд і редагування прозорого кольору або альфа-каналу. При малюванні в альфа-каналі кольорова палітра замінюється відтінками сірого.



Magnifier: Встановлює коефіцієнт масштабування, який використовується для відображення зображення. Слайдер поруч із позначкою дозволяє змінити коефіцієнт масштабування.



Selection: Дозволяє зробити блок навколо областей зображення. Слайдер Opacity дозволяє зробити обрану область напівпрозорою.



Color picker: Дозволяє вибрати колір в зображенні за допомогою миші. Ліва кнопка миші дозволяє вибрати колір переднього плану, права кнопка миші дозволяє вибрати колір фону.



Brush: Використовується для малювання пікселів і ліній на зображенні. Параметри у вікні налаштувань дозволяють змінювати розмір пензля і змінювати його прозорість.



Lines: дозволяє малювати прямі лінії на зображенні. Параметри у вікні налаштувань дозволяють змінювати розмір лінії і змінювати її прозорість.



Rectangle: Використовується для малювання прямокутних фігур на зображенні. Параметри у вікні налаштувань дозволяють вибирати між зафарбованим, контуром і зафарбованим + контуром – прямокутником, змінювати розмір границі і змінювати прозорість прямокутника.



Ellipse: Використовується для малювання кіл та еліпсів. Параметри у вікні налаштувань дозволяють вибирати між зафарбованим, контуром і зафарбованим контуром еліпса, змінювати розмір границі і змінювати прозорість еліпса.



Polygon: Дозволяє малювати замкнуті багатокутники. Лінії, що визначають багатокутник, малюють лівою кнопкою миші, і подвійне клікання миші замикає лінії багатокутника. Параметри у вікні налаштувань дозволяють вибирати між зафарбованим, контуром і зафарбованим + контуром багатокутника, змінювати розмір границі і змінювати прозорість багатокутника.



Shape: Використовується для малювання замкнутого контура довільної форми за допомогою миші. Параметри у вікні налаштувань дозволяють вибирати між зафарбованим, контуром і зафарбованим + контуром, змінювати розмір контуру і змінювати прозорість фігури.



Fill: Заповнює замкнутий контур одним кольором. Можна змінити кольорову гаму у вікні налаштувань.



Spray: Спрей використовується для малювання розпиленням. Можна змінити розмір і тиск розпилювача у вікні налаштувань для малювання.



Text: Використовується для малювання тексту. Виберіть шрифт та атрибути шрифту і введіть текст в діалоговому вікні. Ви можете переміщати текст за допомогою миші, коли діалогове вікно відкрите. Кнопка «Імпорт шрифту» активується, коли ви редагуєте об'єкт «Лічильник», «Оцінка на життя», що відображається як «Числа». Він дозволяє імпортувати номери зі шрифту.



Eraser: Малює прозорим кольором, тому стирає зображення.



Resize: Використовується для зміни розміру зображення. Введіть новий розмір зображення в зонах редагування і натисніть кнопку Apply (Застосувати). Якщо ви виберете Proportional (Пропорційний), іншу координату буде розраховано автоматично, щоб зберегти пропорції зображення. Stretch (Розтягування) розтягує зображення до нового розміру, Resample (Скопіювати за прикладом) буде повторно відображати нове зображення і покращувати ефект зміни розміру.



Rotation: Використовується для повороту зображення. Введіть кут і натисніть кнопку Apply (Застосувати), і цей інструмент поверне зображення навколо його гарячої точки.



Hot spot: гаряча точка – це точка на зображенні, відповідна фактичним координатам X і Y об'єкта. Ви можете встановити положення гарячої точки на зображенні за допомогою цього інструмента. Кнопки Quick Move (швидке переміщення) дозволяють переміщати гарячу точку в певні позиції на зображенні: верхню ліву, верхню-центральну, верхню праву, центр-ліворуч, центр і т. д. Якщо ви редагуєте список зображень, наприклад, напрямок в редакторі анімації, ви переміщаєте гарячу точку в усіх зображеннях, натискаючи ALT, і змінюючи положення гарячої точки.



Action point: Точка дії є місцем розташування в зображенні, де відбувається дія (наприклад, при запуску об'єкта куля буде створена в позиції точки дії). Ви можете встановити положення точки дії за допомогою цього інструменту. Кнопки Quick Move (швидкого переміщення) дозволяють переміщати точку дії в певні положення на зображенні: верхнє ліве, верхнє-центральне, верхнє праве, центр-ліве, центр і т. д. Якщо ви редагуєте список зображень, наприклад, напрямок в редакторі анімації, ви переміщаєте точку дії у всіх зображеннях, натискаючи ALT, і змінюючи положення точки дії.

Кольорова палітра

Кольорова палітра залежить від об'єкта, який ви редагуєте. Якщо ви редагуєте об'єкт з поточного фрейма, він буде містити 256-кольорову палітру кадру. Якщо ви редагуєте значок об'єкта, він буде містити стандартну палітру півтонів Windows. Якщо ви редагуєте альфа-канал, він буде містити відтінки сірого кольору. Натисніть один з

кольорів в палітрі кольорів, щоб вибрати колір для використання. Використовуйте ліву кнопку, щоб вибрати колір переднього плану, і праву кнопку, щоб вибрати колір фону.

Прямокутники під палітрою кольорів містять колір фону, колір переднього плану і прозорий колір. Щоб малювати прозорим кольором, клікніть поле Transparent color box (Прозорий колір) за допомогою лівої чи правої кнопки.

Прозорий колір

Кожне зображення в Clickteam Fusion 2.5 має прозорий колір, навіть якщо він відображається як непрозорий. Щоб змінити прозорий колір, ви можете натиснути клавішу SHIFT, клікнувши за кольором в колірній палітрі або вибрати інструмент Transparency (Прозорість) на панелі інструментів і вибрати колір на зображенні.

У версії Clickteam Fusion 2.5 будь-який колір може бути прозорим (в попередніх версіях Clickteam Fusion 2.5 тільки чорний колір був прозорим).

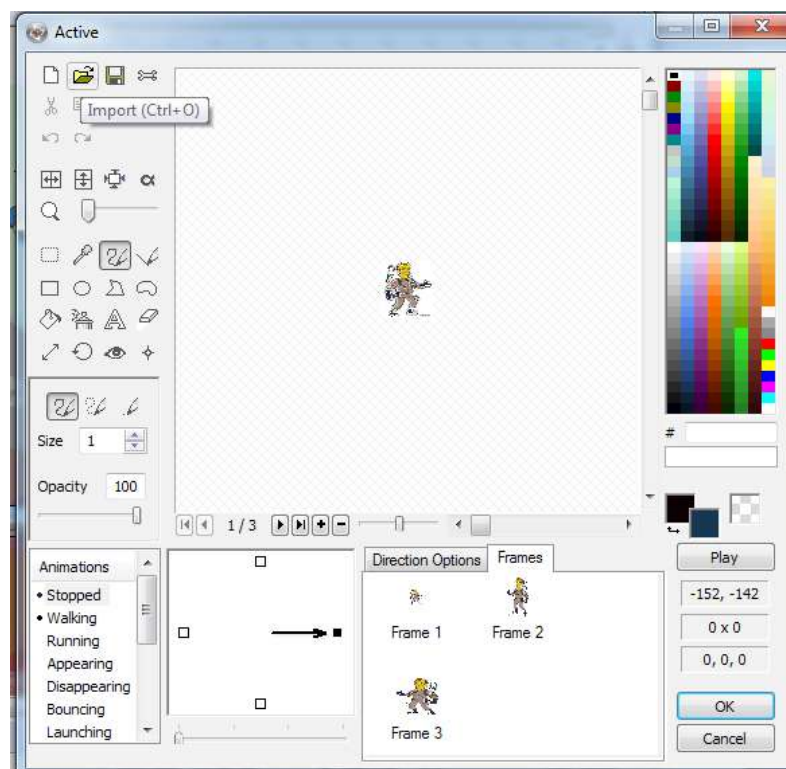
Індикатори

3 індикатора в нижній частині діалогового вікна містять:

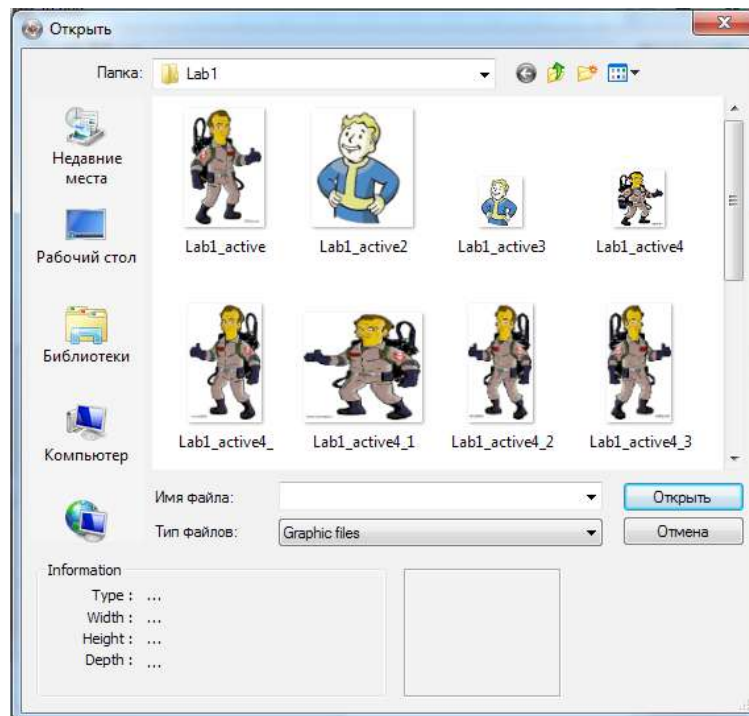
- координати курсора миші;
- розмір виділення;
- значення RGB пікселя під курсором миші.

Діалог Імпорт

Коли ви натискаєте кнопку «Імпортувати», відкривається селектор файлів, що дозволяє вибрати файл для імпорту зображення:

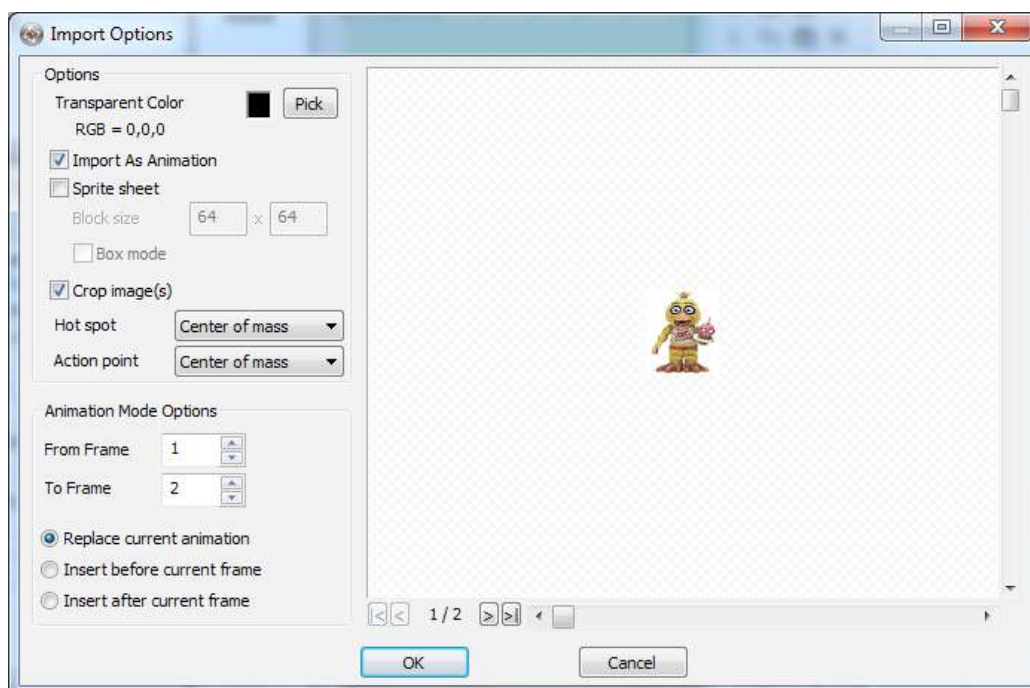


Залежно від типу зображення або анімації, яку ви імпортуєте, ви зможете вибрати або прості файли зображень, або файли анімації:



Примітка: нумеровані файли зображень вважаються анімацією. Наприклад, якщо ви виберете файл IMAGE001.BMP, а файли IMAGE002.BMP і IMAGE003.BMP існують в тому ж каталозі, то Clickteam Fusion 2.5 завантажить 3 зображення і відобразить їх в діалоговому вікні «Імпорт зображень». У верхній частині діалогового вікна відображається зображення для імпорту. Використовуйте смуги прокрутки для відображення прихованих областей.

Коли файл зображення для імпорту вибраний, відкриється наступне діалогове вікно:



Прозорий колір (Transparent color)

Зазначений колір буде сприйматися як прозорий (області під об'єктом буде видно). Використовуйте кнопку «Вибір», щоб вибрати інший колір безпосередньо з зображення, або двічі клікніть кольоровий квадрат, щоб вибрати колір в селекторі кольорів. Якщо ви не хочете використовувати прозорий колір, просто виберіть колір, якого немає в зображенні.

Режим коробки (Box mode)

Ця опція дуже потужна: вона дозволяє захоплювати кілька зображень, що містяться в блоках в одному файлі зображення. Щоб ця опція працювала, ви повинні дотримуватися деяких правил при збереженні файла зображення. По-перше, кожен графічний об'єкт, який ви хочете імпортувати, повинен бути оточений прямокутником кольору, відмінного від кольору фону зображення. Один і той самий колір має використовуватися для визначення всіх прямокутників, які оточують графічні об'єкти, що містяться в зображенні. По-друге (необов'язково) ви можете визначити гарячі точки (hot-spot) і точки дії (action points) кожного зображення з пікселем іншого кольору в горизонтальній і вертикальній лініях прямокутників. Щоб визначити гарячу точку, вкажіть її положення за допомогою пікселя іншого кольору у верхній лінії (координата X) і в лівій лінії (координата Y). Щоб визначити точку дії, помістіть інший піксель в нижню лінію (координата X) і праву лінію (координата Y).

Процес захоплення працює зліва направо і зверху вниз (наприклад, читання). Якщо ви захопите анімацію, верхній лівий кадр буде захоплений першим.

Імпорт як анімація (Import As Animation)

Якщо цей параметр не вибрано, перше імпортоване зображення замінить зображення, що редагується в редакторі зображень. Якщо цей параметр вибрано, активується інша група параметрів:

- From Frame (З кадру) – вказує індекс першого зображення, що цього буде імпортовано.
- To Frame (До кадру) – вказує індекс останнього зображення, що цього буде імпортовано.
- Replace current animation (Замінити поточну анімацію) – список редагованих кадрів буде очищено перед імпортом нових зображень.
- Insert before current frame (Вставити перед поточним фреймом) – імпортовані зображення будуть вставлені перед редагованим кадром.
- Insert after current frame (Вставка після поточного кадру) – імпортовані зображення будуть вставлені після редагованого кадру.

Вибір імпорту

Імпортує виділену частину зображення. Щоб вибрати частину зображення, натисніть кнопку Select (Вибрати) і намалюйте прямокутник навколо області для імпорту.

Імпорт в якості вибору

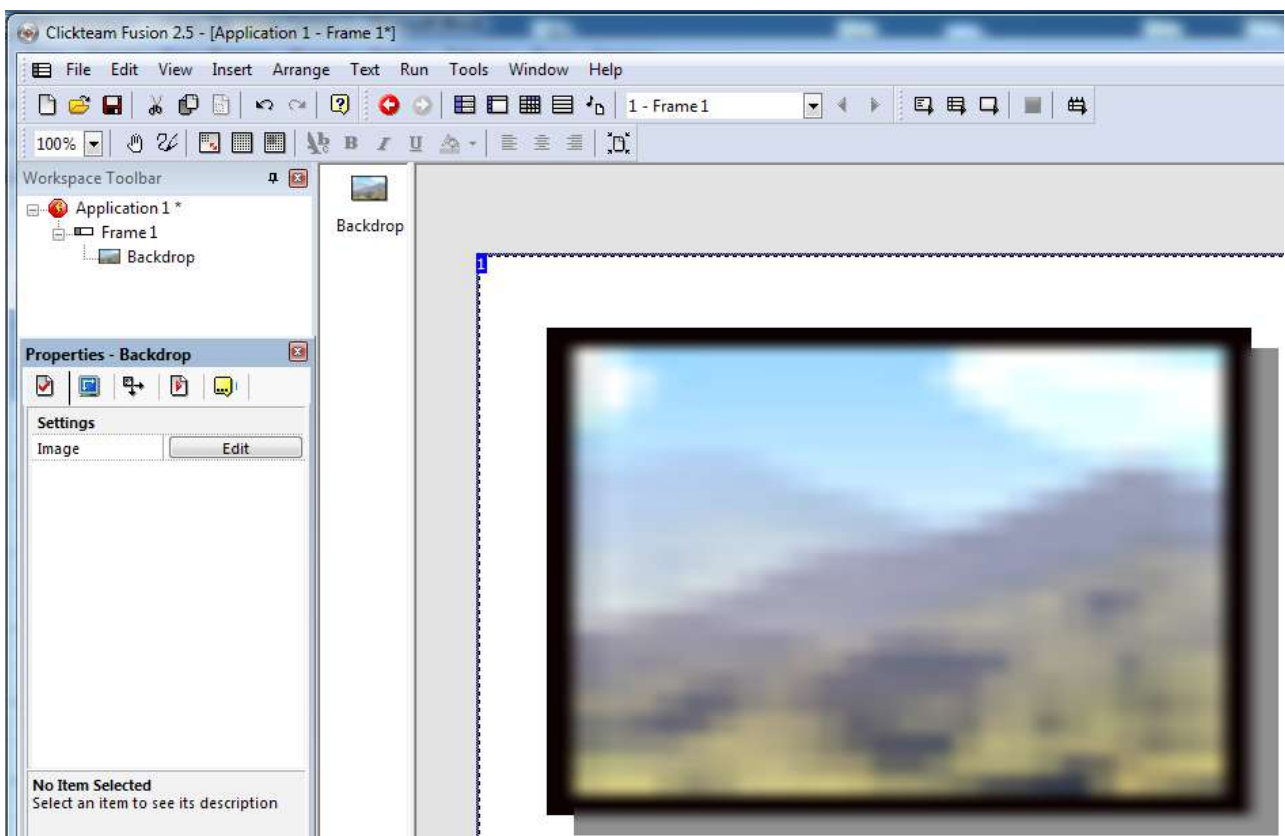
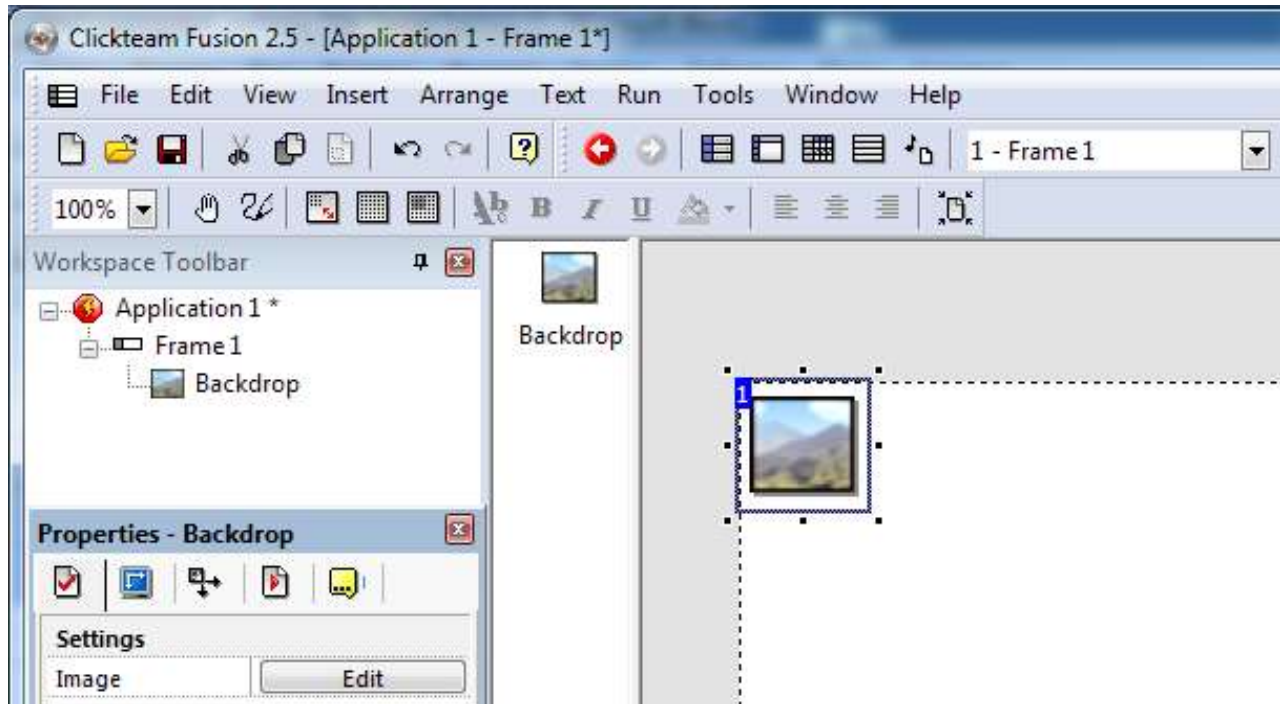
Якщо цей прапорець встановлений, імпортоване зображення не буде замінювати поточне зображення, але буде імпортовано як вибір в редакторі зображень.

Приклад

Storyboard Editor -> DClick Thumbnail

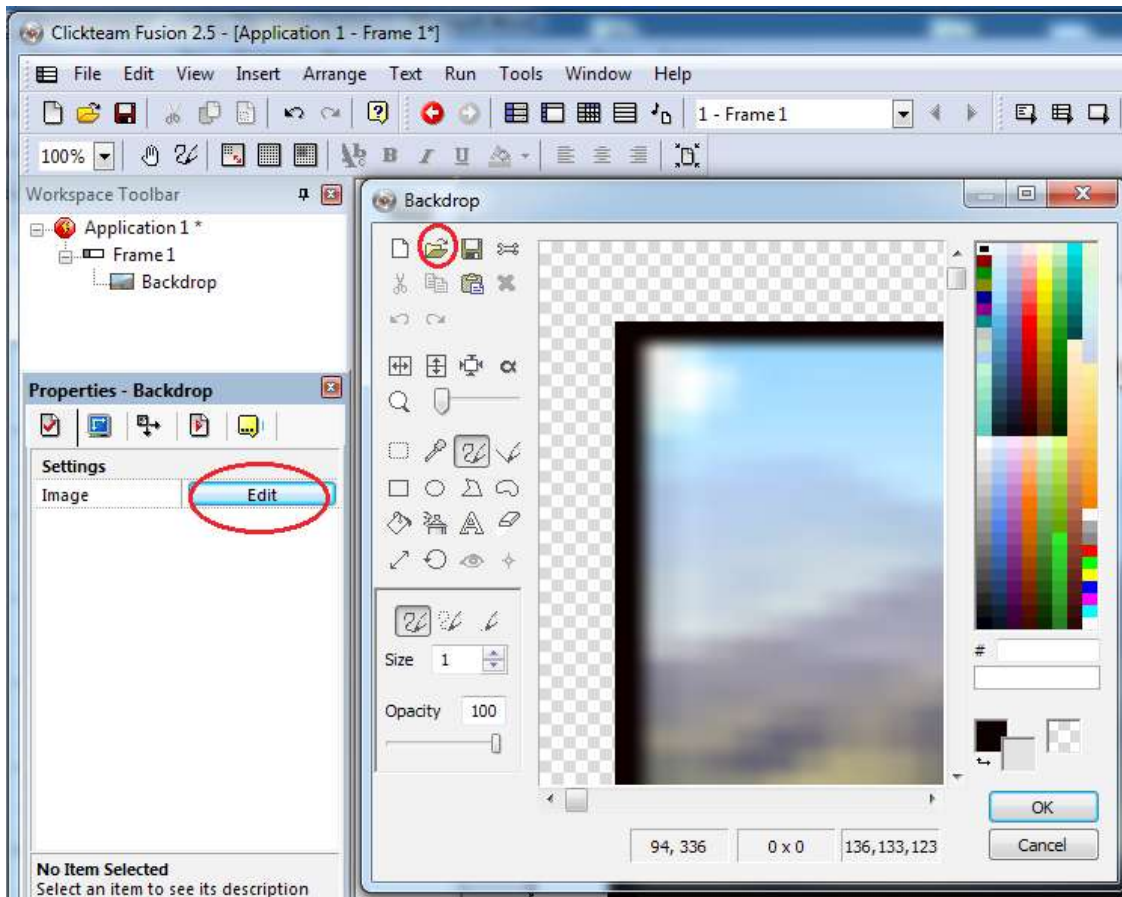
RClick в області фрейма Insert Object -> Background -> Backdrop -> OK

З'явиться об'єкт фону. Клікніть по ньому так, щоб навколо нього з'явилися обмежувальні точки. Перемістіть його в лівий верхній кут і мишею за точки розтягніть на всю область фрейма.

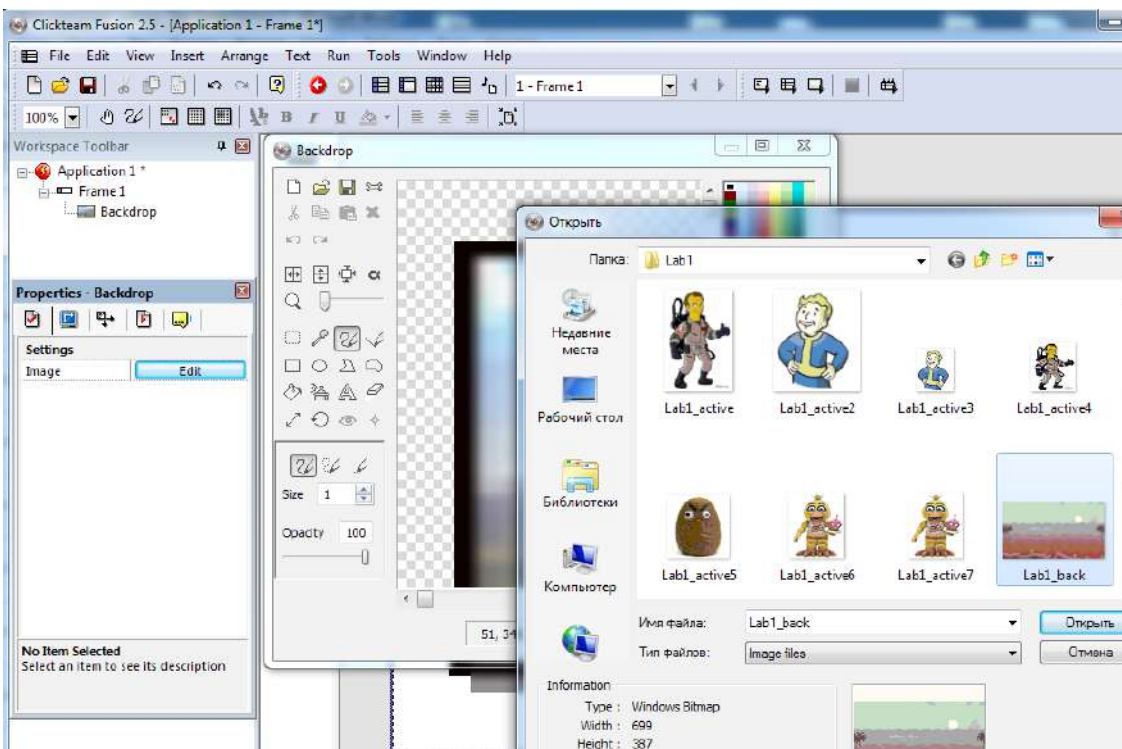


Лабораторні роботи

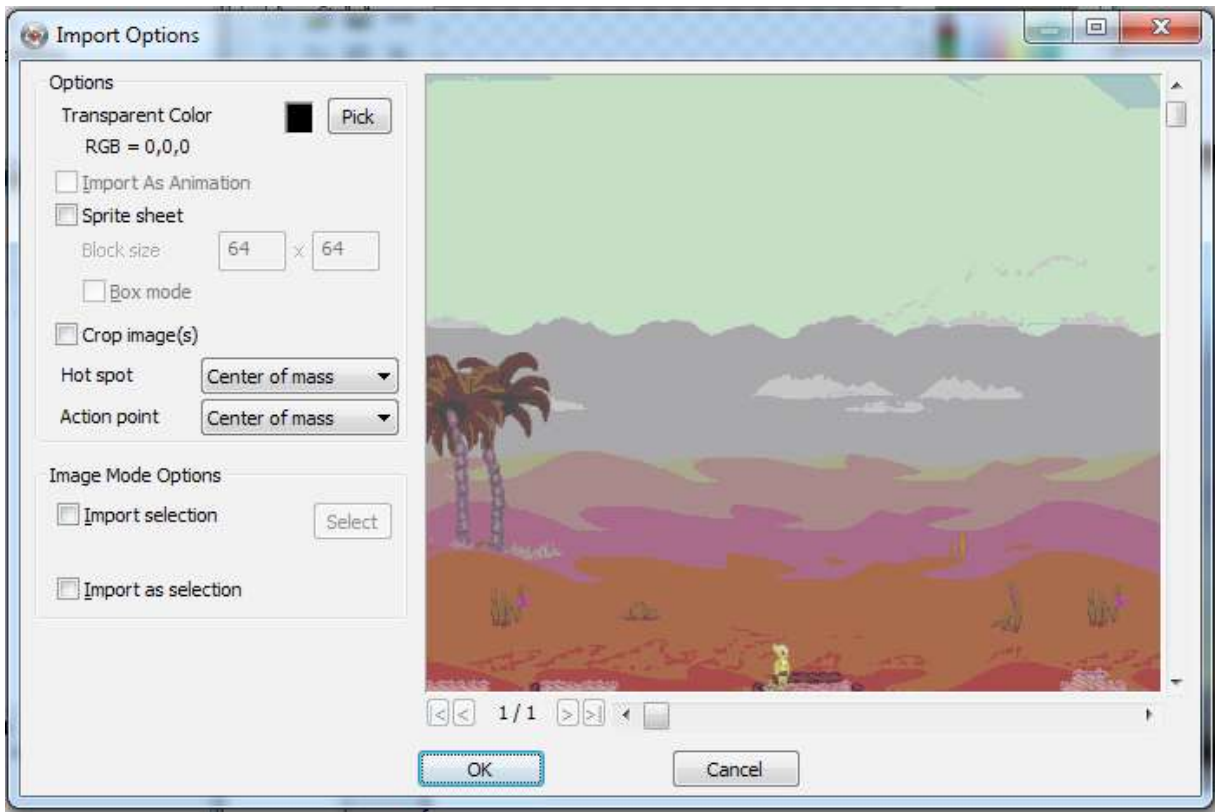
Замість наявного малюнка можна завантажити свій малюнок. Для цього у вікні властивостей об'єкта Backdrop (Properties – Backdrop) клікнути по кнопці Edit, з'явиться вікно редактора зображення об'єкта:



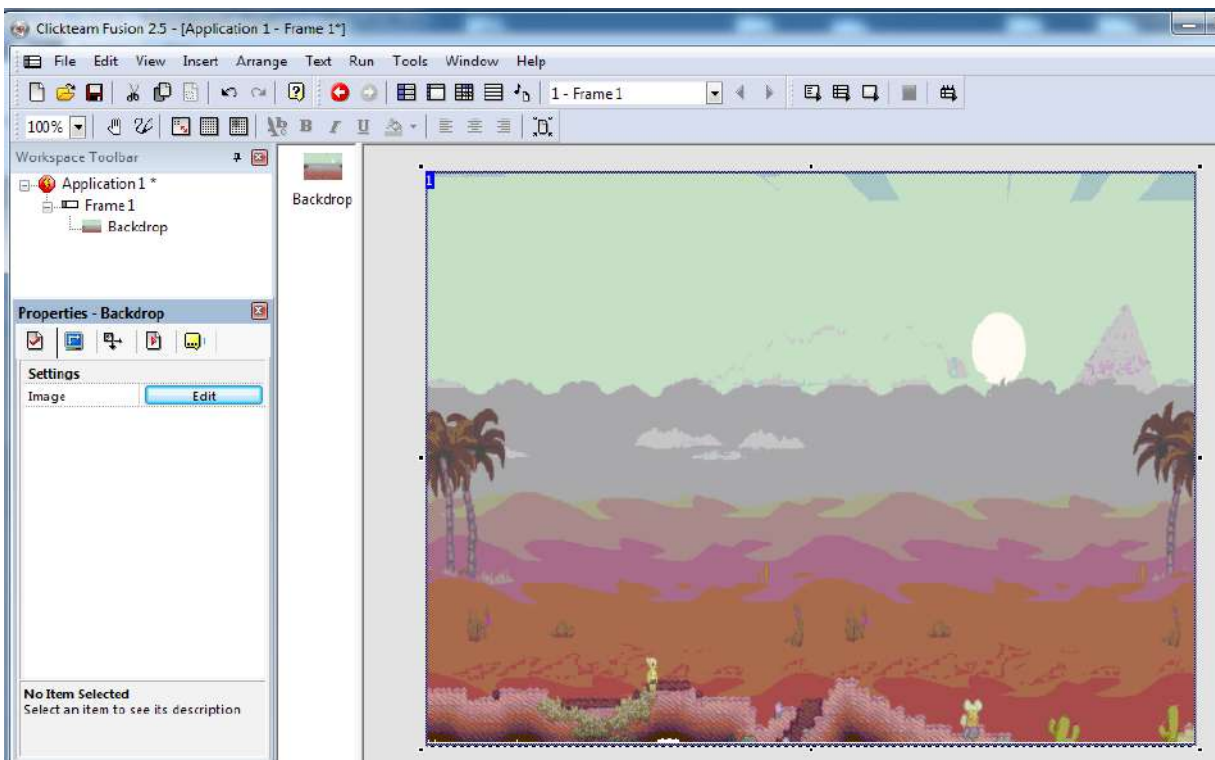
У вікні редактора зображення об'єкта клікнути по кнопці Import (її виділено червоним колом).



У вікні «Відкрити» вибрати файл з розширенням .bmp і клікнути по кнопці «Відкрити». З'явиться вікно Import Options, в ньому виконати налаштування зображення і натиснути ОК:

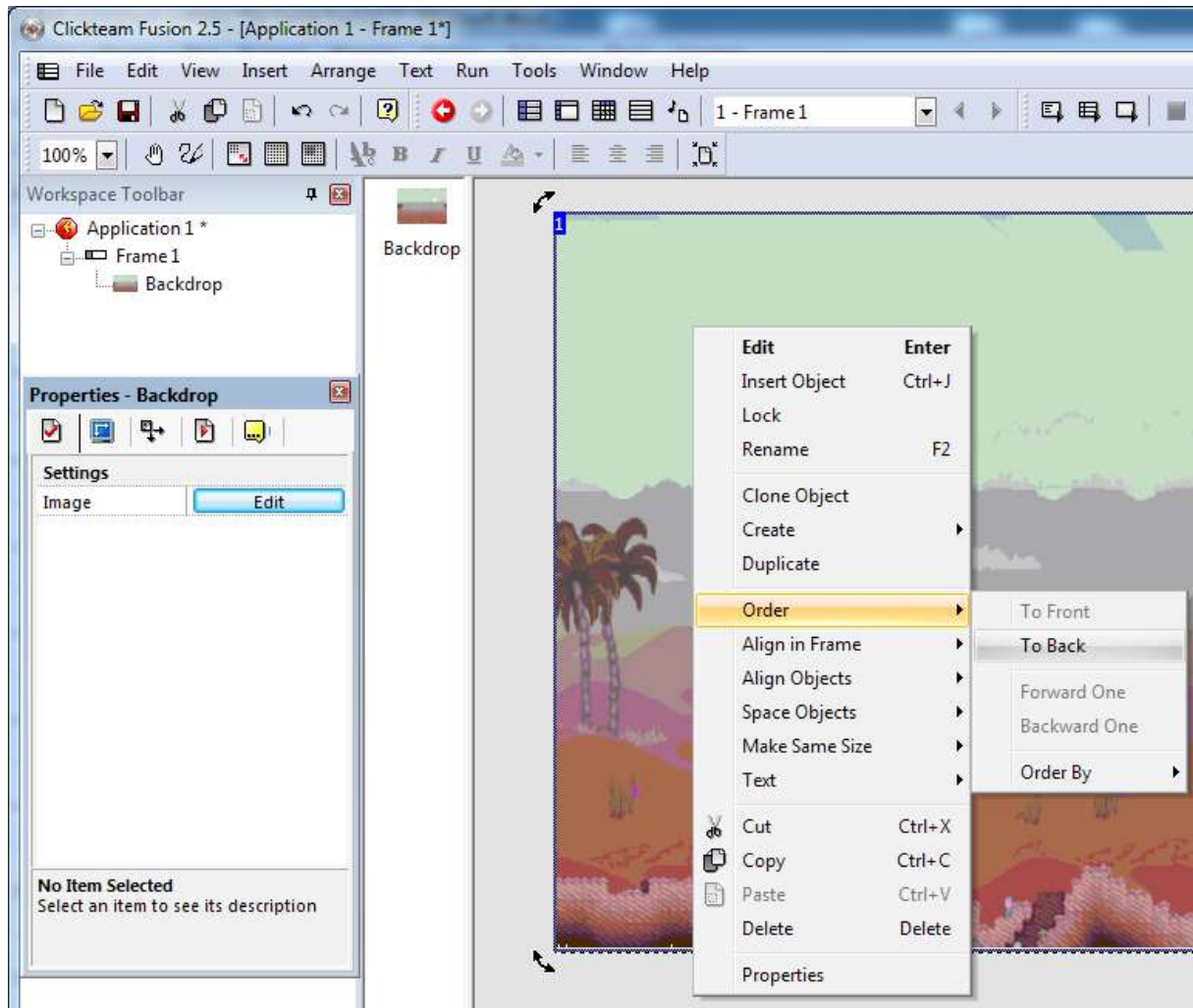


Після повернення у вікно редагування об'єкта Backdrop натиснути ОК. Відбудеться повернення у вікно редагування фрейма. Якщо зображення не збігається за розміром з розміром фрейма, можна його розтягнути мишею:



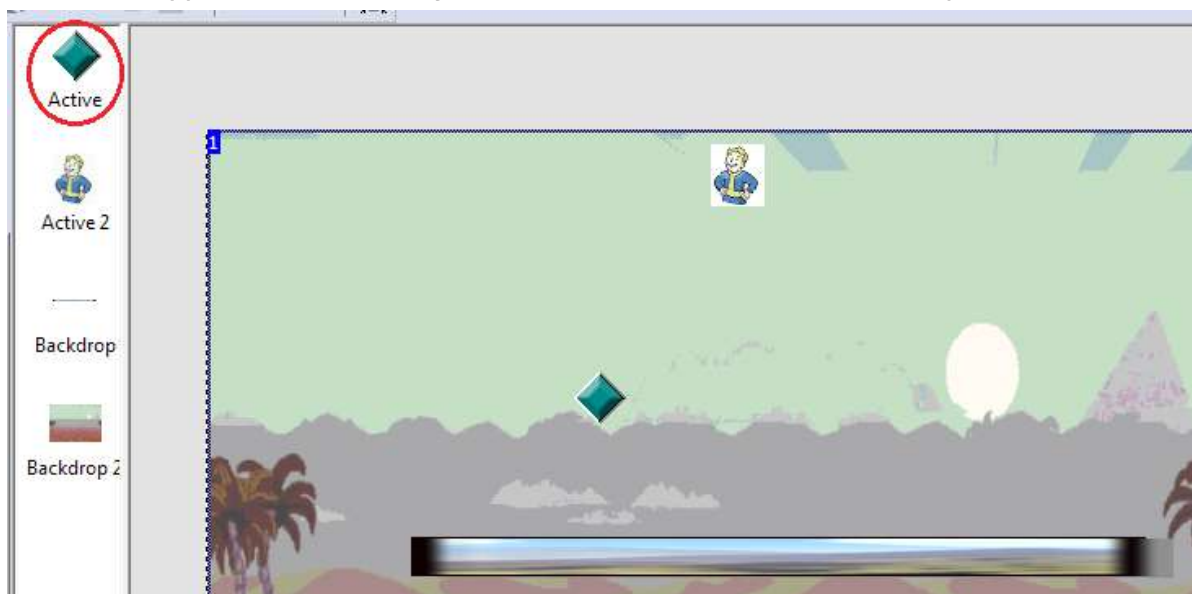
Лабораторні роботи

Якщо фоновий малюнок було додано або завантажено після додавання інших об'єктів, то потрібно вказати, що він розташовується на задньому плані – RClick в області фрейма, Order -> To Back:

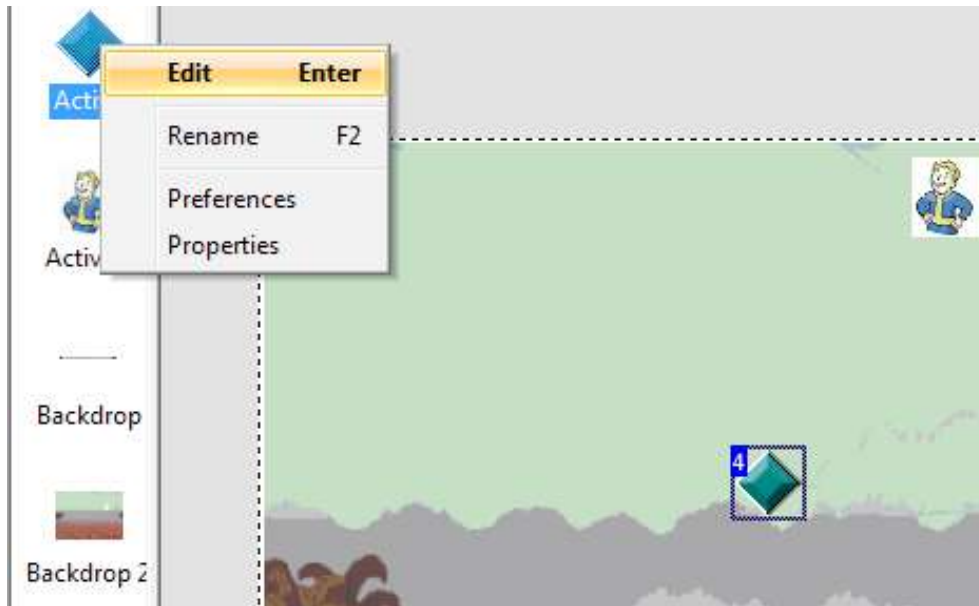


НАЛАШТУВАННЯ ЗОБРАЖЕННЯ АКТИВНОГО ОБ'ЄКТА

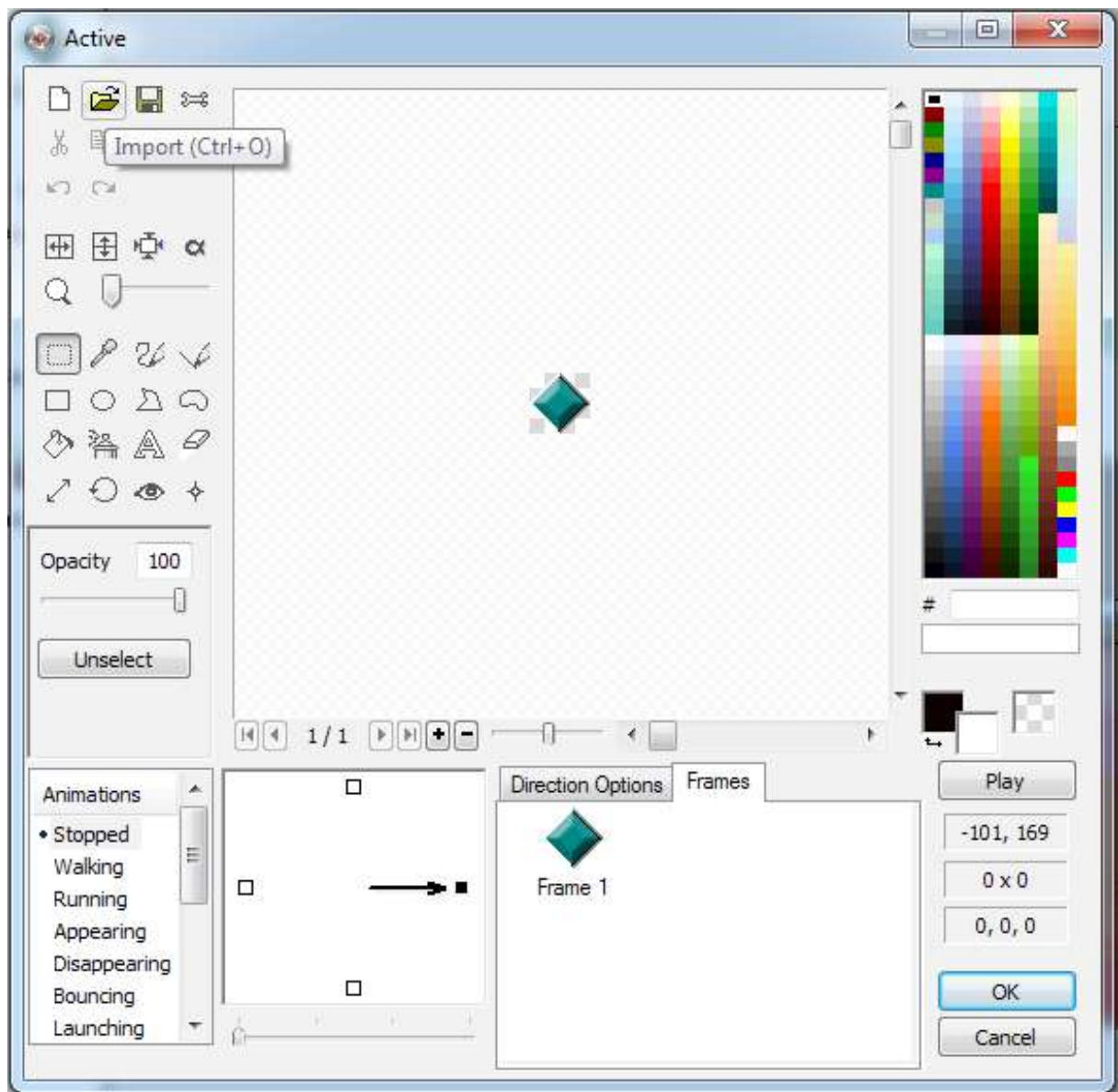
RClick області фрейма, Insert Object -> Active. З'явиться зелений ромб:



Вибрати зліва Active, на ньому RClick -> Edit:

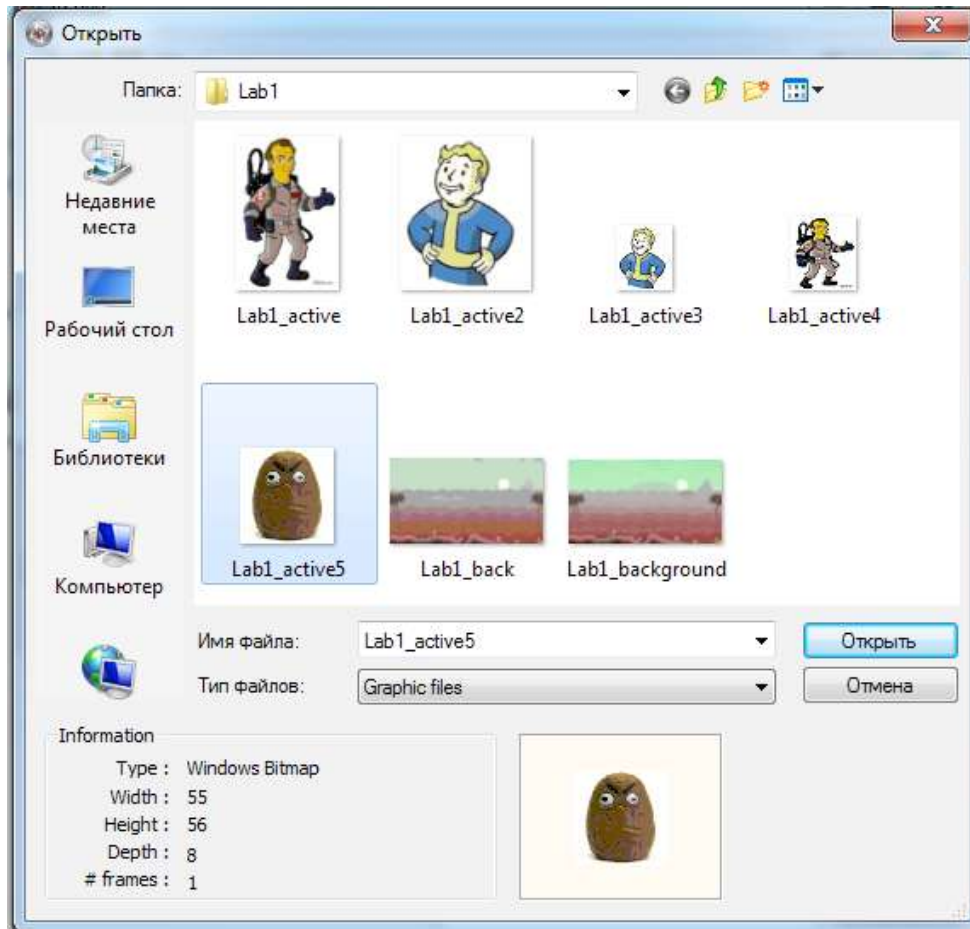


Відкриється вікно редагування виду об'єкта:



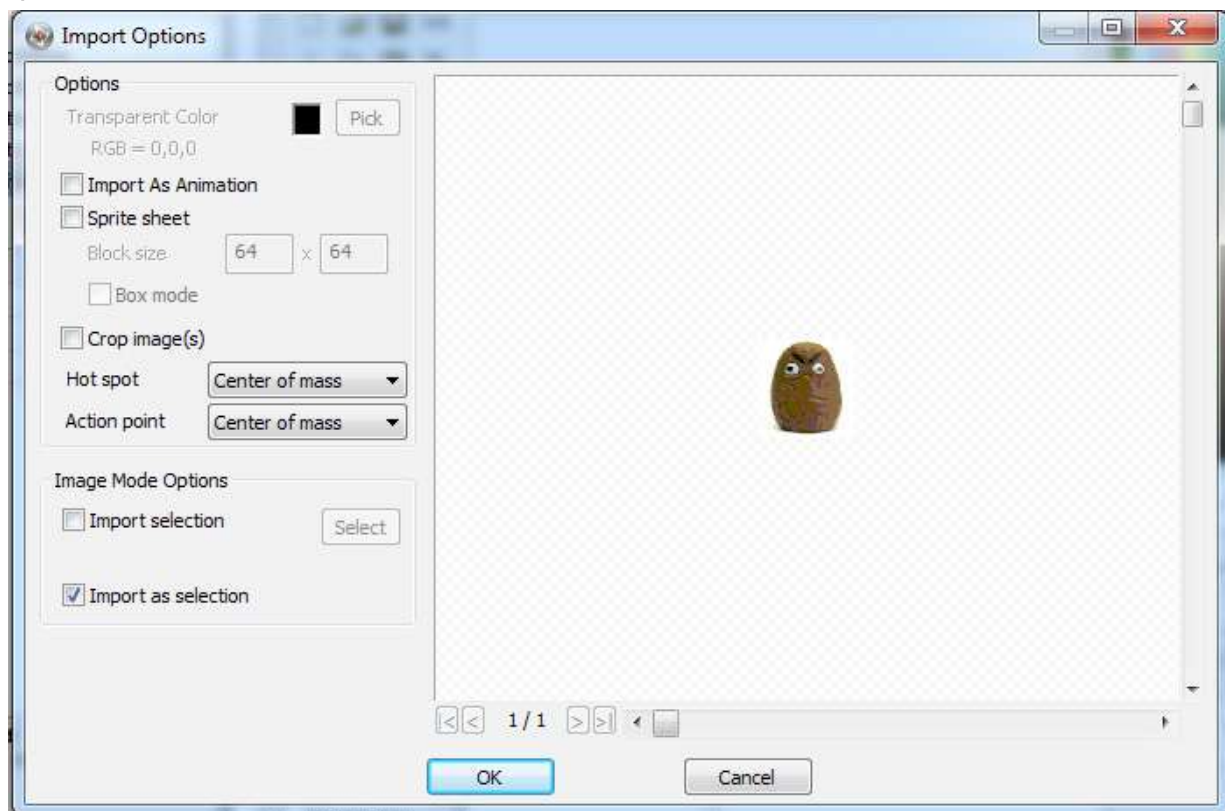
Лабораторні роботи

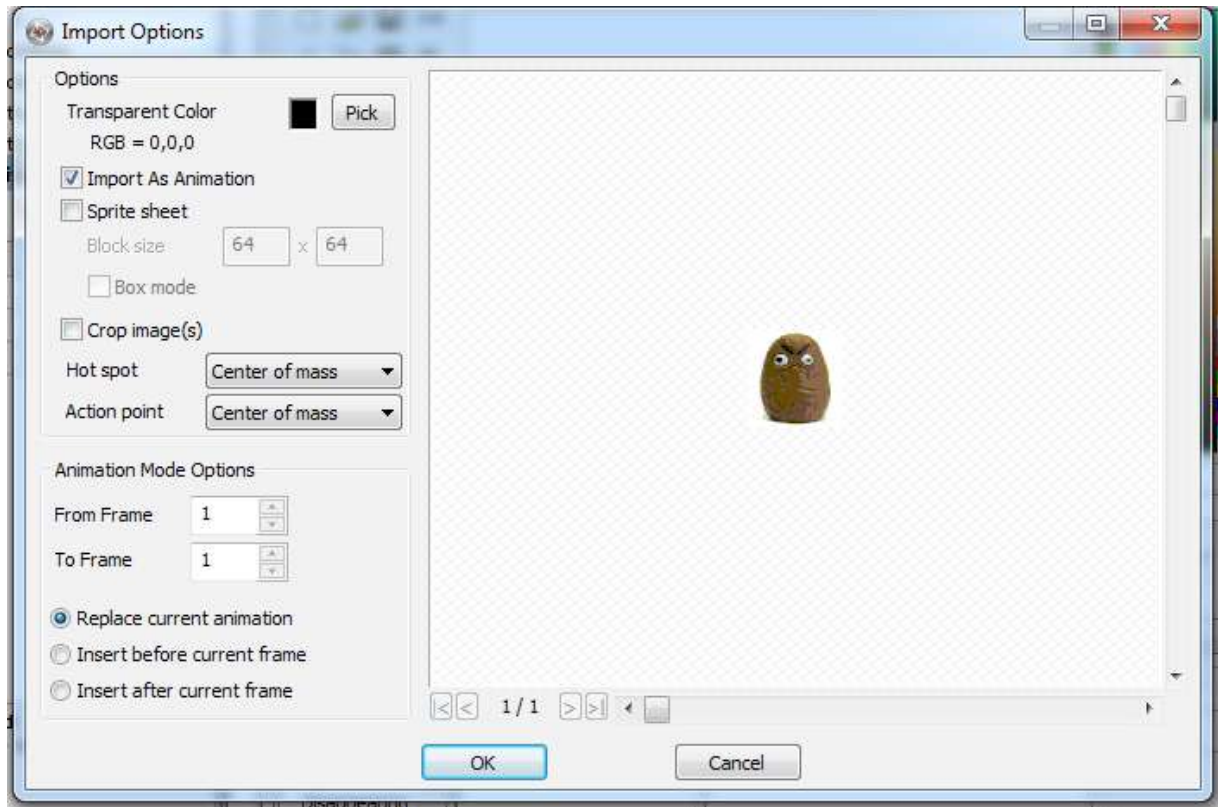
У цьому вікні вибрати кнопку Import, відкриється вікно вибору файла:



У ньому вибрати: файл -> Відкрити.

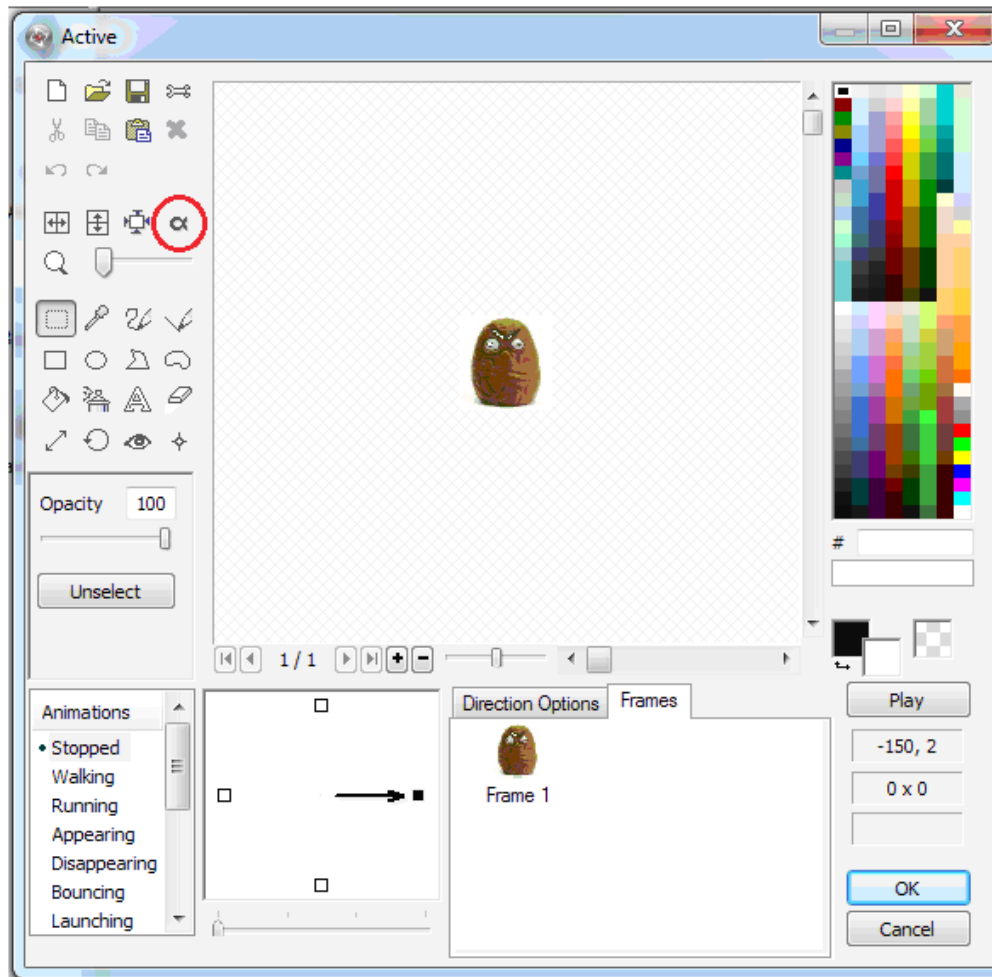
Відкриється вікно:





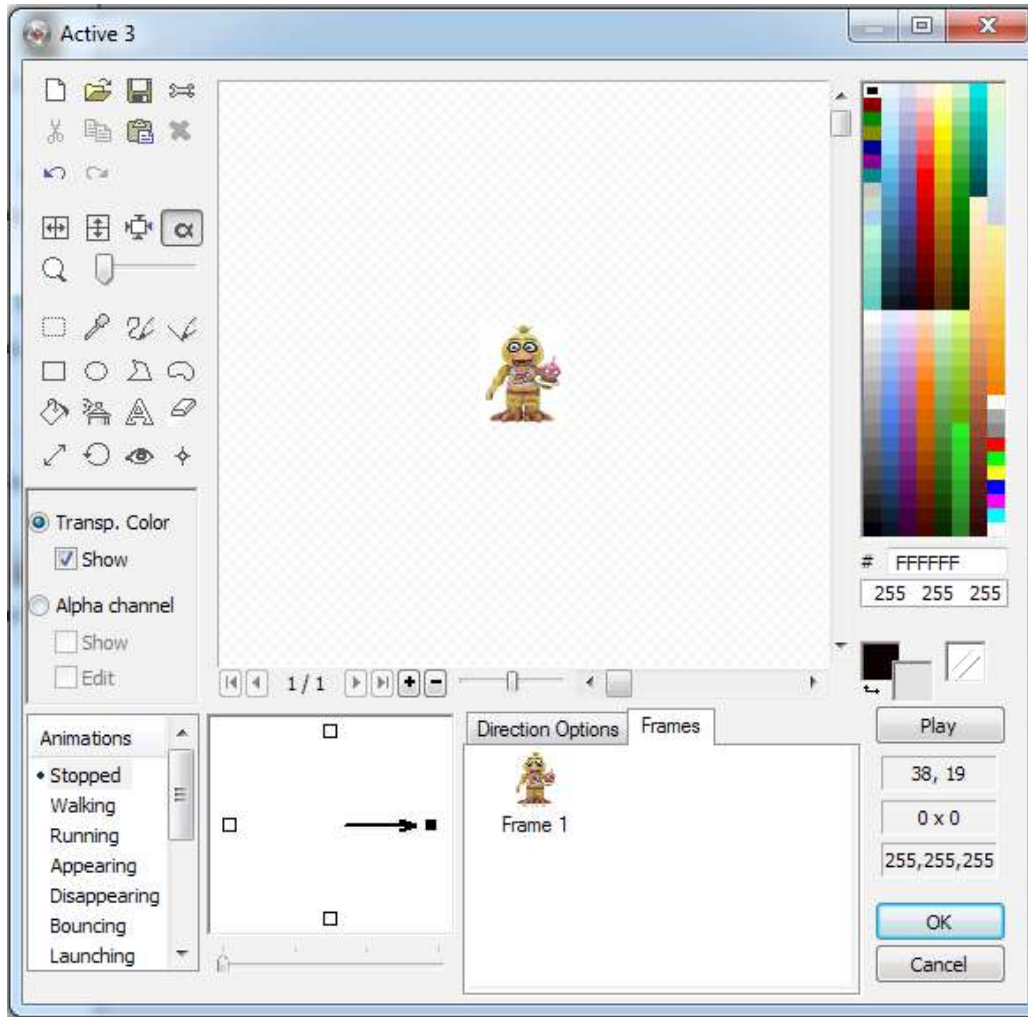
OK.

Повернення до вікна:

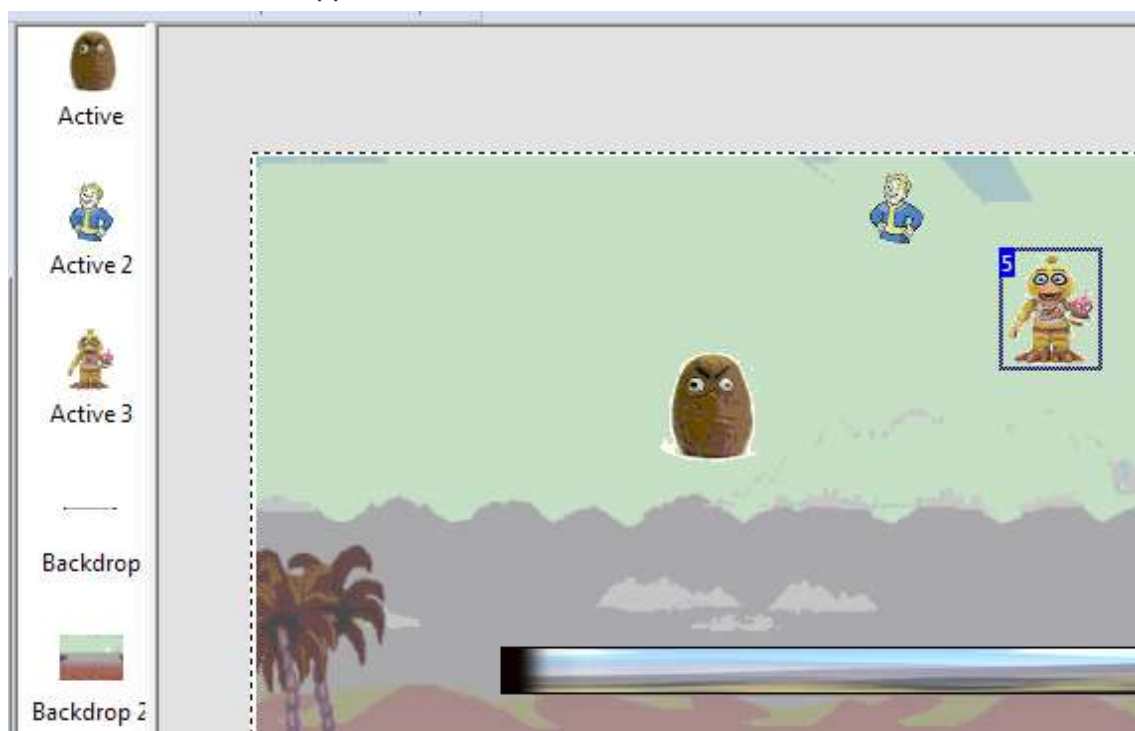


Лабораторні роботи

Зробити прозорим все навколо фігурки: вибрати Transparency (обведено червоним), піпеткою клікнути на тлі навколо фігурки:



ОК. Об'єкт з'являється на фреймі:



ЛАБОРАТОРНА РОБОТА №3

АНІМАЦІЯ ПЕРСОНАЖІВ У ДОДАТКУ CLICKTEAM FUSION

МЕТА РОБОТИ:

- вивчити можливості редактора анімації;
- дослідити варіанти руху активних об'єктів;
- налаштувати анімації ігрових персонажів.

ПОРЯДОК РОБОТИ.

1. Вивчити інструкцію до лабораторної роботи.
2. Відкрити програму редактора Clickteam Fusion.
3. Відкрити проект, створений в лабораторній роботі №2.
4. По черзі задавати різні параметри руху будь-якого зі створених активних об'єктів і досліджувати особливості руху і управління об'єктом.
5. Для кожного створеного за сценарієм об'єкта вибрати і налаштувати його рух.
6. Виконати анімацію ігрових об'єктів.
7. Перевірити працездатність проекту.
8. Зберегти створений проект.
9. Продемонструвати працездатність проекту викладачеві.
10. Закрити додаток Clickteam Fusion.
11. Оформити звіт. Захистити роботу.

ВЛАСТИВОСТІ РУХУ

Властивості Руху (Movement) вказують, коли і як рухається об'єкт на екрані. Clickteam Fusion 2.5 забезпечує різні види рухів:

РУХ НА ОСНОВІ ФІЗИКИ

Ці види руху дозволяють створювати реалістичні ігри, в яких використовуються принципи фізики (гравітація, зіткнення і т. д.). До них належать:

Static – статичний;

Eight Directions – у восьми напрямках;

Space Ship – космічний корабель;

Bouncing Ball – стрибки м'яча;

Race-Car – гоночна машина;

Platform – по платформі;

Spring – пружина;

Axial – аксіальний, у напрямку осі.

НЕФІЗИЧНИХ РУХ

Ці види руху не використовують фізичний движок для роботи: об'єкти рухаються менш реалістично, ніж з фізичними рухами. Вони можуть використовуватися в простих іграх, де симуляція не обов'язково повинна бути повною.

Переміщення по траєкторії може бути дуже корисним у фізичному середовищі: воно дозволяє змусити персонажів переміщуватись шляхом – наприклад, охоронців, які очікують на ворога в замку.

Static – статичний;

Bouncing Ball – стрибки м'яча;

Path – по заданій траєкторії;

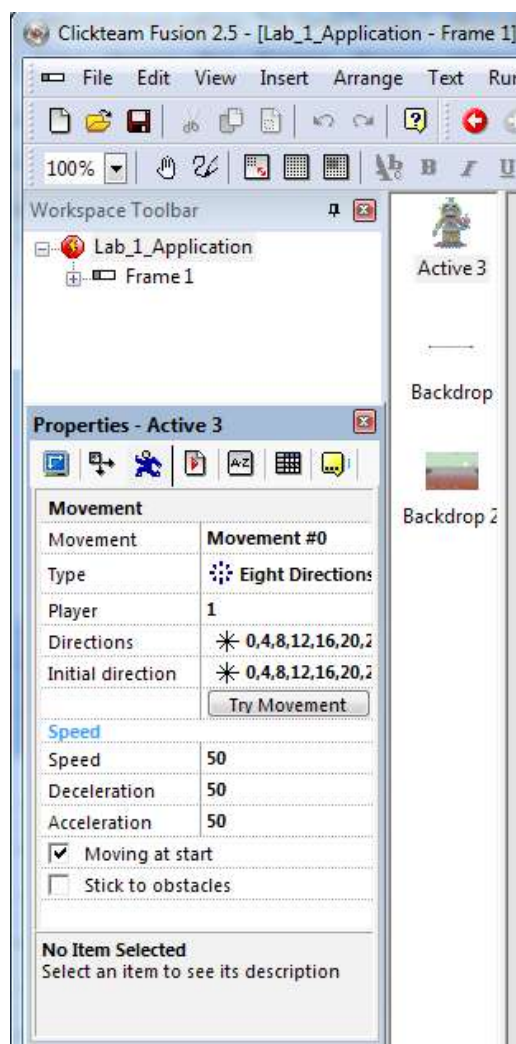
Mouse – керований мишею;

Eight Directions – у восьми напрямках;

Race-Car – гоночна машина;

Platform – по платформі.

Переміщення об'єкта змінюється в залежності від обраного типу руху. Тип руху задається на панелі властивостей об'єкта:



MOVEMENT – РУХ

У Clickteam Fusion 2.5 і об'єкти можуть використовувати більше одного руху. Наприклад, перший рух може бути рухом Path, а друге – рухом Bouncing ball. Конкретна дія в редакторі подій вказує на перехід від одного руху до іншого. Цей рядок переміщення дозволяє додавати рух у список. Він відкриває просте діалогове вікно, в якому ви можете додавати, видаляти, перейменовувати рух.

Для перших додатків може не знадобитися визначати більш одного руху для об'єктів. Цю властивість можна ігнорувати. Але при створенні більш складного додатка ця властивість може знадобитися.

TYPE – ТИП

Цей блок дозволяє вибрати тип поточного руху зі списку, зазначеного вище.

INITIAL DIRECTION – ПОЧАТКОВИЙ НАПРЯМОК

Коли об'єкт створюється на фреймі, він має початковий напрямок руху. Властивість Type дозволяє вказати початковий напрямок руху об'єкта. Натисніть на номер, щоб відкрити селектор напрямку і виберіть один напрямок. Ви можете вибрати більше ніж один напрямок, в цьому випадку Clickteam Fusion 2.5 вибере випадковий напрям між обраними.

TRY MOVEMENT – СПРОБУВАТИ РУХ

Ця кнопка дозволяє побачити рух у роботі. Це дасть вам змогу оцінити швидкість і початковий напрямок руху об'єкта. Натисніть ESCAPE або натисніть кнопку QUIT, щоб вийти з вікна.

THE NON-PHYSICAL MOVEMENTS – НЕФІЗИЧНИЙ РУХ

Рух Path може виявитися дуже корисним, навіть в грі з фізичним управлінням, щоб об'єкти прямували заздалегідь визначеним шляхом в ігровій сцені.

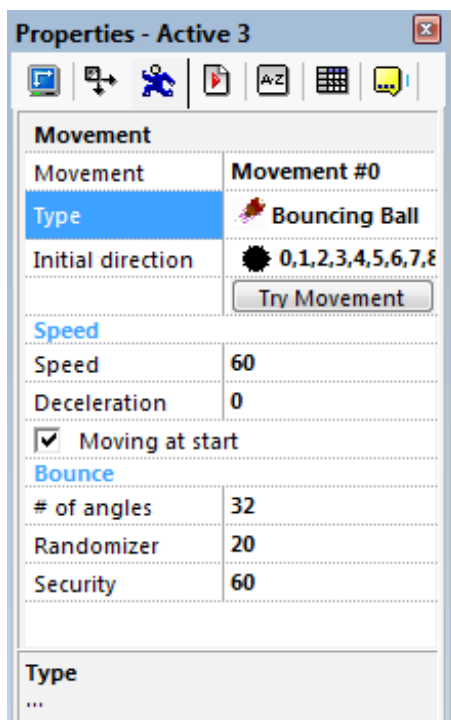
Замовчуванням будь-який об'єкт з нефізичним рухом також створюється в світі, і його положення примусово встановлюється в кожному циклі. Це забезпечує певну взаємодію з іншими об'єктами, але необхідно знати, що ця взаємодія є обмеженою.

ВЛАСТИВОСТІ РУХУ BOUNCING BALL (СТРИБКИ М'ЯЧА)

Рух «стрибки м'яча» змушує об'єкт рухатися як м'яч, який відскакує від перешкоди. Це дуже зручний рух, який можна використовувати для різних об'єктів, а не тільки для м'ячів (прикладі – вороги, цілі і т. д.).

Рух «стрибки» м'яча має такі властивості:

Speed – швидкість. Встановлює швидкість, з якою буде переміщуватися об'єкт, від 0 (статичне положення) до 100 (повна швидкість).



Deceleration – уповільнення. Встановлює величину тертя, що працює проти об'єкта. За замовчуванням, уповільнення встановлюється на нуль, так що м'яч буде скакати на поверхні без тертя. Якщо ви збільшуєте уповільнення, об'єкт буде втрачати швидкість під час переміщення і в кінцевому підсумку зупиниться.

Moving at start – переміщення на початку. Увімкніть прапорець, якщо об'єкт має переміщатися під час запуску програми.

Bounce – відскік

Number of angles – кількість кутів. У цьому полі можна вибрати 3 варіанти: 8, 16 і 32. Ці значення показують кількість дозволених кутів відскоку для об'єкта. Чим більше кутів ви виберете, тим плавніший ефект відскоку.

Randomizer – випадковий відскік

Дає м'ячу можливість відскакувати в різних випадкових напрямках. Чим вище значення, тим більш різноманітний ефект.

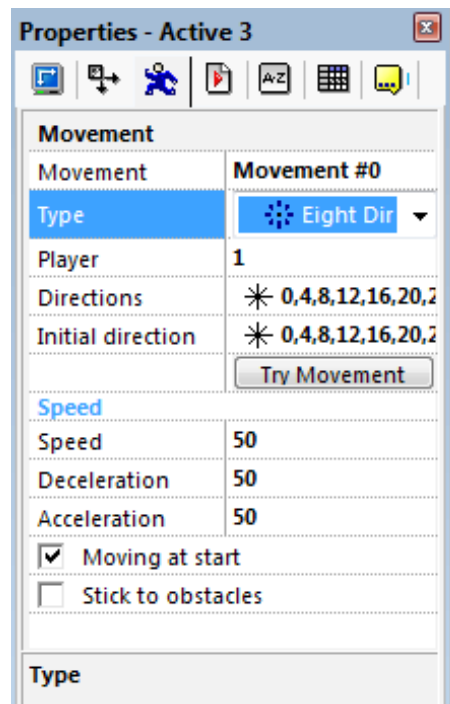
Security – безпека. Запобігає тому, що об'єкт застрягне в циклі відскоків шляхом застосування випадкового відскоку після певної кількості схожих відскоків.

ВЛАСТИВОСТІ РУХУ EIGHT DIRECTIONS MOVEMENT (ВІСІМ НАПРЯМКІВ)

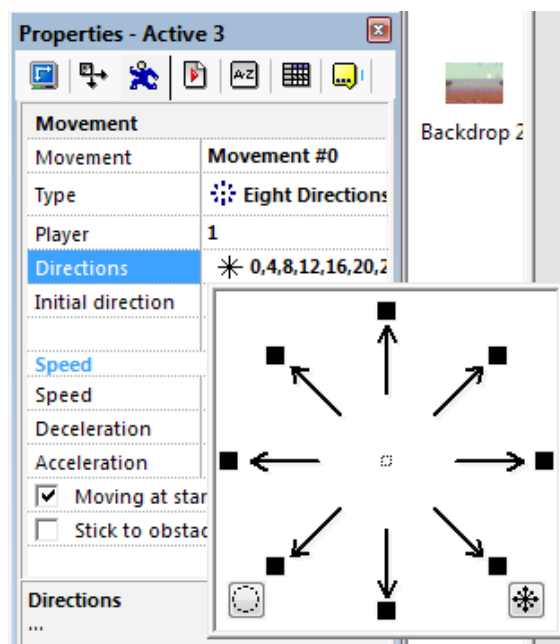
Рух Eight Direction Movement управляється джойстиком або клавіатурою. Це простий рух: об'єкт піднімається, коли гравець натискає вгору, вниз, убік і по діагоналі.

Player- Гравець. Вибір номера гравця (від 1 до 4), якому буде призначено рух.

Directions- Напрямки. Встановлюються дозволени напрямки руху. При натисканні на цифри відкривається селектор напрямку:



За допомогою миші треба вибрати або скасувати необхідні напрямки.



Initial direction- Початковий напрямок. Завдання початкового напрямку руху об'єкта при запуску фрейма. Початковий напрямок вибирається за допомогою миші.

Speed – Швидкість руху об'єкта. Встановлюється максимальна швидкість руху об'єкта.

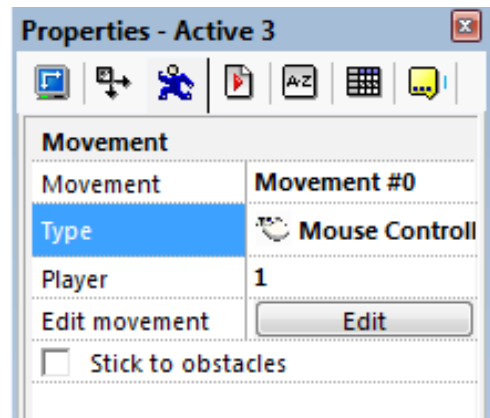
Deceleration – Уповільнення. Встановлюється гранична швидкість руху об'єкта, при якій він зупиняється.

Acceleration – Прискорення. Вказується коефіцієнт прискорення об'єкта, тобто, як швидко об'єкт досягне максимальної швидкості.

Moving at start- Рух спочатку. Якщо прапорець знято, об'єкт при початку фрейма буде нерухомим.

ВЛАСТИВОСТІ РУХУ THE MOUSE MOVEMENT (ПЕРЕМІЩЕННЯМ МИШКИ)

При русі The Mouse Movement, як випливає з його назви, об'єкт переміщується за допомогою миші. Під час роботи програми покажчик миші Windows зникає, і об'єкт буде переміщатися за допомогою миші. Можна обмежити зону, в якій рухається об'єкт.



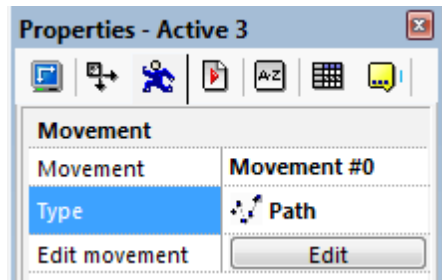
Player- гравець. Вказує гравця, що буде використовувати рух миші. Для фрейма може бути визначений тільки один гравець із заданим номером.

Edit – редагування. Натискання кнопки Edit дозволяє налаштувати зону, в якій об'єкт буде переміщатися. У редакторі відображається рамка, невелике поле з параметрами і прямокутник з квадратами для зміни розміру області. Прямокутник обмежує область переміщення. Після завершення редагування слід натиснути ОК.



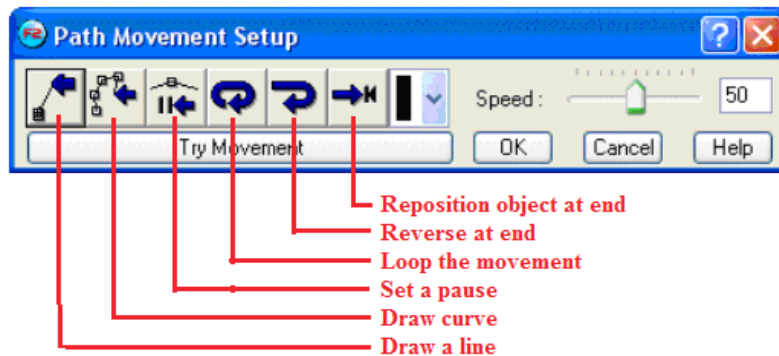
ВЛАСТИВОСТІ РУХУ THE PATH MOVEMENT (ЗАДАНИМ ШЛЯХОМ)

Рух The Path Movement змушує об'єкт слідувати заданому шляху. Цей шлях на фреймі малюють за допомогою миші.



Налаштування руху заданим шляхом здійснюється через діалогове вікно, що відкривається по кнопці EDIT.

Рух по шляху вказує лінія, що складається з різних сегментів. Кожен сегмент відділений від іншого вузлом. Натискання на вузол дозволяє вибрати його для розміщення на фреймі. Можна намалювати рамку для вибору декількох сегментів одночасно.



Draw line – пряма лінія. Дозволяє намалювати пряму лінію на шляху.

Draw curve – крива лінія. Слід натиснути цю кнопку і відзначити курсором на фреймі кінець відрізка. Повторити ці дії для прокладення потрібного шляху. Всі вузли визначаються автоматично. Для завершення кривої двічі клікнути лівою кнопкою миші.

Set a pause – встановити паузу. Вибрати вузол, потім натиснути цю кнопку – відкриється невелике діалогове вікно для введення тривалості паузи для цього вузла. Під час виконання об'єкт зупиняється на заданий час на цьому вузлі.

Loop the movement – рух починається і завершується в одній точці.

Reverse at end – коли об'єкт досягає кінця шляху, він повертається назад до першого вузла тим же шляхом.

Reposition object at end – коли об'єкт досягає кінця шляху, він переміщається в точку початку руху.

Speed slider – повзунок швидкості. Вибрати вузол або групу вузлів, встановити повзунком швидкість руху для цих вузлів.

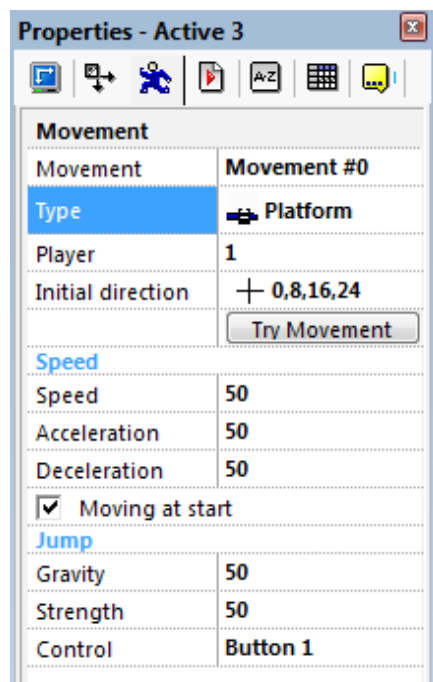
Контекстне меню відкривається при натисканні на вузол правою кнопкою миші. Команди контекстного меню:

- **Draw Line** – додає до вузла нову лінію.
- **Record mouse** – дозволяє перемістити вибраний вузол на нове місце.
- **Set pause** – відкриває невелике діалогове вікно для введення затримки: під час виконання об'єкт буде чекати задану кількість часу на цьому вузлі.
- **Set speed** – відкриває невелике діалогове вікно для встановлення швидкості об'єкта на цьому вузлі.
- **Set name** – дозволяє задати ім'я вузла. Іменованний вузол можна використовувати під час виконання: можна задати умову для визначення, чи досяг об'єкт того чи іншого вузла на шляху, і виконати відповідні дії.
- **Delete** – видаляє заданий вузол.

ВЛАСТИВОСТІ РУХУ THE PLATFORM MOVEMENT (ПО ПЛАТФОРМИ)

Рух Platform Movement змушує об'єкт рухатися як персонаж по платформі. Це складний рух, який виглядає краще, коли об'єкт дотримується певних правил:

- завжди встановлюйте гарячу точку об'єкта між ногами (в середині нижньої лінії);
- переконайтеся, що фігури анімації об'єкта не дуже відрізняються за розміром, інакше об'єкт може застрягнути в зіткненнях;
- визначте анімацію вгору і вниз як сходження і спуск;
- визначте всі інші анімації, такі, як присідання, вставання, стрибок.



Player – гравець. Вказує, до якого гравця (від 1 до 4) відноситься анімація.

Initial direction – початковий напрямок. Відкриває селектор напрямків для вибору напрямку, в якому об'єкт буде створений при запуску фрейма.

Speed – швидкість, – вказує максимальну швидкість руху об'єкта.

Acceleration – уповільнення, – вказує коефіцієнт прискорення руху, тобто швидкість, з якою об'єкт переходить від статичного положення до максимальної швидкості.

Deceleration – уповільнення, – вказує коефіцієнт уповільнення руху, – швидкість, з якою об'єкт буде уповільнювати рух від максимальної швидкості до зупинки.

Moving at start – рух при запуску, – якщо не зазначено, об'єкт буде статичним при запуску фрейма.

Jump/Gravity – стрибок/гравітація, – вказує коефіцієнт сили тяжіння, прискорення, яке змушує об'єкт опускатися. Від 0 (без сили тяжіння) до 100 (дійсно сильна гравітація).

Jump/Strength – стрибок/сила, вказує силу стрибка, тобто початкову силу, яка змушує об'єкт підніматися при стрибку. Висота стрибка об'єкта залежить як від сили тяжіння, так і від сили стрибка, – чим більша гравітація, тим більша сила стрибка на одній висоті.

Jump/Control – клавіші управління стрибком:

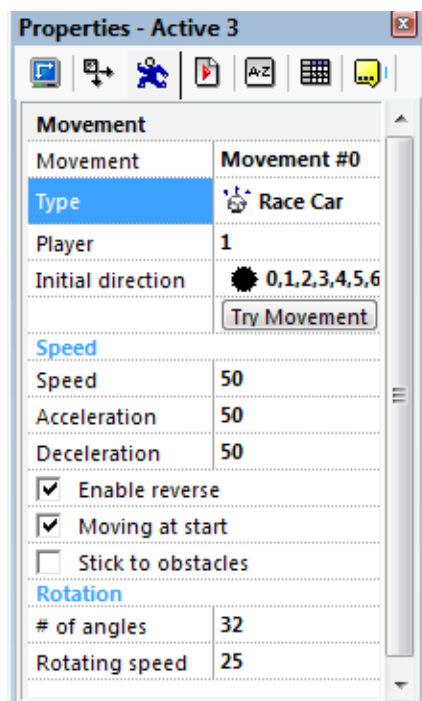
No jump – Об'єкт не стрибає;

Up + Left / Right arrow – Об'єкт стрибає при одночасному натисканні на стрілки Вгору + Вліво / Вправо;

Button 1, Button 2 – Кнопки на джойстику.

ВЛАСТИВОСТІ РУХУ THE RACE CAR MOVEMENT (ГОНОЧНИЙ АВТОМОБІЛЬ)

Рух Race Car переміщує такий об'єкт, як автомобіль, при погляді згори. Ви можете обертати об'єкт за допомогою лівої і правої клавіш, прискорювати клавішею вгору і сповільнювати клавішею вниз. Цей рух працює краще, коли гаряча точка об'єкта встановлена в центрі, і коли об'єкт має всі 32 кути повороту.



Player – гравець. Вказує, до якого з гравців (від 1 до 4) застосовується анімація.

Initial direction – початковий напрямок. Відкриває селектор напрямку для вибору напрямку, в якому об'єкт буде створений при запуску фрейма.

Speed – швидкість. Вказує максимальну швидкість руху об'єкта.

Acceleration – прискорення. Вказує коефіцієнт прискорення руху, швидкість, з якою об'єкт переходить від статичної до максимальної швидкості.

Deceleration – уповільнення. Вказує коефіцієнт уповільнення руху, швидкість, з якою об'єкт буде рухатися від максимальної швидкості до зупинки.

Enable reverse – дозволяється рух в зворотному напрямку з тією ж швидкістю.

Stick to obstacles – обминання перешкод.

Number of angles – кути повороту машини – від 4 до 32.

Rotating speed – швидкість повороту. Вказує, наскільки швидко автомобіль повертається, коли користувач натискає вліво або вправо. Від 0 (без повороту) до 100 (дуже швидко).

ВЛАСТИВОСТІ РУХУ THE STATIC MOVEMENT (СТАТИЧНИЙ РУХ)

При виборі Static movement об'єкт залишається нерухомим. Це не означає, що об'єкт не бере участі в зіткненнях з іншими об'єктами. Статичний об'єкт може бути важливим інструментом в додатку, щоб визначати границі, вказувати двері і т. д.

Статичний рух не має ніяких властивостей.

ВИДИ ФІЗИЧНОГО РУХУ

Починаючи з версії 2.5, Clickteam Fusion 2.5 поставляється з повною версією відомого механізму Box2D. Box2D – це крос-платформений фізичний движок. Він моделює рух і поведінку об'єктів такими, немовби вони перебувають в реальному світі, беручи до уваги масу, щільність, гравітацію і т. д. Результатом є дивовижне і дійсно точне переміщення об'єктів: вони виглядають так, немовби вони були в реальному світі.

Деякі з можливостей Clickteam Fusion 2.5:

- підтримка всіх класичних видів руху в фізиці реального світу;
- розширене переміщення платформи з великою кількістю властивостей і налаштувань;
- повний контроль над фізичним движком, з багатьма діями для охоплення всіх аспектів фізичного світу;
- розширені функції – такі як можливість використання більш одного движка у фреймі.

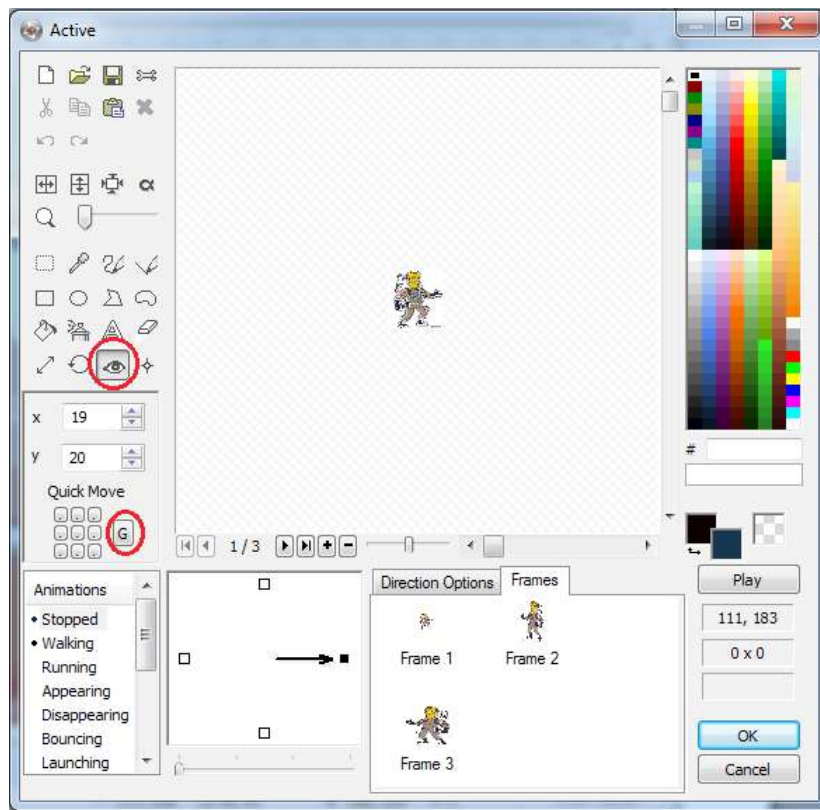
ТИПИ ФІЗИЧНОГО РУХУ (THE PHYSICAL MOVEMENTS):

- Static – Статичний
- Eight directions – Вісім напрямків
- Bouncing ball – Стрибки м'яча
- Race car – Гоночний автомобіль
- Spaceship – Космічний корабель
- Platform – Платформа
- Axial – Осьова
- Background – Задній план
- Spring – Пружина

HOT-SPOTS (ХОТ-СПОТИ)

Щоб об'єкти відображалися правильно в позиції, обчисленій фізичним движком, необхідно встановити «гарячу точку» для всіх зображень об'єкта в правильному положенні. Для руху по платформі гаряча точка повинна бути встановлена в центрі ваги об'єкта.

Клікніть по кнопці  для того, щоб побачити, де розташовується центр ваги об'єкта.



Використовуйте кнопку «G» редактора анімації, щоб розрахувати точне положення центра ваги зображення.

ANIMATION OBJECT (АНІМАЦІЙНИЙ ОБ'ЄКТ)

Об'єкт Animation використовується для відтворення файлів анімації, таких як .FLC або .GIF під час виконання програми. Анімаційні файли генерують зображення, використовуючи серію фотознімків або кадрів анімації. Ці фрейми швидко миготять на екрані, створюючи ілюзію плавного руху. Анімації схожі на маленькі фільми, але вони не супроводжуються звуком. У фреймах можна використовувати об'єкти анімації у вигляді анімованих фільмів, або самостійно призначити об'єктам функції переміщення і положення, щоб вони переміщалися по екрану, як активний об'єкт.

ANIMATION OBJECT – ACTIONS (Дії)

Дії за замовчуванням задаються в наступних підменю:

Movement submenu – Підменю переміщення

Position submenu – Підменю позиції

Direction submenu – Підменю напрямку

Launch an object action – Запуск дії об'єктів

Destroy action – Знищення дії

Visibility – Дії змінюють параметри видимості

- **Make object invisible:** Зробити об'єкт невидимим – приховує поточну анімацію;
- **Make object reappear:** Зробити об'єкт повторно відображуваним – відображає раніше приховану послідовність анімації, починаючи з поточного кадру;
- **Flash object:** Вибраний об'єкт блимає, це досягається при зміні параметрів видимості об'єкта в заданий часовий інтервал;
- **Change ink effect:** Ефект зміни зафарбовування для вибраного об'єкта;
- **Set semi-transparency:** Встановлення напівпрозорості – для вибраного об'єкту змінює коефіцієнт напівпрозорості, між 0 і 128.

ANIMATION- АНІМАЦІЯ

- **Start:** Запуск послідовності анімації;
- **Stop:** Зупинка послідовності анімації;
- **Pause:** Призупиняє послідовність анімації;
- **Resume:** Відновлює послідовність анімації;
- **Next frame:** Перехід до наступного кадру послідовності анімації, якщо послідовність було зупинено або припинено;
- **Previous frame:** Повертається до попереднього кадру послідовності анімації, автоматично зупиняє послідовність анімації;
- **Go to frame:** Перехід до певного кадру послідовності анімації;
- **Change frame duration:** Змінює час, протягом якого кожен кадр відображається на екрані;
- **New file:** Змінює послідовність анімації під час виконання додатка;
- **New file from selector:** Відображає селектор файлів під час виконання програми. Користувач може вибрати нову послідовність анімації з селектора файлів.

ANIMATION OBJECT: CONDITIONS (ОБ'ЄКТ АНІМАЦІЇ: УМОВИ)

Умови для об'єкта анімації встановлюються через наступні підменю:

Collision submenu – Підменю колізії

Movement submenu – Підменю переміщення

Position submenu – Підменю позиції

Direction submenu – Підменю напрямку

Pick or Count submenu – Підменю «Підбір або підрахунок»

Alterable values submenu – Підменю змінених значень

Всі ці підменю підключаються за замовчуванням для рухомих об'єктів.

Visibility – видимість. Такі умови видимості перевіряють видимість того, що вибрано

- **Is «Animation» visible** – перевірка того, чи відображається об'єкт анімації на екрані;
- **Is «Animation» invisible** – перевірка того, чи відображається об'єкт анімації невидимим.

Compare To Current Frame – порівнює номер поточного фрейма, що його відображено в об'єкті Animation, з іншим номером.

End Of Animation? – перевіряє, чи завершено послідовність анімації. Ця умова виконується весь час, коли послідовність анімації знаходиться в процесі, а не тільки у момент досягнення кінця. Цю умову можна комбінувати зі спеціальною умовою «Only One Action When Event Loops» (Тільки одну дію при циклах подій).

Is The Animation Running? – перевіряє, чи запущений об'єкт Animation, тобто чи працює анімація.

Is The Animation Paused? – перевіряє, чи призупинено об'єкт анімації.

User Clicks Within The Animation – перевіряє, чи клікнув користувач у вікні анімації під час виконання програми.

Animation Object: Expressions (Об'єкт анімації: вираження)

Умови дл об'єкта анімації встановлюються через наступні підменю:

Movement submenu – підменю переміщення

Position submenu – підменю позиції

Count submenu – підменю підрахунку

Number Of Current Frame – повертає номер поточного фрейма анімації.

Number Of Frames – витягує загальну кількість фреймів в послідовності анімації.

Width Of Animation – дає ширину фрейма анімаційної послідовності.

Height Of Animation – дає висоту фрейма анімаційної послідовності.

Frame Duration – повертає час, протягом якого кожен фрейм буде відображатися на екрані в мілісекундах (1000 мілісекунд за секунду).

ANIMATION OBJECT: PROPERTIES (ОБ'ЄКТ АНІМАЦІЇ: ВЛАСТИВОСТІ)

ANIMATION – АНІМАЦІЯ

- **Filename:** Містить ім'я файла для відтворення в об'єкті. Тип файла, підтримуваного цим об'єктом, залежить від встановлених графічних фільтрів.

OPTIONS – НАЛАШТУВАННЯ

- **Play from disc** – Відтворення з диска. Файл має зчитуватися з диска по мірі відтворення анімації. Він запобігає завантаженню Clickteam Fusion 2.5 відразу ж перед грою. Якщо цю властивість не вибрано, то при завантаженні великого файла виникають затримки при запуску фрейма.
- **Loop** – Анімація зациклюється, тобто при досягненні кінця запускається спочатку.
- **Hide at start** – Приховує об'єкт при відкритті фрейма.

- **Skip frames** – Пропустити фрейми. Якщо швидкість програми занадто повільна для анімації, ця опція дозволить зберегти темп. Додаток буде стежити за часом анімації і пропускати фрейми, щоб витримати заданий час.
- **Display errors** – Показати помилки. При виникненні помилки відкривається вікно з повідомленням про неї. Корисно при налагодженні програми.
- **Max frames to skip** – Максимальна кількість фреймів для пропуску. Вказується максимальна кількість кадрів для пропуску в один крок. Використовується з властивістю Skip frames.

TRANSPARENT COLOR (DISPLAY PROPERTIES TAB) – ПРОЗОРИЙ КОЛІР (ВКЛАДКА DISPLAY PROPERTIES)

- **First Pixel:** Перший піксель кожного кадру визначатиме прозорий колір, якщо тільки файл анімації вже не містить прозорий колір (наприклад, деякі файли .GIF).
- **Custom:** Дозволяє встановити власний прозорий колір для всієї анімації. Цей колір буде ігноруватися, якщо файл анімації містить прозорий колір (наприклад, деякі файли .GIF).

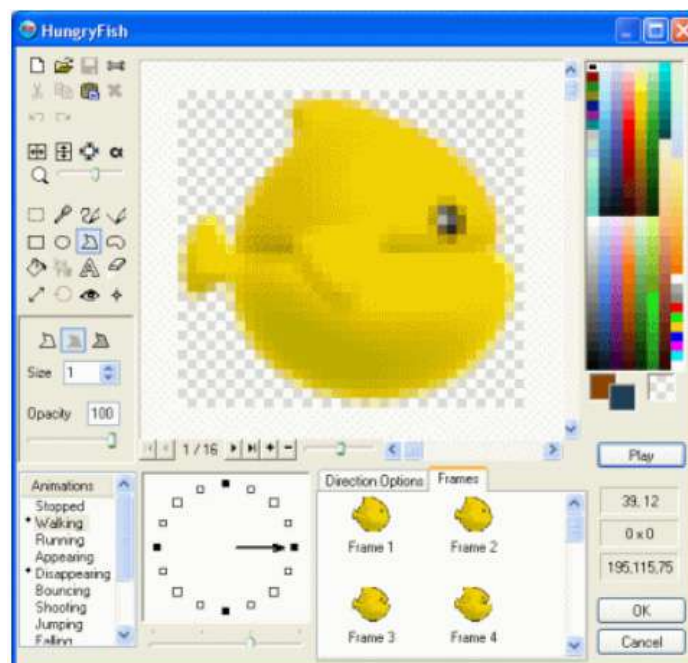
РЕДАКТОР АНІМАЦІЇ

Редактор анімації дозволяє редагувати анімацію активного об'єкта. Він дозволяє створювати, змінювати всі анімаційні зображення, що містяться в цьому об'єкті. Редактор заснований на редакторі зображень.

Редактор анімації відкривається при редагуванні активного об'єкта. Він дозволяє малювати об'єкти і визначати їх анімацію.

Редактор зображень являє собою повний пакет інструментів для малювання, вбудований в Clickteam Fusion 2.5. Він містить всі необхідні інструменти для малювання, імпорту та редагування зображень.

Опис інструментів і правил роботи в області редагування див. в лабораторній роботі №2.



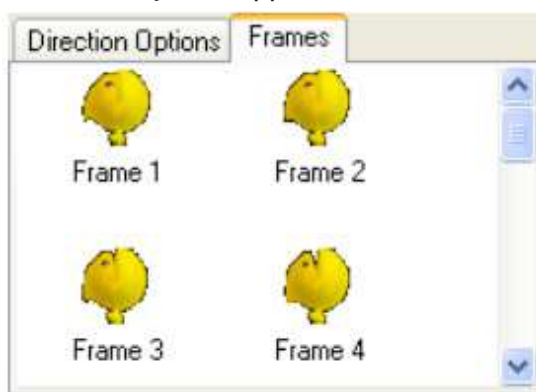
Редактор анімації є розширеною версією редактора зображень. Редактор анімації розширено в нижній частині діалогового вікна. Зліва направо:

- **The animation selector** – Селектор анімації. Активний об'єкт може мати кілька анімацій, і ці анімації використовуються, коли об'єкт переміщується екраном. Анімація складається з анімацій за замовчуванням і призначених для користувача анімацій. Анімація за замовчуванням:
 - **Stopped** – Зупинено. Використовується, коли об'єкт є статичним;
 - **Walking** – Ходьба. Використовується, коли об'єкт рухається повільним темпом;
 - **Running** – Біг. Використовується, коли об'єкт рухається з високою швидкістю;
 - **Appearing** – Поява. Ця анімація буде відтворюватися відразу ж після створення об'єкта;
 - **Disappearing** – Зникнення. Анімація буде відтворюватися при знищенні об'єкта;
 - **Bouncing** – відскакування. Анімація відтворюється, коли об'єкт стрибає на перешкоду;
 - **Launching** – Запуск. Відтворення при запуску об'єкта. Куля створюється в місці розташування точки дії першого зображення анімації;
 - **Jumping** – Стрибки. Використовується при русі по платформі, коли об'єкт стрибає;
 - **Falling** – Падіння. Використовується при русі по платформі, відтворюється, коли об'єкт падає;
 - **Climbing** – Сходження. При русі по платформі, відтворюється, коли об'єкт піднімається по сходах;
 - **Crouch down** – Присідання. Відтворюється при русі по платформі, коли об'єкт присідає;
 - **Stand up** – Вставання. Відтворюється при русі по платформі, коли об'єкт встає;
 - **User animations** – Призначено для анімації, створених користувачем можна визначити необмежену кількість додаткових анімацій, які будуть використовувати в додатку. Щоб створити нову анімацію, клікніть правою кнопкою миші в селекторі анімації і виберіть «Створити».
- **The direction selector** – Селектор напрямку. Розташований праворуч від селектора анімації. Кожна анімація може використовувати 32 різних напрямків. Правильний напрямок, якщо він існує, буде автоматично вибрано і використано підпрограмами руху під час виконання. Якщо напрямку не існує, вибирається найближчий існуючий напрямок. Щоб змінити кількість напрямків, використовуйте повзунок, розташований під селектором напрямку. Він складається з чотирьох налаштувань: 4, 8, 16 і 32 напрямки. 32 напрямки потребують багато пам'яті і повинні бути зарезервовані для основних об'єктів програми. Порожні напрямки відзначені порожнім квадратом в селекторі напрямку, а напрямки, що містять зображення, відзначені заповненою точкою.

При натисканні правою кнопкою миші по одній з заповнених точок відкриється контекстне меню:

- **Cut/Copy/Paste/Delete** – Вирізати/Копіювати/Вставити/Видалити. Для напрямків доступні звичайні функції редагування. Вони копіюють або вставляють все зображення напрямку відразу.

- **Create rotated directions** – Створити поворотні напрямки. Цей параметр дозволяє створювати всі поворотні напрямки анімації одним кліканням миші. Всі напрямки обчислюються шляхом повороту зображень навколо гарячої точки.
- **Create opposite direction** – Створити протилежний напрямок. Створюється протилежний напрямок, об'єкт перевертається навколо гарячої точки зображення.
- **Flip Horizontally** – Горизонтальний поворот. Зображення повертається на 180 градусів по горизонталі.
- **Flip Vertically** – Вертикальний поворот. Зображення повертається на 180 градусів по вертикалі.
- **Invert Frame Order** – Інвертувати порядок кадрів. Порядок проходження фреймів змінюється на зворотний.
- **Direction options tab** – Вкладка «налаштування напрямку». Містить важливі параметри для анімації
- **Lower speed** – Більш низька швидкість. Визначає найбільш низьку швидкість анімації. Це значення показує, наскільки швидко анімація відтворюється, коли об'єкт є статичним. Швидкість – це число від 0 до 100.
- **Higher speed** – Більш висока швидкість. Визначає найбільш високу швидкість анімації, яка використовується при русі об'єкта з максимальною швидкістю. Проміжна швидкість розраховується між мінімальним і максимальним значеннями.
- **Repeat** – Повторення. Вказує, скільки разів анімація повинна відтворюватися. Встановіть прапорець циклу, щоб змусити анімацію відтворюватися постійно.
- **Back to** – Повернутися до. Вказує номер Фрема, починаючи з якого анімація повторюється.
- **Frame tab** – Вкладка Фрейм показує всі фрейми анімації:



RClick на значку фрейма – відкривається меню:

- **Clone frame** – Клонування фрейма. Створюється новий фрейм, з використанням поточного фрейму як шаблону.
- **Zoom** – Масштабування. Відкривається діалогове вікно масштабування, що дозволяє створювати нові фрейми, шляхом збільшення або зменшення поточного фрейма. Нові кадри можна вставити до або після поточного фрейма.
- **Rotation** – Поворот. Створюються нові фрейми, шляхом обертання об'єктів навколо свого гарячого плями. Крім того, нові фрейми можуть бути вставлені до або після

поточного фрейма.

- **Flip horizontally** – повернути по горизонталі.
- **Flip vertically** – повернути по вертикалі.
- **Select all** – Вибрати все. Виділяє всі фрейми анімації.
- **Unselect** – Зняти виділення. Знімає виділення фреймів.
- **Cut/Copy/Paste** – Вирізати/Копіювати/Вставити. Функції редагування також працюють на фреймах.
- **Delete** – Видалення вибраних фреймів
- **The control bar** – Панель управління. Відображається під областю редагування зображення і дозволяє швидко переміщатися по кадрах анімації.



Перелік кнопок панелі управління зліва направо:

- **First frame** – Перший кадр. Відображає перший фрейм анімації;
- **Previous frame** – Попередній кадр. Відображення попереднього фрейму анімації;
- **Frame Number** – Номер кадру. Вказує номер поточного фрейма і загальну кількість фреймів, що містяться в анімації;
- **Next frame** – Наступний кадр. Відображає наступний фрейм анімації;
- **Last frame** – Останній кадр. Відображає останній фрейм анімації;
- **Clone frame** – Клонування фрейма. Створюється новий фрейм, що дублює поточний. Новий фрейм вставляється після поточного фрейма;
- **Delete frame** – Видалити фрейм. Знищує поточний фрейм;
- **Display previous or next frame** – Відобразити попередній або наступний фрейм. Цей параметр перетворює редактор анімації в світлову таблицю: він дозволяє відображати попередній або наступний фрейм напівпрозора над поточним фреймом. Ви маєте можливість копіювати попереднє зображення і створити новий фрейм з нього. Переміщення курсора ліворуч відображає попередній фрейм, а праворуч – наступний фрейм.

ЛАБОРАТОРНА РОБОТА №4

СТВОРЕННЯ ПОДІЙ І ДІЙ ВДОДАТКУ CLICKTEAM FUSION

МЕТА РОБОТИ:

- вивчення способів створення подій;
- вивчення способів створення варіантів дій об'єктів;
- застосування способів створення подій і дій до ігрових персонажів проекту.

ПОРЯДОК РОБОТИ.

1. Вивчити інструкцію до лабораторної роботи.
2. Відкрити програму редактора Clickteam Fusion.
3. Завантажити проект, створений в попередніх лабораторних роботах.
4. Ознайомитися з варіантами подій в ігровому додатку і способами їх створення в проекті.
5. Ознайомитися з варіантами дій в ігровому додатку і способами їх створення в проекті.
6. Створити події та дії об'єктів проекту відповідно до сценарію. Якщо буде потрібно, виконати коригування сценарію. Записати текст зміненого сценарію в звіті.
7. Сформулювати коментарі до подій і дій.
8. Перевірити працездатність проекту.
9. Зберегти створений проект.
10. Продемонструвати працездатність проекту викладачеві.
11. Закрити додаток Clickteam Fusion.
12. Оформити звіт. Захистити роботу.

ПОДІЇ, УМОВИ, ДІЇ І ФУНКЦІЇ

Кожен елемент, який ви додаєте до вашої програми у Clickteam Fusion 2.5, є об'єктом певного типу. Кожен об'єкт може реагувати на будь-яку кількість подій, мати багато умов, виконувати дії і функції, пов'язані з ними.

Події є основою додатків, створених в середовищі Clickteam Fusion 2.5. Кожна подія складається з однієї або кількох умов, дій та іноді функцій, які визначені в Event Editor (редакторі подій). Кожна подія очікує певної умови в додатку, а потім виконує відповідні дії.

Conditions (Умови) – перевіряються в додатках і за результатами перевірки запускаються actions (дії), пов'язані з ними. Колізії, рух і видимість – це приклади умов. Наприклад, два об'єкти стикаються, об'єкт починає відскакувати, або користувач клікає мишею, – це умови, що можуть викликати дію.

Дії запускаються в додатку, коли умова стає істинною. Як тільки певну умову виявлено, відбувається пов'язана дія (дії). Прикладами дій можуть бути рух, зміна параметрів видимості і перехід до нового кадру. Наприклад, об'єкт може перестати рухатися, стати невидимим або перейти до останнього кадру в додатку в результаті виконання певної умови.

Прикладом взаємодії між подіями, умовами і діями об'єкта може бути зіткнення космічного корабля з метеоритом, що призводить до руйнування космічного корабля. Подія являє собою комбінацію умови (зіткнення між космічним кораблем і метеоритом) і дії (знищення космічного корабля).

Функції дозволяють отримувати дані від одного об'єкта для використання іншим об'єктом або для використання в дії. Наприклад, якщо ви хочете відобразити кількість життів, які залишилися у гравця 3 в об'єкті Score, ви можете використовувати функцію Current number of lives (Поточна кількість життів) об'єкта Player для відображення в об'єкті Score.

УМОВИ

Умови тестуються в додатках і можуть викликати одну або кілька дій. Clickteam Fusion 2.5 визначає під час виконання, чи є умова істинною або хибною, а якщо вона істинна, запускає відповідну дію. Наприклад, умова може наказувати додатку перейти до наступного кадру, коли користувач натискає на певний об'єкт лівою кнопкою миші. Якщо користувач не натискає відповідний об'єкт, тоді умова є хибною, а поточний фрейм залишається на екрані.

Типові умови, з яких можуть складатися події:

- кількість життів гравця 1 = 0;
- початок відео;
- чи натиснув користувач «Enter key»?
- чи дорівнює значення таймера певному числу?
- чи були знищені всі солдати?
- зіткнення ракети з метеоритом.

Заперечення умов (Conditions that can be negated)

Можна створити умову, яке заперечує наявну умову, що є в меню умов об'єкта. Це робиться за допомогою оператора Negate. Наприклад, якщо ви хочете створити умову, яка перевіряє, чи не перебуває покажчик миші на конкретному об'єкті, можна заперечити умові Check for mouse pointer over an object (Перевірити, чи вказує миша на об'єкт). Заперечувальна умова буде істинною, якщо миша не знаходиться на вибраному об'єкті.

Нижче наведено список деяких умов, для яких може бути виконано заперечення:

- Sound – Звук;
- Array – Масив;
- CD-Audio;
- File – Файл;

- Hi-Score – Найбільший рахунок;
- Netscape – Браузер;
- ODBC – Відкрите підключення до бази даних;
- Picture – Малюнок;
- Window Control – Управління вікнами.

Умови об'єкта миші і клавіатури, які можна заперечувати:

- Repeat while key is pressed – Повторювати, поки натиснуто кнопку;
- Check for mouse pointer in a zone – Перевірка, чи знаходиться покажчик миші в зоні;
- Check for mouse pointer over an object – Перевірка покажчика миші над об'єктом;
- Repeat while mouse-key is pressed – Повторення, поки натиснута кнопка миші.

Умови об'єкта Player (гравець), які можна заперечувати:

- Повторення, поки натиснута кнопка джойстика;

Умови колізій або зіткнень (пов'язані з багатьма об'єктами), які можна заперечувати:

- Overlapping another object – Перекриття іншого об'єкта
- Overlapping a backdrop – Перекриття фону

Умови об'єктів List (список) і Combo Box (вибір варіантів), які можна заперечувати:

- Is enabled? – Чи є доступним?
- Have focus? – Чи має фокус?
- Is visible? – Чи є видимим?

Умови об'єкта Sub-Application (вкладений додаток), які можна заперечувати:

- Is application finished? – Чи завершено додаток?
- Frame has changed? – Чи відбулася зміна фрейма?

Умови об'єкта Button (кнопка), які можна заперечувати:

- Is visible? – Чи є видимою?

Умови Date & Time (Дата і час), які можна заперечувати

- Compare to chrono – Порівняти з хронометражем;
- Compare to count down – Порівняти із зворотним відліком;
- Is visible? – Чи є видимим?

Умови об'єкта Edit (редагування), які можна заперечувати:

- Has entry zone just been modified? – Чи змінено зону входу?
- Does entry zone have input focus? – Чи має вхідна зона фокус введення?
- Is entry zone enabled? – Чи включено зону входу?
- Is entry zone visible? – Чи є зона входу видимою?
- Is undo information available? – Чи доступна інформація про скасування?
- Is text a number? – Чи є текст числом?
- Is text selected? – Чи вибрано текст?

Умови об'єкта Search (пошук), що можна заперечувати:

- Search in progress? – Виконується пошук?

- Building index? – Індекс побудови?
- Whole word search? – Пошук усіх слів?
- Case search? – Пошук за регістром?
- Search in subdirectories? – Пошук в підкаталогах?

Оператора АБО (OR operators)

Оператори OR можуть спростити програму подій, дозволяючи групувати дії в одному рядку подій.

Оператор OR може бути вставлено в групу умов події, наприклад:

+ Умова1

+ Умова2

OR

+ Умова3

OR

+ Умова4

+ Умова5

Під час виконання ця група подій буде істинною, коли (Умова 1 і Умова 2) є істинною, АБО Умова 3 є істинною АБО (Умова 4 і Умова 5) істинні. Якщо будь-яка група умов або кілька цих груп умов істинні, дії виконуються. Оператор OR дозволяє групувати дії в одному рядку події замість того, щоб їх дублювати в кількох рядках.

Два види OR – фільтрувальне АБО (filtered OR) і логічне АБО (logical OR). Вони працюють по-різному, і необхідно розуміти їх перед застосуванням. Різниця полягає в тому, які об'єкти вибираються умовами.

The Filtered OR operator – Фільтруючий оператор АБО

Цей оператор використовується в більшості випадків.

Приклад: якщо

+ Миша над об'єктом 1

OR (filtered)

+ Миша знаходиться над об'єктом 2, то

- Знищити об'єкт 1

- Знищити об'єкт 2

Якщо миша знаходиться над об'єктом 1, об'єкт 1 знищується, але об'єкт 2 залишається на місці. Якщо миша знаходиться над об'єктом 2, об'єкт 2 знищується, але об'єкт 1 залишається на місці. Фільтруючий оператор OR досліджує всі умови події і маркує всі об'єкти як перевірені. Об'єкти, що містяться в хибному стані, також відзначені, але дія, пов'язана з ними, не відбуватиметься. З цієї причини дії у події з фільтруючим оператором OR впливатимуть тільки на об'єкти, вибрані в умовах, які є істинними.

The Logical OR operator- Логічний оператор АБО

Цей оператор не позначає об'єкти за помилковими умовами. Оскільки об'єкти

не позначені, то за замовчуванням дія, пов'язана з не зазначеними об'єктами, застосовується до всіх об'єктів. У нашому прикладі, якщо ми замінимо фільтруючий оператор OR логічним OR, отримаємо наступний результат:

якщо миша знаходиться над об'єктом 1, то обидва об'єкти знищуються, якщо об'єкт знаходиться над об'єктом 2, то обидва об'єкти знищуються. Уявіть, що миша знаходиться над об'єктом 1, обрано об'єкт 1. Викликається дія. Дія Destroy object 1 знаходить виділений об'єкт (той, що знаходиться під мишею) і видаляє його. Дія Destroy object 2 не знаходить зазначених об'єктів і знищуються всі входження об'єкта 2.

Використання будь-якого з операторів OR залежить від розробника гри і від програми, яку він створює.

ДІЇ (ACTIONS)

Дії відбуваються як результат виконання умов в додатках. Як тільки певну умову виявлено, відбувається пов'язана з нею дія (дії).

Кожен тип об'єкта має свій власний набір дій, який може відбуватися, коли виявляється істинна умова. Наприклад:

- живі об'єкти можуть бути отримані або втрачені;
- об'єкти підрахунку можуть бути збільшені або зменшені;
- об'єкти розкадровки можуть викликати перехід до наступного кадру, викликати перехід до попереднього кадру, перезапустити поточний фрейм і завершити додаток;
- активні об'єкти можуть бути переміщені, анімовані або знищені;
- об'єкти Netscape можуть запускати певні веб-сайти World Wide Web.

Можна отримати доступ до дій об'єкта в редакторі подій. У верхньому рядку редактора подій є рядок значків, які представляють доступні типи дій. Набори цих значків відрізняються в залежності від того, які об'єкти включені в додаток. Можна ввести свої дії, клікнувши правою кнопкою миші по квадрату сітки під значком і вибравши дію із спливаючого меню.

ВПОРЯДКУВАННЯ ДІЙ В РАМКАХ ПОДІЇ

Порядок додавання дій в подію визначає порядок виконання дій, коли стають справжніми умови цих подій. Порядок дій може бути дуже важливим. Наприклад, якщо подія має дві умови: одна для збільшення лічильника, а інша для вибору елемента зі списку, що відповідає значенню лічильника, порядок є дуже важливим. Якщо лічильник зараз дорівнює 1 і його значення збільшується на 1, подія буде вибирати другий елемент у списку. Однак, якщо дії виконуються в зворотному порядку, то вибирається перший елемент у списку.

Для зміни порядку дій у події після його введення можна натиснути правою кнопкою миші номер події і вибрати Edit the Actions (Змінити дії). Це відкриває редактор списку подій, який відображає тільки дії для вибраної події. Ви можете перетягувати дії в списку, щоб змінити порядок їх виконання.

ФУНКЦІЇ

Функції дозволяють отримувати дані з одного об'єкта для використання в іншому об'єкті. Наприклад, можна отримати дату, що відображається в об'єкті Date & Time, і відобразити її в об'єкті String.

Доступ до функцій можна отримати за допомогою кнопки Retrieve Data From An Object (Витягти дані з об'єкта) в редакторі Expression (виразів). Кожна функція завантажує певне значення в рівняння, що відображається в редакторі Expression. Редактор Expression відображається щоразу, коли потрібно ввести число або текстовий рядок.

Існує два типи функцій – General і Object.

General Functions- загальні функції

Загальні функції дістають дані з об'єктів Restricted objects (з обмеженим доступом), що доступні тільки в редакторі подій. Об'єкти з обмеженим доступом – Special (спеціальні), Sound (звукові), Storyboard Controls (фреймових елементів управління), Timer (таймер), Create New Objects (створення нових об'єктів), Mouse & Keyboard (миша і клавіатура) і Player (гравець). Наприклад, функції Player дозволяють отримати поточне значення рахунку і поточну кількість життів, які залишилися у гравця. Інші приклади загальних функцій включають в себе отримання поточного номера фрейма, загальну кількість об'єктів в кадрі, поточну горизонтальну (X) позицію миші і поточне значення таймера.

Object Functions- функції об'єктів

Функції Object дістають дані з об'єктів, відмінних від об'єктів з обмеженим доступом. Доступні опції залежать від типу об'єкта. Наприклад, можна отримати швидкість, положення і змінні значення активного об'єкта, номер поточної доріжки об'єкта CD-Audio і поточну URL-адресу об'єкта Netscape.

THE EVENT EDITOR – РЕДАКТОР ПОДІЙ

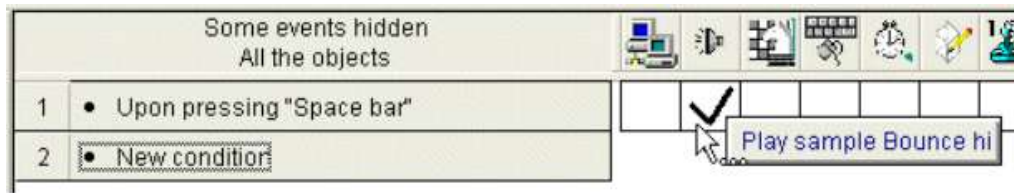
Редактор подій – це місце, де вводяться події. Події запускають всі дії і визначають логіку гри або програми. Події складаються з однієї або декількох умов і однієї або декількох дій.

Редактор подій відображає умови і дії у вигляді таблиці. Умови перераховані вертикально в лівій колонці, а дії – горизонтально в рядку.

Для введення нової умови слід натиснути кнопку New Condition (Нова умова), що відображається в лівій колонці, потім натиснути правою кнопкою миші на об'єкті, який необхідно протестувати в діалоговому вікні, і вибрати умову із спливаючого меню.



Щоб ввести нову дію, знайдіть умову, яка викличе дію, і знайдіть по горизонталі об'єкт, який ви хочете змінити. Клікніть правою кнопкою миші в квадраті, розташованому під об'єктом в рядку з умовою – з'явиться спливаюче меню, в якому потрібно вибрати дію.



Drag & Drop – перетягнути і кинути

Редактор подій керується мишею з використанням технології drag & drop. Можна перетягувати такі елементи:

- умови між різними подіями;
- дії (прапорці) від одного об'єкта до іншого;
- події для зміни порядку в списку;
- об'єкти на верхньому рядку, щоб налаштувати необхідний порядок їх розташування;
- можна перебудувати редактор подій так, як зручно, перетягуючи рядки і змінюючи розмір.

Blocks – блоки

Можна легко створювати блоки подій і виконувати операції «вирізати/вставити/видалити»: редактор подій реагує як будь-який редактор на операції вирізання та вставки. Звичайні поєднання клавіш застосовуються і до редактора подій. Можна вибрати параметри вирізання/вставки і в головному меню.

Undo/Redo – скасувати/повторити

Редактор подій містить багаторівневу систему скасування/повтору, яку можна використовувати при виникненні помилки.

Re-ordering – зміна порядку

Можна змінити порядок дій в рядку події. Для цього двічі клікніть в середині рядка або клікніть правою кнопкою миші по номеру події і виберіть Edit the actions (Редагувати дії), відкриється редактор дій.

ПАНЕЛЬ ІНСТРУМЕНТІВ РЕДАКТОРА ПОДІЙ

Панель інструментів редактора подій містить такі індикатори:



Налаштування масштабування: використовуйте повзунок, щоб змінити розмір подій на дисплеї, від 0 до 200.



Перейти до рядка. Натисніть на цей значок, щоб перейти безпосередньо до певного номера рядка. Дуже зручно для великих програм.



Перейти до колонки. Натисніть на цей значок, щоб курсор перемістився в заданий стовпець. Дуже зручно, якщо у вас багато об'єктів.



Показати всі події



Відображає тільки події, що пов'язані з активними об'єктами.



Відображає тільки події, що пов'язані з мультимедійними об'єктами (CD, відео і т. д.).



Відображає тільки події, що пов'язані з текстовими об'єктами.



Відображає тільки події, що пов'язані з ігровими об'єктами.



Відображає тільки події, що пов'язані з елементами управління Windows (наприклад, кнопкою).



Відображає тільки події, що пов'язані з розширеними об'єктами.



Показує всі приховані події



Приховує системні об'єкти з горизонтальною панелі об'єктів.



Приховує активний об'єкт на панелі об'єктів.



Приховує мультимедійні об'єкти на панелі інструментів об'єктів.



Приховує об'єкти, пов'язані з текстом, на панелі інструментів об'єктів.



Приховує ігрові об'єкти на панелі інструментів об'єктів.



Приховує об'єкти управління Windows на панелі інструментів об'єктів.



Приховує розширені об'єкти на панелі інструментів об'єктів.

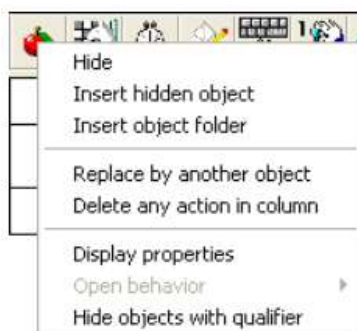


Показує всі приховані об'єкти (видалити фільтри подій).

THE EVENT EDITOR MENUS – МЕНЮ РЕДАКТОРА ПОДІЙ

Редактор подій містить багато спливаючих меню: вони забезпечують дуже швидкий доступ до ряду функцій. Ці меню перераховані нижче.

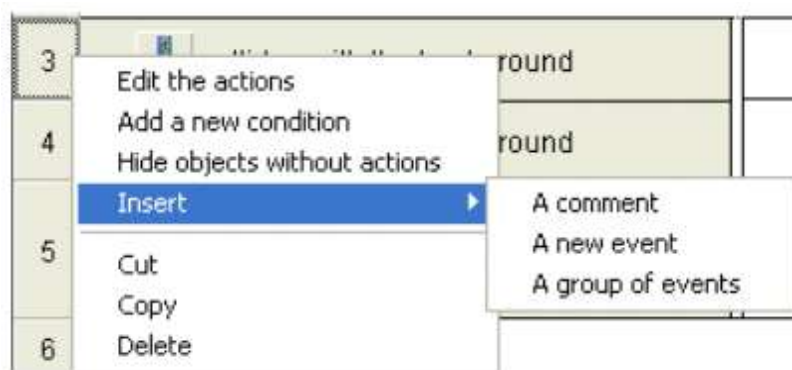
THE OBJECTS BAR MENU – МЕНЮ ОБ'ЄКТІВ



Це меню відкривається при натисканні правою кнопкою миші на одному з об'єктів в панелі об'єктів.

Hide – Приховати	Приховує об'єкт з дисплея. Використовуйте цей параметр, якщо у вас багато об'єктів: об'єкти зберігаються в іншій прихованій папці. Спеціальний значок з'являється праворуч від панелі об'єктів, щоб повернути об'єкт, або ви можете використовувати наступний параметр.
Insert hidden object – Insert object folder – Вставити папку об'єктів	Відкриває діалог для вибору та вставки прихованих об'єктів. Вставити приховані об'єкти з папки об'єктів – це зручно при класифікації об'єктів на панелі. При виборі цього варіанту створюються дві дужки в місці натискання. Тепер можна перетягувати об'єкти між цими дужками. Подвійне клікання на одній з дужок закриє папку: всі об'єкти буде збережено в папці. Щоб знову відкрити папку, слід двічі клікнути по ній.
Replace by another object – Замінити іншим об'єктом	Відкриває нове діалогове вікно для вибору іншого об'єкта. Відображаються тільки об'єкти того ж типу, що й обраний об'єкт. Виберіть об'єкт і натисніть «ОК»: редактор подій замінить всі входження першого об'єкта іншим об'єктом як в діях, так і в умовах.
Delete any action in column – Видалити дії з колонки	Опція видалить всі дії в стовпці. Наприклад, якщо потрібно видалити всі зразки і звуки, виберіть цей параметр для об'єкта Speaker.
Display properties – відобразити властивості	Відображає властивості об'єкта на панелі інструментів властивостей. Якщо панель інструментів властивостей було закрито, вона відкривається.
Open behavior – відкрити поведінку	Якщо об'єкт містить поведінку, це підменю дозволяє відкрити редактор подій поведінки.
Hide objects with qualifiers – Приховати об'єкти з кваліфікаторами	Ця установка доступна лише для кваліфікаторів. Всі об'єкти, що мають цей визначник, будуть приховані, але визначник залишиться видимим.

THE EVENT MENU – МЕНЮ ПОДІЙ



Це меню відкривається при натисканні на один з номерів ліворуч від дисплея.

Edit the actions – редагувати дії	Ця опція відкриває редактор дій для зміни порядку дій.
Add a new condition – Додати нову умову	Відкриває діалогове вікно об'єктів для вибору нової умови. Нова умова додається у кінці списку умов.

Hide objects without action – Приховати об'єкти без дій	Приховує всі об'єкти, які не містять ніяких дій. Попередження: це може видалити багато об'єктів з дисплея, які доведеться повторно вводити вручну.
Insert/A new comment – Вставити/коментар	Вставляє нові коментарі в список подій. Відкриває діалог параметрів коментарів. Застосовується для великих програм.
Insert/A new event – Вставити/Нова подія	Вставка нової події в поточну позицію.
Insert/A group of events – Вставити/Групу подій	Нова група подій вставляється в поточну позицію.
Cut – Вирізати	Копіює рядок події в буфер обміну і видаляє його.
Copy – Копіювати	Копіює рядок події в буфер обміну.
Delete – Видалити	Видаляє поточний рядок.

THE CONDITION MENU – МЕНЮ УМОВ

Це меню відкривається при натисканні правою кнопкою миші на умову в списку подій.

Edit – Редагування	Змінює поточний дію. Це може відкрити інші діалогові вікна і редактори.
Replace – Замінити	Замінює по заданій умові. Ця опція відкриває діалогове вікно New Condition для вибору умови.
Insert – Вставити	Вставляє нову умову. Ця опція відкриває діалогове вікно нової умови.
Negate – Згорнути	Згортає поточну умову.
OR operator (filtered) Фільтруючий оператор АБО	Вставляє фільтруючий оператор OR в список умов подій.
OR operator (logical) Логічний оператор АБО	Вставляє логічний оператор OR в список умов подій.
Cut – Вирізати	Копіює поточну умову в буфер обміну і видаляє її з екрану.
Copy – Копіювати	Копіює поточну умову в буфер обміну, залишаючи на екрані.
Paste – Вставити	Вставляє умову, що міститься в буфері обміну.
Delete – Видалити	Видаляє поточний стан.

THE ACTION MENUS – МЕНЮ ДІЙ

При натисканні правою кнопкою миші по комірці в сітці з'являється спливаюче меню. Це меню залежить від об'єкта, на який ви натиснули.

THE EVENT-LIST EDITOR – РЕДАКТОР СПИСКУ ПОДІЙ

Редактор списку подій – це інший спосіб редагування програми подій. Відображення подій в цьому редакторі виконується у вигляді списку, причому умови і дії перераховані одна за одною. Можна використовувати як цей редактор, так і редактор подій.

Для введення нової події слід натиснути кнопку New condition (Створити умову): відкриється діалогове вікно, яке представляє всі об'єкти. Клікнути правою кнопкою миші по об'єкту, який слід перевірити, і вибрати умову в списку.

- New Condition 

Після створення умови можна додати додаткові умови до події, клікнувши правою кнопкою миші на умові і вибравши Insert (Вставити). Після завершення редагування умов натиснути на рядок Insert your action here (Вставте свою дію тут):

- Upon pressing "Space bar"  : Insert your action here

Відкриється діалогове вікно, в якому будуть представлені всі об'єкти програми. Клікнути правою кнопкою миші по об'єкту, який потрібно змінити, і вибрати дію в списку. Нова дія буде вставлено в подію. Можна визначити кілька дій: клікніть правою кнопкою миші по дії і виберіть опцію Вставити у спливаючому меню.

Drag & Drop – перетягнути і кинути

Як і більшість редакторів, редактор списку подій підтримує перетягування. Ви можете перетягувати дії, умови або рядки подій.

Cut & Paste – Вирізати і вставити

Редактор списку подій дозволяє створювати блоки подій, умов і дій. Можна використовувати звичайні команди вирізання/вставки, а подія, умова або дія під курсором будуть скопійовані або вставлені.

Undo/Redo – Скасувати/Повторити

Редактор списку подій, як і редактор подій, зберігає кілька функцій Undo / Redo.

Printing the events – Друк подій

Однією з головних переваг редактора списку подій є його можливість роздруковувати список подій. Вибрати Print (Друк), Print preview (Попередній перегляд) або Print Setup (Параметри друку) у меню File (Файл) в рядку головного меню.

THE EXPRESSION EDITOR – РЕДАКТОР ВИРАЗІВ

Редактор виразів з'являється, коли потрібно ввести значення для дії або умови. Тому його вікно відкривається з редактора подій.

Редактор виразів працює як калькулятор. Поточний вираз відображається в зоні редагування, для введення нового значення слід клікати по кнопках.



Цифрові кнопки використовуються для введення цифр, функціональні кнопки – для введення функцій, таких як Sin і Cos. Деякі функції запитують параметр, параметри виділяються знаками <> і автоматично вставляються при виборі функції. Замініть вміст між кутовими дужками відповідним параметром. Щоб вибрати такі кутові дужки, натисніть кнопку Next (Далі).

Деякі дії або умови запитують текстові вирази, укладені в лапки, наприклад, «Цей рядок», і функції, що закінчуються на \$.

Retrieve data from object - Отримати дані від об'єкта. Ця кнопка відкриває діалогове вікно, в якому представлені всі об'єкти програми. Клікніть правою кнопкою миші на одному з об'єктів, відкриється спливаюче меню, в якому представлені можливі вирази об'єкта, в ньому потрібно вибрати, яке значення необхідно дістати з об'єкта. Натисніть «OK», і в зоні редагування з'явиться нова функція.

Check current expression - Перевірити поточний вираз. Натисніть цю кнопку, щоб перевірити правильність виразу. Якщо з'являється «Valid expression», вираз написано правильно. Якщо з'являється «Syntax error» (Синтаксична помилка), то у виразі допущена помилка, яку слід виправити.

ЛАБОРАТОРНА РОБОТА №5

ЗВУКОВЕ ОФОРМЛЕННЯ І ЗАВЕРШЕННЯ ГРИ В ДОДАТКУ SLICKTEAM FUSION

МЕТА РОБОТИ:

- знайомство з можливостями перевірки стану об'єкта;
- вивчення варіантів знищення об'єкта в ході гри;
- вивчення можливості ведення рахунку в грі;
- знайомство з можливостями завершення гри.

ПОРЯДОК РОБОТИ.

1. Вивчити інструкцію до лабораторної роботи.
2. Відкрити програму редактора Clickteam Fusion.
3. Завантажити проект, створений у попередніх лабораторних роботах.
4. Ознайомитися з можливостями використання координат об'єкту при його переміщенні.
5. Виконати перевірку стану об'єкта в ході гри.
6. Створити події і ефекти знищення об'єктів відповідно до сценарію гри.
7. Створити звуковий супровід і звукові ефекти.
8. Перевірити працездатність проекту.
9. Зберегти створений проект.
10. Продемонструвати працездатність проекту викладачеві.
11. Оформити звіт. Захистити роботу.

ВПОРЯДКУВАННЯ РОЗТАШУВАННЯ ОБ'ЄКТІВ НА ФРЕЙМІ

Інструменти упорядкування об'єктів на фреймі доступні через меню Arrange:

- **Order** – Порядок: дозволяє змінити порядок об'єктів.

Примітка 1: деякі типи об'єктів не можуть перебувати перед або позаду інших типів в даному шарі. Фонові об'єкти не можуть перебувати перед активними об'єктами, а активні об'єкти не можуть перебувати перед об'єктами управління Windows. Щоб відображати фонові об'єкти перед активними об'єктами, необхідно створити новий шар і перемістити ці фонові об'єкти на новий шар.

Примітка 2: це не стосується об'єктів управління Windows, тобто об'єктів, що відображаються в їх власному вікні, оскільки цей вид об'єкта завжди відображається зверху усіх шарів і зверху будь-якого фону або активного об'єкта.

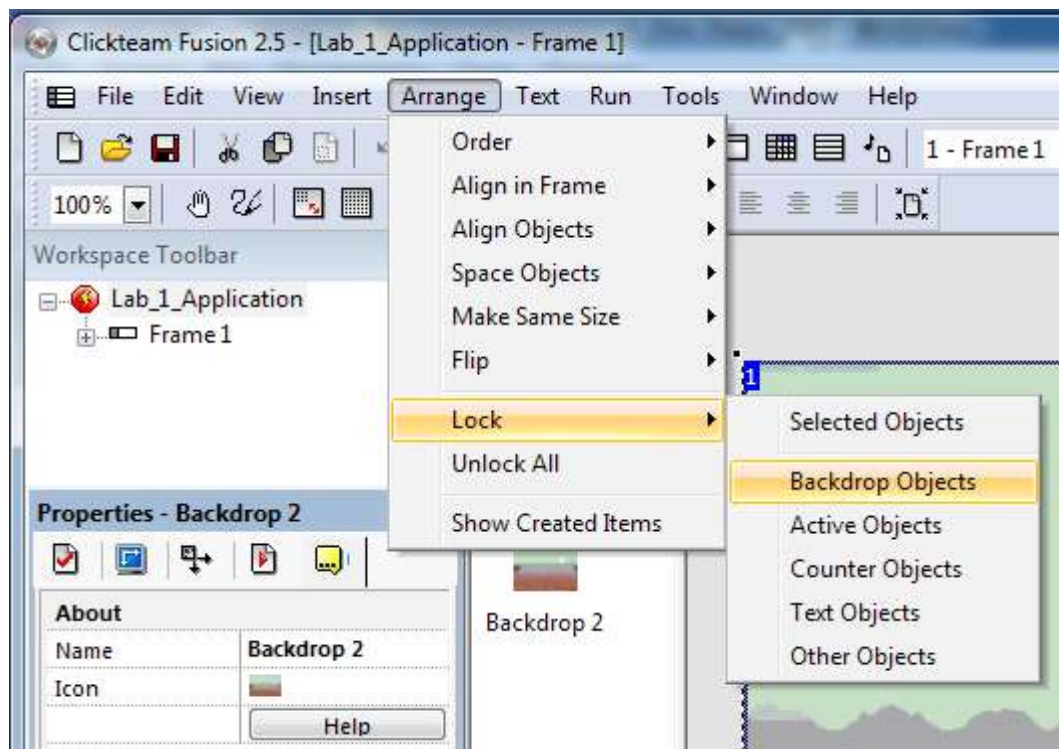
- **To Front**: розміщує об'єкт зверху інших в тому ж шарі.
- **To Back**: розміщує об'єкт за іншими на тому ж шарі.

- **Forward one:** поміщає об'єкт зверху наступного об'єкта в тому ж шарі.
- **Backward one:** поміщає об'єкт позаду попереднього об'єкта в тому ж шарі.
- **By X / Y:** сортує вибрані об'єкти, зменшуючи або збільшуючи координати.
- **Align In Frame:** дозволяє вирівняти обрані об'єкти у фреймі.
- **Align Objects:** дозволяє вирівняти обрані об'єкти по відношенню до виділеного об'єкту.
- **Space Objects:** дозволяє рівномірно розподілити вибрані об'єкти по горизонталі або по вертикалі.
- **Make Same Size:** дозволяє зробити розмір вибраних об'єктів рівним розміру виділеного об'єкта по горизонталі або по вертикалі.
- **Flip:** дозволяє перевернути вибрані об'єкти по горизонталі або по вертикалі.
- **Lock:** дозволяє заблокувати всі об'єкти заданих типів. Об'єкти для блокування можна вибрати.
- **Unlock All:** розблокує всі заблоковані об'єкти. Примітка. Об'єкти також можна розблокувати по одному, виділивши об'єкт і натиснувши SHIFT + CTRL.

БЛОКУВАННЯ ФОНУ

Заблокований фон не можна перетягнути, змінити розмір і т.д. Для блокування фону у вікні редактора фрейма вибрати об'єкт фону Backdrop, в меню вибрати команди

Arrange -> Lock -> Backdrop Objects:

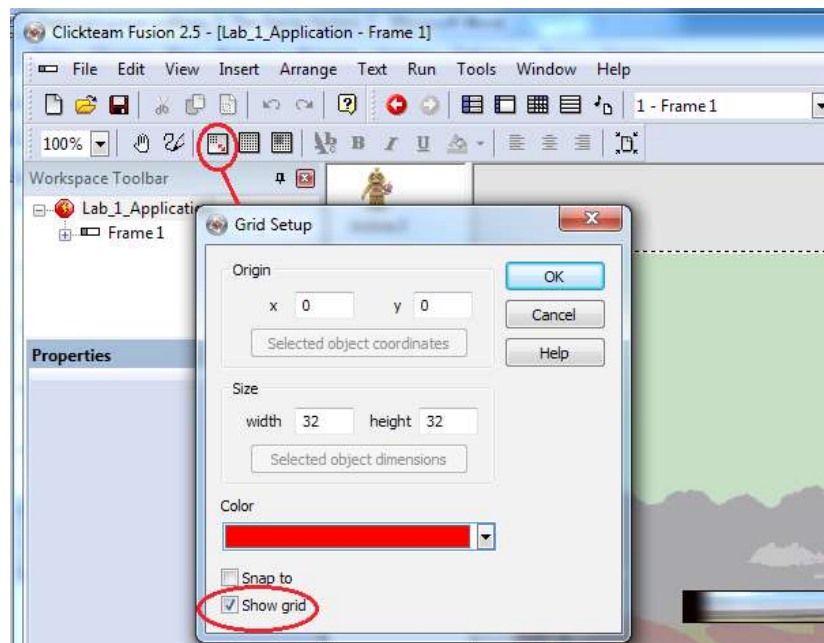


Аналогічним чином можна заблокувати й інші об'єкти. Заблоковані об'єкти не будуть переміщатися на екрані.

Для зняття блокування виконати команди меню Arrange -> Unlock All.

ВІДОБРАЖЕННЯ КООРДИНАТНОЇ СІТКИ

У вікні редактора фрейма вибрати об'єкт фону Backdrop. На панелі інструментів відкрити вікно Grid Setup, в ньому встановити прапорець Show Grid:



Origin – координати початкової точки сітки – лівого верхнього кута;

Size – встановлення кроку сітки в пікселях – значення в полях width (ширина) height (висота);

Show Grid встановлення прапорця показує сітку;

Snap to grid – встановлення прив'язки до сітки: при створенні і переміщенні об'єкти розташовуються з прив'язкою до вузлів сітки.

Після встановлення параметрів сітки натиснути OK.

Синім овалом виділені координати лівого верхнього кута активного об'єкта, червоним овалом – поточні координати курсора, між ними – розмір об'єкта:



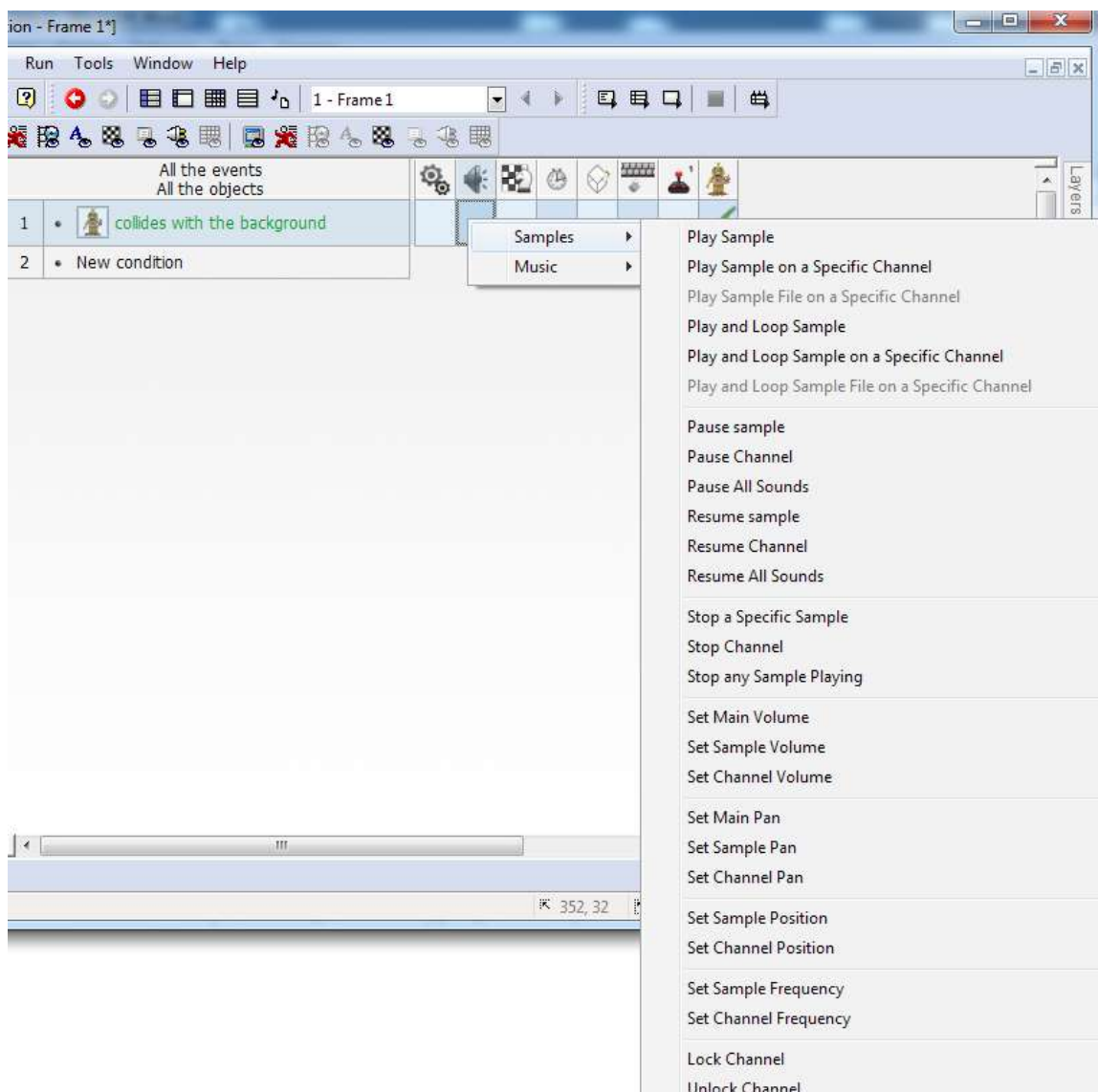
THE SOUND OBJECT – ОБ'ЄКТ ЗВУКУ

Об'єкт Sound відтворює різні звукові файли в додатках Clickteam Fusion 2.5. Формати залежать від версії Clickteam Fusion 2.5 та від встановлених звукових фільтрів. Всі версії Clickteam Fusion 2.5 підтримують як мінімум файли .WAV, .MOD і .MID. Фрагменти і музика можуть бути запущені, зупинені і зациклені з використанням цього об'єкта. Об'єкт Sound може виявити ситуацію, коли звук або музика не відтворюються, якщо потрібно викликати дію в кінці певного звукового ефекту.

Звуки, що відтворюються в Play Sample і Play Music об'єкта Sound, вбудовуються в файл програми. Звуки, що відтворюються в файлах Play Sample File і Play Music File, не вбудовуються в додаток і мають бути скопійовані при копіюванні автономного додатка.

ДІЇ ЗІ ЗВУКОМ. МЕНЮ SAMPLES

Меню дій зі звуком за зразком випадає при натисканні на кнопку Samples.



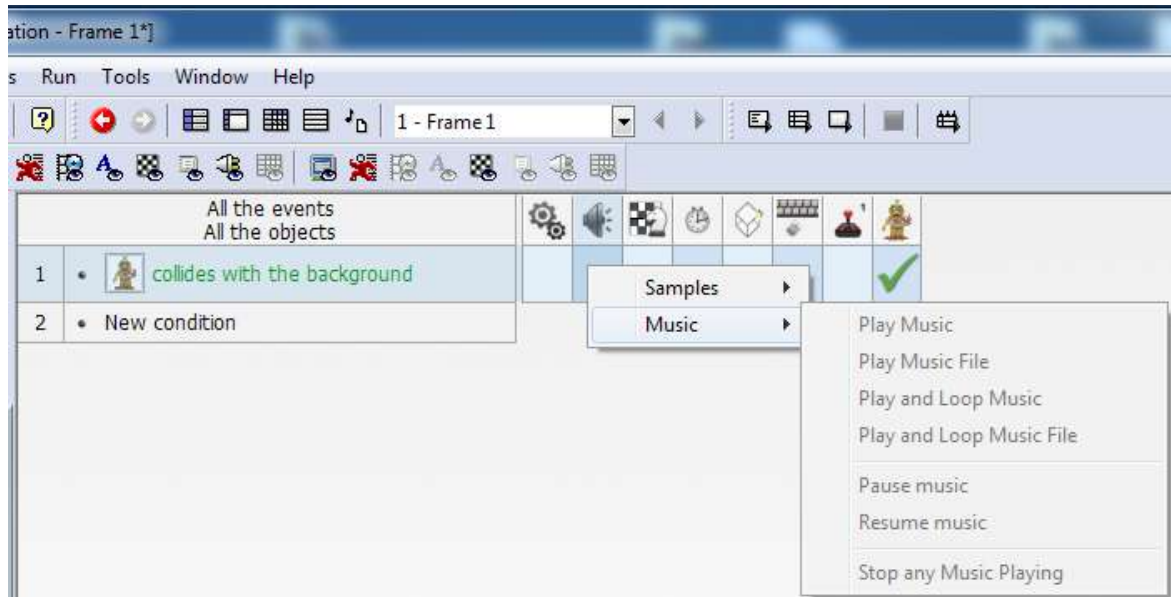
- **Play Sample** – відтворюється певний фрагмент звуку;
- **Play sample on a specific channel** – відтворення звуку на одному з 32 різних доступних звукових каналів;

- **Play sample file on a specific channel** – відтворення фрагмента зовнішнього файлу на одному з 32 різних доступних звукових каналів;
- **Play And Loop Sample** – відтворення звукового фрагмента в циклі. Якщо число циклів дорівнює 0, фрагмент програватиметься постійно;
- **Play and loop sample on a specific channel** – відтворення звуку на одному з 32 різних доступних звукових каналів і повторення зразка вказану кількість разів. Якщо число циклів дорівнює 0, фрагмент програватиметься постійно;
- **Play and loop sample file on a specific channel** – відтворення зовнішнього файлу на одному з 32 різних доступних звукових каналів і повторення фрагмента задану кількість разів;
- **Pause sample** – зупинка відтворення звуку;
- **Pause channel**- зупинка звуку, відтворюваного тільки на певному каналі (1-32), всі інші канали будуть продовжувати відтворюватися;
- **Pause all sounds** – зупинка всіх відтворюваних звуків;
- **Resume sample** – відновлення зупиненого фрагмента;
- **Resume channel**- відновлення відтворення певного каналу (1-32), що не заважає іншим каналам;
- **Resume all sounds** – відновлення всіх призупинених фрагментів;
- **Stop a specific sample** – зупиняє конкретний фрагмент, відтворюваний в даний момент;
- **Stop Channel** – зупиняє звук, відтворюваний на певному каналі (1-32), що не заважає іншим каналам;
- **Stop any sample playing** – зупинка відтворення будь-якого звуку;
- **Set main volume** – встановлення основної гучності – зміна рівня гучності; гучність позначається числом від 0 (звук відсутній) до 100 (гучний);
- **Set sample volume** – зміна рівня гучності конкретного фрагмента; гучність позначається числом від 0 (звук відсутній) до 100 (гучний);
- **Set channel volume** – зміна рівня гучності на певному каналі (1-32); гучність позначається числом від 0 (звук відсутній) до 100 (гучний);
- **Set main pan** – встановлює основну панорамну гучність; панорама позначається цифрою від -100 (весь звук зліва, без звуку праворуч) до 100 (весь звук праворуч, без звуку зліва), 0 знаходиться посередині;
- **Set sample pan** – встановлює загальну гучність для певного фрагмента; значення від -100 (весь звук зліва, без звуку праворуч) до 100 (весь звук праворуч, без звуку зліва), 0 знаходиться посередині;
- **Set channel pan** – встановлює панорамну гучність певного каналу (1-32);
- **Set sample position** – для фрагмента із заданим іменем встановлюється відступ від початку програвання в мілісекундах;
- **Set channel position** – описані вище дії застосовуються для певного каналу (1-32);
- **Lock Channel** – блокує канал; заблоковані канали ігноруються дією Play Sample, коли він шукає доступний канал для відтворення звуку;
- **Unlock Channel** – розблоковує канал;
- **Set sample frequency** – змінює частоту конкретного зразка; 0 – використовує вихідну частоту звуку;

- **Set channel frequency** – змінює частоту конкретного каналу (1-32). 0 – використовує вихідну частоту звуків, відтворених на цьому каналі.

МЕНЮ MUSIC

Меню для дій з музикою випадає при натисканні на кнопку Music



- **Play Music** – відтворення певного музичного MIDI-файлу;
- **Play Music File** – відтворення зовнішнього музичного MIDI-файлу;
- **Play And Loop Music** – відтворення музичного MIDI-файлу стільки разів, скільки вказано; якщо кількість циклів дорівнює 0, музика звучатиме безперервно;
- **Play And Loop Music File** – відтворення музичного MIDI-файлу вказану кількість разів;
- **Pause music** – тимчасова зупинка музики;
- **Resume music** – відновлення припиненої музики;
- **Stop any music playing** – зупинка відтворення будь-якої музики.

УМОВИ ДЛЯ ОБ'ЄКТА SOUND

Умови фрагментів перевіряють відтворення або призупинення звукових фрагментів.

Умови фрагментів:

- **Is a sample not playing?** – перевіряється, чи не відтворюється будь-який звук;
- **Is a specific sample not playing?** – перевіряється, чи не відтворюється конкретний звуковий фрагмент;
- **Is a specific channel not playing?** – перевіряється, чи не відтворюється конкретний звуковий канал;
- **Is a specific sample paused?** – перевіряється, чи не призупинено конкретний фрагмент;
- **Is a specific channel paused?** – перевіряється, чи не призупинено певний фрагмент.

ВИРАЗИ ДЛЯ ОБ'ЄКТА SOUND

Main Volume – повертає основний рівень гучності додатка, від 0 до 100;

Sample Volume – повертає рівень гучності певного фрагмента, від 0 до 100; Parameter = ім'я фрагмента;

Channel Volume – повертає рівень гучності певного каналу, між 0 і 100; Параметр = індекс каналу (1-32);

Main Pan – повертає основну панораму додатка від -100 до +100.

Sample Pan – повертає панораму певного фрагмента, між -100 і +100; параметр = ім'я фрагмента;

Channel Pan – повертає панорамування певного каналу, між -100 і +100; параметр = індекс каналу (1-32);

Sample Position – повертає позицію конкретного фрагмента в мілісекундах; параметр = ім'я фрагмента;

Channel Position – повертає позицію певного каналу в мс; параметр = індекс каналу (1-32);

Sample Duration – повертає тривалість певного фрагмента в мілісекундах; параметр = ім'я фрагмента;

Channel Duration – повертає тривалість певного каналу в мілісекундах; параметр = індекс каналу (1-32);

Sample Frequency – повертає частоту звуку певного фрагмента в мілісекундах; параметр = ім'я фрагмента. Якщо фрагмент не чути, то повертає 0;

Channel Frequency – повертає частоту конкретного каналу. Параметр = індекс каналу (1-32); зауваження: якщо фрагмент відтворюється на цьому каналі, ця функція повертає частоту цього фрагмента, в іншому випадку повертає стандартну частоту даного каналу (0 = за замовчуванням).

Навчальне видання

НІКІТІН Сергій Олександрович
НІКІТІНА Людмила Олексіївна

ОСНОВИ КОМП'ЮТЕРНИХ ІГОР ТА ІГРОВИХ ПРОГРАМ ДОВІДНИК МОДУЛЯ

Роботу до видання рекомендував *О.А. Серков*

Редактор *Л.П. Гобельовська*
Комп'ютерна верстка *Ю.Ф. Власова*
Коректор *Г.В. Сільченко*

Підписано до друку 07.02.2018 р.
Формат 60x84 1/16. Папір офсетний.
Цифровий друк.
Гарнітура DINPro. Ум. друк. арк. 17,21.
Наклад 50 прим. Зам. № ГЛ00-000476.

Видавець і виготовлювач:
ТОВ «Друкарня Мадрид»
61024, м. Харків, вул. Максиміліанівська, 11
Тел.: (057) 756-53-25
www.madrid.in.ua info@madrid.in.ua

Свідоцтво суб'єкта видавничої справи:
ДК № 4399 від 27.08.2012 р.

