



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРНЕТИКИ ТА СИСТЕМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ**

ШЕРСТЮК В.Г.

**ЕЛЕКТРОННИЙ НАВЧАЛЬНИЙ
ПОСІБНИК**

«Основи розробки комп'ютерних ігор»

*Для підготовки студентів
на першому (бакалаврському) рівні вищої освіти,
галузі знань 12 «Інформаційні технології»,
спеціальності 121 «Інженерія програмного забезпечення»,
освітньо-професійної програми «Програмна інженерія»*

ХЕРСОН-2018

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРНЕТИКИ ТА СИСТЕМОЇ ІНЖЕНЕРІЇ
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ**

ЕЛЕКТРОННИЙ НАВЧАЛЬНИЙ ПОСІБНИК

«Основи розробки комп'ютерних ігор»

*Для підготовки студентів
на першому (бакалаврському) рівні вищої освіти,
галузі знань 12 «Інформаційні технології»,
спеціальності 121 «Інженерія програмного забезпечення»,
освітньо-професійної програми «Програмна інженерія»*

**GAMEHUB: «СПІВРОБІТНИЦТВО МІЖ УНІВЕРСИТЕТАМИ ТА
ПІДПРИЄМСТВАМИ В СФЕРІ ІГРОВОЇ ІНДУСТРІЇ В УКРАЇНІ»**

**GAMEHUB: «UNIVERSITY-ENTERPRISES COOPERATION IN GAME
INDUSTRY IN UKRAINE»
561728-EPP-1-2015-1- ES-EPPKA2-CBHE-JP**

The handbooks were performed with support of the Erasmus+ Programme of the European Union (561728-EPP-1-2015-1- ES-EPPKA2-CBHE-JP). 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.

ХЕРСОН-2018



**УДК 004.4
О–75**

Основи розробки комп'ютерних ігор: електронний навчальний О–75 посібник для підготовки студентів на першому (бакалаврському) рівні вищої освіти, галузі знань 12 «Інформаційні технології», спеціальності 121 «Інженерія програмного забезпечення» / Укладач: В.Г. Шерстюк. – Херсон: видавництво ФОП Вишемирський В.С., 2018. – 210 с.

ISBN 978-617-7573-92-9 (електронне видання)

Укладач:

Шерстюк В.Г., д.т.н., професор, завідувач кафедри Програмних засобів і технологій;

Рецензенти:

Шарко О.В., д.т.н, професор кафедри транспортних технологій Херсонської державної морської академії.

Львов М.С., д.ф-м.н, професор, завідувач кафедри інформатики, програмної інженерії та економічної кібернетики Херсонського державного університету

Розглянуто на засіданні кафедри Програмних засобів і технологій,
Протокол №11 від 23 травня 2017 року.

Рекомендовано до друку Вченою радою Херсонського національного технічного університету,
Протокол №9 від 2 червня 2017 року.



Цей матеріал ліцензовано на умовах Ліцензії Creative Commons CC BY-NC-SA

Із Зазначенням Авторства — Некомерційна — Поширення На Тих Самих Умовах 4.0 Міжнародна

УДК 004.4

ISBN 978-617-7573-92-9

© Шерстюк В.Г., 2018
© ФОП Вишемирський В. С., 2018

ЗАГАЛЬНИЙ ВСТУП

Електронний навчальний посібник складений відповідно до програми навчальної дисципліни «Основи розробки комп'ютерних ігор», яка входить до дисциплін підготовки студентів на першому (бакалаврському) рівні вищої освіти, галузі знань 12 «Інформаційні технології», спеціальності 121 «Інженерія програмного забезпечення», освітньо-професійної програми «Програмна інженерія» за спеціалізацією «Розробка комп'ютерних ігор», яка впроваджена в рамках виконання міжнародного проекту Еразмус+ 561728-ERP-1-2015-1- ES-ERPKA2-SVHE-JP «GameHub: Співробітництво між університетами та підприємствами в сфері ігрової індустрії в Україні».

Основною метою навчального посібника є узагальнення й систематизація теоретичних надбань у галузі розробки комп'ютерних ігор із застосуванням сучасних візуальних середовищ.

Посібник містить довідник дисципліни, лекційний матеріал, методичні вказівки до виконання кожної теми, питання для самоконтролю, задачі для самостійного розв'язку, критерії оцінювання практичних завдань та список рекомендованої літератури.

Запропонована послідовність тем та їх поєднання у три розділи створюють умови для логічного засвоєння змісту дисципліни. Питання, що розкривають сутність відповідних тем, забезпечують викладення методологічних основ дисципліни «Основи розробки комп'ютерних ігор», у разі успішного засвоєння яких студенти будуть вміти: застосовувати сучасні методи проектування, розробки та виробництва ігрових додатків; обирати інструменти розробки у відповідності до поставлених вимог; визначати ігрову концепцію; створювати сюжет, визначати та описувати персонажів, рівні, правила гри; моделювати сценарій гри та виконувати його розкадровку; визначати шляхи проходження гри; здійснювати дизайн гри, ігрових персонажів та об'єктів, враховуючи їх ігрові ролі; виявляти вплив прийнятих дизайнерських рішень та їх динаміки на психоемоційний стан гравця; визначати і вибудовувати зв'язки між ігровими персонажами та подіями в сценарії гри, забезпечуючи динаміку ігрового середовища.

У разі успішного вивчення матеріалів дисципліни «Основи розробки комп'ютерних ігор» студенти розвинуть такі особисті якості: креативний спосіб мислення при генерації ідей стосовно створення концепції та проекту комп'ютерної гри; самостійно вирішувати питання, що відносяться до створення комп'ютерних ігор; впевнено обґрунтовувати актуальність та презентувати ігрові концепції та проекти; систематизувати професійні знання; аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси з урахуванням сучасних досягнень науки і техніки.

ЗМІСТ

Довідник дисципліни	5
1. Вступ	5
2. Опис дисципліни та її модулів.....	6
3. Перелік компетентностей та результати навчання.....	7
4. Міждисциплінарні зв'язки.....	9
5. Форми навчання.....	9
6. Порядок проведення атестації з дисципліни.....	10
7. Зворотній зв'язок.....	144
8. Викладацький склад та допоміжні джерела.....	144
9. Модуль М4.1 Проектування комп'ютерних ігор.....	166
10. Модуль М4.2 Візуальна розробка комп'ютерних ігор.....	29
11. Модуль М4.3. Виконання групового фінального проекту.....	41
Лекція 1	47
Лекція 2	66
Лекція 3	82
Лекція 4	103
Лабораторна робота 1	122
Лабораторна робота 2	151
Лабораторна робота 3	165
Питання для самоконтролю	195
Завдання для виконання групового фінального проекту	200
Критерії оцінювання	205

ДОВІДНИК ДИСЦИПЛІНИ

1. ВСТУП

Предметом навчальної дисципліни є основи проектування ігрових програм, сучасні методи та технології розробки ігрових додатків за допомогою візуального інструментарію. Навчання студента зорієнтовано на командну розробку крос-платформних ігрових додатків.

Мета дисципліни

Метою викладання навчальної дисципліни «Основи розробки комп'ютерних ігор» є формування у студентів системи компетенцій щодо практичного використання технології розробки комп'ютерних ігор із застосуванням сучасних візуальних середовищ.

Основні завдання

Основними завданнями дисципліни «Основи розробки комп'ютерних ігор» є:

- вивчення основ проектування, підтримки та просування комп'ютерних ігор;
- засвоєння особливостей реалізації циклу розробки та базових елементів технології розробки комп'ютерної гри;
- набуття вміння використовувати сучасний інструментарій для візуальної розробки ігрових додатків;
- оволодіння навичками командної розробки простих крос-платформних ігрових додатків.

Очікувані результати

В результаті вивчення дисципліни «Основи розробки комп'ютерних ігор» студент має отримати:

- знання методів проектування, підтримки та просування і технології розробки комп'ютерних ігор;
- знання і вміння реалізовувати базові елементи циклу розробки типової комп'ютерної гри;
- вміння наскрізного проектування простих комп'ютерних ігор різних жанрів, від побудови сценарію до просування гри;
- навички командної розробки простих крос-платформних комп'ютерних ігор різних жанрів з використанням сучасних інструментів візуальної розробки.

Структура дисципліни

Дисципліна «Основи розробки комп'ютерних ігор» складається з наступних змістовних модулів:

- М4.1. Проектування комп'ютерних ігор;
- М4.2. Візуальна розробка комп'ютерних ігор;
- М4.3. Виконання групового фінального проекту.

Модуль М4.1. «Проектування комп'ютерних ігор» включає теоретичні та практичні аспекти проектування, розробки, підтримки та просування комп'ютерних ігор.

Модуль М4.2. «Візуальна розробка комп'ютерних ігор» включає практичні аспекти візуальної розробки комп'ютерних ігор з використанням сучасного інструментарію.

Модуль М4.3. «Виконання групового фінального проекту» є завершальним в навчальній програмі дисципліни та призначений для набуття студентами практичних навичок командної розробки простих крос-платформних комп'ютерних ігор.

2. ОПИС ДИСЦИПЛІНИ ТА ЇЇ МОДУЛІВ

Дисципліна «Основи розробки комп'ютерних ігор»

Галузь знань: 12 «Інформаційні технології».

Спеціальність: 121 «Інженерія програмного забезпечення»

Рівень: бакалавр.

Семестр: 5

Кількість кредитних одиниць ECTS: 4,0.

Кількість годин:

- всього акад. годин – 120,
- з них лекцій – 16 год.,
- практичних занять – 6 год.,
- лабораторних занять – 22 год.,
- семінарів – 4 год.,
- самостійної роботи – 72 год.

Форма контролю: залік, груповий фінальний проект.

Викладач: д.т.н., професор Шерстюк В.Г.

Модуль М4.1. Проектування комп'ютерних ігор

Кількість кредитних одиниць ECTS: 1,0.

Кількість годин:

- всього акад. годин – 30,
- з них лекцій – 8 год.,
- практичних занять – 6 год.,
- лабораторних занять – 4 год.,
- самостійної роботи – 12 год.

Модуль М4.2. Візуальна розробка комп'ютерних ігор

Кількість кредитних одиниць ECTS: 2,0.

Кількість годин:

всього акад. годин – 60 год.,
з них лекцій – 8 год.,
лабораторних занять – 18 год.,
семінари – 2 год.,
самостійної роботи – 32 год.

Модуль М4.3. Виконання групового фінального проекту

Кількість кредитних одиниць ECTS: 1,0.

Кількість годин:

всього акад. годин – 30,
з них семінарів – 2 год,
самостійної роботи – 28 год.

3. ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ ТА РЕЗУЛЬТАТИ НАВЧАННЯ*

ЗАГАЛЬНІ (УНІВЕРСАЛЬНІ) КОМПЕТЕНТНОСТІ

ЗК-1 Здатність до абстрактного мислення, аналізу та синтезу.

ЗК-2 Здатність застосовувати знання в практичних ситуаціях.

ЗК-3 Здатність вчитися й оволодівати сучасними знаннями, здійснювати пошук, оброблення й аналіз інформації з різних джерел.

ЗК-4 Здатність генерувати нові ідеї (креативність), працювати в команді, бути критичним і самокритичним, розробляти проекти, приймати обґрунтовані рішення.

ЗК-5 Здатність оцінювати та забезпечувати якість виконуваних робіт.

ЗК-6 Здатність застосовувати математичний апарат, а також теоретичні, методичні й алгоритмічні основи інформаційних технологій під час вирішення прикладних і наукових завдань в області інформаційних систем і технологій.

СПЕЦІАЛЬНІ (ФАХОВІ) КОМПЕТЕНТНОСТІ

ФК-1 Здатність опанувати сучасні технології математичного моделювання об'єктів, процесів і явищ, розробляти обчислювальні моделі та алгоритми чисельного розв'язання задач математичного моделювання з урахуванням похибок наближеного чисельного розв'язання професійних задач; здійснювати формалізований опис задач дослідження операцій в організаційно-технічних і соціально-

* Освітньо-професійна програма першого (бакалаврського) рівня вищої освіти. Спеціальність 121 Інженерія програмного забезпечення, спеціалізація Розробка комп'ютерних ігор.

економічних системах різного призначення, визначати їх оптимальні рішення, будувати моделі оптимального вибору управління з урахуванням змін параметрів економічної ситуації, оптимізувати процеси управління в системах різного призначення та рівня ієрархії.

- ФК-2* Здатність реалізовувати багаторівневі обчислювальні моделі на основі архітектури клієнт-сервер (включаючи сховища, бази та банки даних і знань) для забезпечення обчислювальних потреб багатьох користувачів.
- ФК-3* Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами, технічним завданням та стандартами.
- ФК-4* Здатність здійснювати аналіз і функціональне моделювання процесів, побудову та застосування функціональних моделей систем; застосування методів та інструментальних засобів для управління процесами життєвого циклу систем відповідно до вимог замовника.
- ФК-5* Здатність опанувати та комплексно застосовувати базові загальні знання в області програмування (у тому числі, структурного, функціонального, логічного, об'єктно-орієнтованого, паралельного) та візуального проектування системного та прикладного програмного забезпечення; володіти алгоритмічним мисленням; проектувати та розробляти програмне забезпечення на основі інтеграції провідних сучасних технологій (із застосуванням відповідних моделей, методів та алгоритмів обчислень, структур даних); застосовувати об'єктно-орієнтований підхід під час проектування складних програмних систем методами програмної інженерії для реалізації програмного забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик.
- ФК-6* Здатність опанувати та комплексно застосовувати базові знання в області принципів, методів і алгоритмів комп'ютерної графіки під час розробки графічних інтерфейсів взаємодії людини з комп'ютером.
- ФК-7* Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності загальної функціональності і надійності програмного забезпечення.

ПРОГРАМНІ РЕЗУЛЬТАТИ НАВЧАННЯ

- ПРН-1* Здатність аналізувати проблеми щодо створення програмного забезпечення.
- ПРН-2* Здатність, аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.
- ПРН-3* Здатність використовувати знання щодо методів та засобів збору, формулювання та аналізу вимог до програмного забезпечення.

- ПРН-4* Здатність застосовувати знання ефективних підходів щодо проектування програмного забезпечення.
- ПРН-5* Здатність розуміти основні процеси, фази та ітерації життєвого циклу програмного забезпечення.
- ПРН-6* Здатність розуміти і застосовувати знання сучасних підходів щодо оцінки та забезпечення якості програмного забезпечення.
- ПРН-7* Здатність розуміти і застосовувати знання щодо відповідних математичних понять, методів доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення.
- ПРН-8* Здатність розуміти і застосовувати знання щодо відповідних математичних понять, методів доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення.
- ПРН-9* Здатність застосовувати на практиці фундаментальні концепції і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.
- ПРН-10* Здатність застосовувати знання методів компонентної розробки програмного забезпечення, виділяючи інтерфейси і реалізації та взаємодію між модулями, підсистемами і компонентами.

4. МІЖДИСЦИПЛІНАРНІ ЗВ'ЯЗКИ

Дисципліна «Основи розробки комп'ютерних ігор» є стартовою для фахової підготовки студентів за спеціалізацією «Розробка комп'ютерних ігор». Дисципліни, що є попередніми та забезпечують засвоєння навчального матеріалу дисципліни «Основи розробки комп'ютерних ігор»:

1. Основи програмування.
2. Алгоритми та структури даних.
3. Основи програмної інженерії.
4. Основи комп'ютерної графіки.

5. ФОРМИ НАВЧАННЯ

Навчальний процес полягає у проведенні навчальних занять аудиторно або дистанційно з використанням платформи Moodle (або іншої використовуваної в університеті) у форматі лекцій, практичних занять, лабораторних робіт, у виконанні студентами самостійної роботи, групових фінальних проєктів, контрольних заходів.

Аудиторна робота включає лекції, практичні заняття та лабораторні роботи.

Лекція є основною формою проведення навчальних занять, що призначена для засвоєння теоретичного матеріалу.

Лабораторні заняття є формою навчальних занять, на яких студенти працюють в командах (малих робочих групах), відпрацьовуючи навички і

вміння розробки ігрових додатків шляхом послідовного виконання наскрізного завдання, використовуючи певне ігрове комп'ютерне обладнання та інструментальне програмне забезпечення.

Практичне заняття є формою навчальних занять, на яких студенти працюють в командах (малих робочих групах), вирішуючи певні завдання проектування ігрового додатку, презентуючи та обговорюючи отримані результати.

Семінар є формою навчальних занять, на яких студенти презентують перед аудиторією результати виконання завдань, обговорюють в загальній дискусії отримані результати та оцінюють результати роботи колег.

Консультації є формою навчальних занять, на яких студент отримує відповіді викладача на конкретні запитання або пояснення певних теоретичних положень чи практичних аспектів роботи з середовищем візуальної розробки.

Самостійна робота спрямована на отримання студентом нових знань шляхом оволодіння навчальним матеріалом та самостійного вирішення завдань. Самостійна робота студентів передбачає підготовку до навчальних занять і контрольних заходів та здійснюється у вільний від аудиторних навчальних занять час. Підготовка до поточних аудиторних занять полягає в аналізі літератури, інтернет-матеріалів за тематикою навчального модуля, оформленні звітів за виконаними лабораторними роботами.

Контактні години передбачають індивідуальні консультації та он-лайн контроль студентів з використанням системи дистанційного навчання Moodle (або іншої використовуваної в університеті).

6. ПОРЯДОК ПРОВЕДЕННЯ АТЕСТАЦІЇ З ДИСЦИПЛІНИ

В навчальному процесі при вивченні студентами навчальної дисципліни використовуються засоби поточного та підсумкового контролю. Застосовується 100-бальна система оцінювання.

Поточний контроль полягає в:

- оцінюванні результатів виконання студентами лабораторних робіт;
- оцінювання виконання завдань та обговорення результатів на практичних заняттях;
- оцінювання участі студентів в обговоренні на семінарі;
- оцінюванні виконаних студентами групових фінальних проєктів при їх обговоренні перед студентською аудиторією на заключному семінарі;
- оцінюванні модульних контрольних робіт наприкінці вивчення змістових модулів у форматі тесту (максимально 4 бали).

Підсумковий контроль має форму заліку, що проводиться наприкінці семестру у форматі тесту (максимально 40 балів).

Рейтингове оцінювання навчальних успіхів студента виконується шляхом простого підсумовування всіх результатів поточного та підсумкового

контролю. Мінімальна кількість балів, що зараховується як позитивний результат, дорівнює 60 (за 100-бальною системою).

Подання та оцінювання звіту про виконання лабораторних робіт

Лабораторні роботи виконуються малими робочими групами студентів протягом аудиторного заняття. У кінці кожного заняття група демонструє викладачу виконане завдання.

За результатами виконання лабораторної роботи студенти мають самостійно оформити звіт, в якому зазначити основні кроки, що були виконані у ході даної роботи, та отримані за ними результати.

Звіт подається викладачу засобами системи дистанційного навчання Moodle (або іншої використовуваної в університеті) не пізніше одного тижня після виконання лабораторної роботи. За кожен тиждень відтермінування подання звіту нараховується -1 штрафний бал.

Викладач оцінює повноту, самостійність та своєчасність виконання студентами лабораторної роботи за 4-бальною шкалою, використовуючи критерії оцінювання, наведені в Додатку 1.

Узагальнені результати виконання курсу лабораторних робіт за результатами студентських звітів заслуховуються та обговорюються у студентському середовищі на прикінцевому семінарському занятті на останньому тижні модуля.

Оцінювання виконання завдань та обговорення результатів на практичних заняттях

На практичних заняттях студенти працюють малими групами, паралельно виконуючи завдання, поставлені викладачем, протягом аудиторного заняття.

У кінці кожного практичного заняття проводиться обговорення результатів, отриманих студентами виконанні завдань в групах.

Кожна група демонструє результати виконання завдань перед аудиторією, та відповідає на питання, які виникають у викладача та студентів.

Викладач оцінює оригінальність та коректність виконання завдань студентом та його участь у обговоренні за 4-бальною шкалою, використовуючи критерії оцінювання, наведені в Додатку 2.

Оцінювання участі студентів в обговоренні на семінарі

На семінарських заняттях студенти малими робочими групами презентують отримані результати виконання завдань перед аудиторією та обговорюють їх у загальній дискусії.

Наприкінці семінару студенти оцінюють навчальні успіхи друг друга.

Критерії оцінювання обговорення результатів на семінарах наведені в Додатку 3.

Подання та оцінювання групового фінального проекту

На першому тижні навчальних занять за дисципліною відбувається рандомне формування малих робочих груп студентів (чисельністю по 4-5 осіб), кожна з яких буде виконувати фінальний груповий проект.

Впродовж навчального модуля М4.1 малі робочі групи студентів виконують завдання на практичних заняттях, що дозволяє їм визначитися з проектом комп'ютерної гри, створюючи ігрову концепцію й ігрову модель, а потім моделюючи ігрове середовище.

Впродовж навчального модуля М4.2 малі робочі групи студентів виконують лабораторні роботи, засвоюючи певні навички розробки комп'ютерної гри за допомогою візуального середовища WIMIS.

Потім впродовж навчального модуля М4.3 малі робочі групи студентів, використовуючи отримані на попередньому модулі навички, реалізують концепцію і проект комп'ютерної гри, отримані як результат виконання першого модуля.

Звіт за результатами виконання проекту надається викладачеві через систему дистанційного навчання Moodle (або іншу використовувану університетом) не пізніше передостаннього (п'ятнадцятого) тижня навчання.

На останньому (шістнадцятому) тижні групи студентів на заключному семінарі презентують звіт за результатами проекту, обговорюють їх та отримують підсумкову оцінку.

Критерії оцінювання заключного звіту за результатами виконання групового проекту наведені в Додатку 4.

Продовження терміну подання проекту можливе лише при наявності поважної причини, передбаченої Положенням про організацію навчального процесу в університеті. За кожний тиждень прострочки подання фінального групового проекту знімається 2 бали (максимально 10 балів).

Оцінювання модульних контрольних робіт

Модульні контрольні роботи проводяться наприкінці останнього тижня вивчення змістовних модулів у формі тестування за допомогою системи дистанційного навчання Moodle (або іншої використовуваної в університеті).

Тести складаються з теоретичних питань та практичних завдань за навчальним матеріалом змістовного модуля.

Модульна контрольна робота оцінюється системою дистанційного навчання без участі викладача за 4-бальною шкалою.

Підсумковий контроль

Підсумковий контроль (залікове тестування) проводиться наприкінці останнього (шістнадцятого) тижня навчальних занять у формі тестування за допомогою системи дистанційного навчання Moodle (або іншої використовуваної в університеті).

Тести складаються з теоретичних питань та практичних завдань за навчальним матеріалом всіх змістових модулів дисципліни.

Підсумкове залікове тестування оцінюється системою дистанційного навчання без участі викладача за 40-бальною шкалою.

Метод оцінки дисципліни в цілому

Оцінки студентів за результатами вивчення змістовних модулів підсумовуються. Підсумкова залікова оцінка (до 40 балів) додається.

Таким чином розраховується сумарна підсумкова оцінка студента в балах за навчальною дисципліною.

Вид оцінювання	Оцінка
Оцінка змістовного модуля М4.1.	до 20
Оцінка змістовного модуля М4.2.	до 20
Оцінка змістовного модуля М4.3.	до 20
Залікове тестування	до 40
Разом	до 100

Графік проведення оцінювання з навчальної дисципліни

Номер тижня	Оцінювання
4	Оцінка змістовного модуля М4.1.
8	Оцінка змістовного модуля М4.2.
15	Подання фінального групового проекту
16	Оцінка фінального групового проекту
16	Підсумкове залікове тестування

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

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

7. ЗВОРОТНІЙ ЗВ'ЯЗОК

Інформація щодо результатів виконання практичних і лабораторних робіт надається кожному студенту:

- а) індивідуально та прилюдно наприкінці заняття;
- б) дистанційно через платформу Moodle (або іншу використовувану університетом).

Тестування виконується виключно засобами системи дистанційного навчання (Moodle або іншої використовуваної університетом).

Інформація щодо результатів тестування надається студентам по його завершенню дистанційно через платформу Moodle (або іншу використовувану університетом).

Інформація щодо результатів презентації та обговорення лабораторного курсу (модуль М4.2) та групового проекту (модуль М4.3) надається наприкінці відповідного семінарського заняття.

Загальні результати за дисципліною (у формі заліку) надаються на 16 тижні навчання після завершення навчальних занять та підсумкового тестування.

Консультації за дисципліною проводяться викладачем очно у відповідності до робочого графіку та заочно засобами системи дистанційного навчання Moodle (або іншої використовуваної університетом), електронного листування та/або електронної конференції на протязі вивчення навчальної дисципліни згідно графіку навчального процесу.

Посилання на сторінку дисципліни «Основи розробки комп'ютерних ігор» на платформі Moodle: [URL]

Контактні дані для on-line допомоги та консультування:

викладач: д.т.н., професор Шерстюк Володимир Григорович,

e-mail: vgsherstyuk@gmail.com

Skype: vsherstyuk

8. ВИКЛАДАЦЬКИЙ СКЛАД ТА ДОПОМІЖНІ ДЖЕРЕЛА

Обов'язки викладача

Обов'язки викладача полягають у проведенні лекцій, практичних і лабораторних занять відповідно до навчальної програми та графіку навчального процесу, а також контролю якості отриманих знань, умінь і навичок.

Обов'язки координатора дисципліни

Обов'язки координатора полягають у розробці та внесенні змін до навчальних модулів у відповідності до освітньо-професійних програм та навчальних планів; координації і керівництві роботою професорсько-

викладацького складу щодо організації навчального процесу та контролю якості навчання.

Обов'язки допоміжного персоналу

Допоміжний персонал здійснює підготовку комп'ютерної техніки та спеціалізованого ігрового обладнання до виконання практичних і лабораторних робіт студентами та надає технічну підтримку студентам під час їх виконання.

Контактні дані викладача:

д.т.н., професор Шерстюк Володимир Григорович,
e-mail: vgsherstyuk@gmail.com
Skype: vsherstyuk

9. МОДУЛЬ М4.1 ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ ІГОР

Опис модуля

Галузь знань:	12 «Інформаційні технології».
Спеціальність:	121 «Інженерія програмного забезпечення»
Рівень:	бакалавр
Назва дисципліни:	Основи розробки комп'ютерних ігор
Назва модуля:	Проектування комп'ютерних ігор
Семестр:	4
Кредитних одиниць:	1,0
Академічних годин:	30,
з них лекцій	– 8 год.,
практ. занять	– 6 год.,
лабор. робіт	– 4 год.,
самостійної роботи	– 12 год.
Викладач:	д.т.н., професор Шерстюк В.Г.

Вступ

Модуль «Проектування комп'ютерних ігор» є вступним.

Модуль включає базовий навчальний матеріал щодо основних понять у галузі комп'ютерних ігор, з акцентом на особливості побудови ігрової моделі та ігрового середовища для ігор різних жанрів та на реалізацію життєвого циклу комп'ютерної гри.

Мета модуля

Метою модуля є формування у студентів системи початкових понять, знань, умінь і навичок в галузі розробки ігрових додатків, що включає до себе методи проектування і розробки ігрових програмних продуктів, розвиток креативного і логічного мислення, формування наукового світогляду, прищеплення схильності до творчості.

Результати навчання

Знання та їх використання

У випадку успішного оволодіння матеріалами даного модуля студент буде знати базові поняття предметної галузі; особливості ігор різних типів та сфери їх застосування; організацію ігрового процесу та основні етапи проектування і розробки гри; учасників процесу, їх ролі та необхідні професійні компетенції; структуру та архітектуру ігрового додатку та принципи його побудови; життєвий цикл комп'ютерної гри та взаємозв'язки його етапів; особливості проектування, розробки та виробництва ігрових додатків; особливості створення ігрових ідей, концепцій, моделей та

середовищ; номенклатуру та особливості ігрових рушіїв й інструментальних засобів розробки комп'ютерних ігор.

Дослідницькі навички

У разі успішного вивчення модуля студент буде вміти створювати креативні концепції гри та оцінювати коло потенційних гравців, аналізувати можливість проектування комп'ютерної гри у відповідності до прийнятої концепції, включаючи ігрову модель та ігрове середовище.

Спеціальні вміння

У разі успішного вивчення модуля студент буде вміти: застосовувати сучасні методи проектування, розробки та виробництва ігрових додатків; обирати інструменти розробки у відповідності до поставлених вимог; визначати ігрову концепцію; створювати сюжет, визначати та описувати персонажів, рівні, правила гри; моделювати сценарій гри та виконувати його розкадровку; визначати шляхи проходження гри; здійснювати дизайн гри, ігрових персонажів та об'єктів, враховуючи їх ігрові ролі; виявляти вплив прийнятих дизайнерських рішень та їх динаміки на психоемоційний стан гравця; визначати і вибудовувати зв'язки між ігровими персонажами та подіями в сценарії гри, забезпечуючи динаміку ігрового середовища.

Соціальні вміння

У разі успішного вивчення модуля студент буде вміти працювати в складі професійної команди розробників, усвідомлюючи свої персональні обов'язки та відповідальність за результати роботи.

Особисті якості

У разі успішного вивчення модуля студент розвине такі особисті якості:

- креативний спосіб мислення при генерації ідей стосовно створення концепції та проекту комп'ютерної гри
- самостійно вирішувати питання, що відносяться до створення комп'ютерних ігор;
- впевнено обґрунтовувати актуальність та презентувати ігрові концепції та проекти;
- систематизувати професійні знання; аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси з урахуванням сучасних досягнень науки і техніки.

Перелік компетентностей та результати навчання

Загальні (універсальні) компетентності	ЗК-4, ЗК-5, ЗК-6
--	------------------

Спеціальні (фахові) компетентності	ФК-1, ФК-2, ФК-3, ФК-5, ФК-6
Програмні результати навчання	ПРН-4, ПРН-7, ПРН-8, ПРН-10

Календарний план семестру і структура модуля М4.1.

Місце модуля М4.1 в структурі дисципліни

Номер	Модулі дисципліни	Тиждень навчання
1	М4.1 Проектування комп'ютерних ігор	1-4
2	М4.2 Візуальна розробка комп'ютерних ігор	5-8
3	М4.3 Виконання групового фінального проекту	9-16

Інформаційне наповнення модуля М4.1

Номер тижня	Зміст
1	Комп'ютерна гра як об'єкт розробки
2	Цикл розробки комп'ютерної гри
3	Розробка ігрової моделі
4	Розробка ігрового середовища

Порядок слідування занять по тижням

Тиждень навчання			
1	Лекція 1	Лабор. 1	Лабор. 1
2	Лекція 2	Пр. зан. 1	
3	Лекція 3	Пр. зан. 2	
4	Лекція 4	Пр. зан. 3	

Форми навчання

Навчальний процес полягає у проведенні навчальних занять аудиторно або дистанційно з використанням платформи Moodle (або іншої) у форматі лекцій, практичних занять, лабораторних робіт, у виконанні студентами самостійної роботи та контрольних заходів.

Аудиторна робота включає 8 годин лекцій, 6 годин практичних занять та 4 години лабораторної роботи.

Самостійна робота студентів в обсязі 12 годин передбачає підготовку до навчальних занять і контрольних заходів та здійснюється у вільний від аудиторних навчальних занять час.

Контактні години передбачають індивідуальні консультації та он-лайн контроль студентів з використанням платформи Moodle.

Порядок проведення атестації

Модуль включає лекції, лабораторну роботу та практичні заняття.

Поточне оцінювання студента передбачає оцінювання викладачем виконання і захисту студентом лабораторної роботи, а також виконання студентом завдань на практичних заняттях.

Підсумкове оцінювання модуля передбачає виконання модульної контрольної роботи.

Графік проведення оцінювання модуля М4.1

Номер тижня	Оцінювання
1	Поточна оцінка за виконання та захист лабораторної роботи №1
2	Поточна оцінка за виконання завдання практичної роботи №1
3	Поточна оцінка за виконання завдання практичної роботи №2
4	Поточна оцінка за виконання завдання практичної роботи №3
4	Підсумкова оцінка за модульну контрольну роботу

Метод оцінювання модуля

Модуль оцінюється за рейтинговою системою за 20 бальною шкалою. Рейтингова оцінка модуля є накопичувальною та включає всі поточні оцінки та підсумкову оцінку модульної контрольної роботи.

Вид завдання	Оцінка
Поточна оцінка за виконання та захист лабораторної роботи №1	4
Поточна оцінка за виконання завдання практичної роботи №1	4
Поточна оцінка за виконання завдання практичної роботи №2	4
Поточна оцінка за виконання завдання практичної роботи №3	4
Підсумкова оцінка модульної контрольної роботи	4

Максимальна сумарна поточна оцінка за всі практичні та лабораторну роботи в межах навчального модуля складає 16 балів.

Оцінювання виконання та захисту студентом лабораторної роботи виконується за критеріями, наведеними в Додатку 1.

Оцінювання виконання студентом завдань на практичних заняттях виконується за критеріями, наведеними в Додатку 2.

Всі набрані студентом бали підсумовуються (максимально студент може отримати 20 балів), штрафні бали віднімаються (максимально мінус 4 бали).

Сумарна оцінка (від 0 до 20 балів) є індивідуальною рейтинговою оцінкою засвоєння студентом навчального модуля М4.1.

Оцінювання результатів тестування

Модульна контрольна робота проводиться наприкінці 4-го тижня модуля М4.1 у форматі тестування за допомогою системи дистанційного навчання Moodle (або іншої використовуваної університетом).

Тест складається з низки теоретичних питань, що охоплюють всі теми навчального модуля М4.1.

Модульна контрольна робота оцінюється у 4 бали.

Навчальна програма та матеріали модуля М4.1

Лекція 1. Комп'ютерна гра як об'єкт розробки

Анотація

Лекція знайомить з основними поняттями предметної галузі комп'ютерних ігор, історією розвитку комп'ютерних ігор, класифікацією ігор та особливостями різних їх класів, особливостями гравців та ігрового процесу, етапами розробки комп'ютерної гри, вимогами до розробників комп'ютерних ігор.

Мета лекції

Ознайомити студентів з предметною галуззю та етапами її історичного розвитку, забезпечити розуміння основних понять та особливостей цієї галузі, особливостей ігор різних типів та класів, пояснити та формалізувати різні аспекти організації ігрового процесу, розкрити зв'язок з методами та технологією проектування і розробки ігор, професійними компетенціями розробників ігор.

Очікувані результати

Студент має:

- оволодіти базовими поняттями предметної галузі: комп'ютерна гра, гравець, геймплей, геймдизайн, геймконтент тощо;
- розуміти та вміти аналізувати особливості ігор різних типів з погляду на сфери їх застосування та на процес їх проектування й розробки;
- знати різні аспекти організації ігрового процесу та вміти використовувати їх під час розробки гри;
- знати основні етапи процесу проектування і розробки комп'ютерної гри та учасників процесу, розуміти їх роль та необхідні професійні компетенції.

Контрольні запитання

- Що є комп'ютерною грою?

- Що таке ігровий процес?
- Які є види гравців?
- Як враховуються особливості різних типів гравців при розробці гри?
- Які різновиди комп'ютерних ігор Ви знаєте?
- Які є сфери застосування ігор різних типів?
- Які етапи розвитку комп'ютерних ігор Ви знаєте?
- Які етапи розробки комп'ютерної гри Ви знаєте?
- Який склад має команда розробників гри?
- Які ролі відіграють члени команди розробників в процесі розробки гри?
- Які вимоги до професійних компетенцій кожного з учасників команди розробників гри?
- Що таке ігровий жанр?
- Що таке візуалізація гри?

Лекція 2. Цикл розробки комп'ютерної гри.

Анотація

Лекція знайомить з життєвим циклом комп'ютерної гри та його етапами. Розглядаються методи проектування, розробки та виробництва ігрових додатків, пояснюються особливості командної роботи, структура та принципи побудови комп'ютерної гри, використання ігрових рушіїв та інструментальних засобів розробки комп'ютерних ігор.

Мета лекції

Ознайомити студентів з життєвим циклом комп'ютерної гри, пояснити особливості реалізації його етапів та розкрити взаємозв'язки його етапів. Розкрити особливості структури та архітектури ігрового додатку та визначити принципи його побудови, ознайомити студентів з сучасними методами проектування, розробки та виробництва ігрових додатків, номенклатурою та особливостями використання ігрових рушіїв та інструментальних засобів розробки комп'ютерних ігор.

Очікувані результати

Студент має:

- знати структуру ігрового додатку та розуміти принципи його побудови, вміти побудувати архітектуру ігрового додатку відповідно до поставлених вимог;
- знати сутність життєвого циклу комп'ютерної гри та розуміти взаємозв'язки його етапів;

- знати сучасні методи проектування, розробки та виробництва ігрових додатків та вміти коректно їх використовувати на практиці;
- знати номенклатуру та особливості використання ігрових рушіїв й інструментальних засобів розробки комп'ютерних ігор, вміти правильно обирати інструменти розробки у відповідності до поставлених вимог.

Контрольні запитання

- Що таке ігровий рушій?
- Що таке ігровий конструктор?
- Що таке життєвий цикл комп'ютерної гри?
- З яких етапів складається життєвий цикл комп'ютерної гри?
- В чому полягають задачі кожного з етапів життєвого циклу комп'ютерної гри?
- В чому полягає взаємозв'язок етапів життєвого циклу комп'ютерної гри?
- Яку структуру має ігровий додаток?
- В чому полягають особливості архітектури ігрових додатків?
- Які існують методи проектування та розробки ігрових додатків?
- В чому полягають особливості командної розробки КІД?
- Які існують види інструментальних засобів розробки ігрових додатків?
- Якими є сфери застосування різних типів ігрових рушіїв?

Лекція 3. Розробка ігрової моделі.

Анотація

Лекція знайомить з особливостями створення геймплею комп'ютерної гри, різними підходами до розробки ігрової ідеї та концепції, підходами до визначення сюжету та персонажів гри, ігрових рівнів, формалізації правил гри та написання відповідного сценарію. Розглядаються поняття ігрового стану, сцени, сценарію та моделі їх подання.

Мета лекції

Ознайомити студентів з принципами та методами розробки ігрової ідеї та концепції, створення сюжету та персонажів гри, ігрових правил та сценаріїв, з моделями подання сцен та станів та визначення шляхів проходження гри.

Очікувані результати

Студент має:

- володіти основними поняттями ігрової моделі;
- розуміти особливості створення ігрової ідеї та вміти визначати ігрову концепцію;

- вміти створювати сюжет, визначати та описувати персонажів, рівні, правила гри;
- вміти моделювати сценарій гри та виконувати розкадровку;
- вміти визначати шляхи проходження гри.

Контрольні запитання

- Що таке геймплей?
- Що таке геймабіліті?
- Що таке ігрова модель та з чого вона складається?
- Що таке сюжет гри?
- Що таке стан гри?
- Що таке сцена гри?
- Як описати правила гри?
- Як подаються сценарії гри?
- Що таке ігровий баланс?
- Як визначити множину шляхів проходження гри?
- Як забезпечити ігровий баланс у комп'ютерній грі?
- Що таке крива розвитку гравця та як вона залежить від сценарію?
- Що таке Storytelling?
- Що таке Storyboard?

Лекція 4. Розробка ігрового середовища.

Анотація

Лекція знайомить з особливостями створення ігрового середовища, дизайну комп'ютерної гри, ігрових об'єктів та персонажів, розкриває взаємозв'язки між сюжетом, персонажами гри та ігровими об'єктами, особливостями дизайну ігрових персонажів та об'єктів та їх психологічними ефектами, а також принципами використання графіки, анімації та звукових ефектів для створення ігрових ресурсів, що забезпечують певний психічний та психоемоційний вплив на гравця.

Мета лекції

Ознайомити студентів з принципами та методами розробки ігрового середовища, ігрових об'єктів та персонажів, дизайну гри та сучасними підходами до створення ігрових ресурсів.

Очікувані результати

Студент має:

- володіти основними поняттями ігрового середовища;
- розуміти особливості та вміти здійснювати дизайн гри, ігрових персонажів та об'єктів, враховуючи їх ігрові ролі;

- знати особливості створення ігрових ресурсів з використанням сучасних графічних, анімаційних, відео- та звукових ефектів та вміти враховувати їх взаємодію;
- вміти виявляти вплив прийнятих дизайнерських рішень та їх динаміки на психоемоційний стан гравця;
- вміти визначати і вибудовувати зв'язки між ігровими персонажами та подіями в сценарії гри, забезпечуючи динаміку ігрового середовища.

Контрольні запитання

- З чого складається ігрове середовище?
- Від чого залежить динаміка ігрового середовища?
- Що таке ігрова роль?
- Що таке персонаж?
- В чому полягає динаміка персонажу?
- Які ігрові об'єкти належать до ігрового середовища?
- Що таке ігровий ресурс?
- Які є форми існування ігрового ресурсу?
- Як створюються персонажі та об'єкти у комп'ютерній грі?
- Як добирається дизайн персонажів в залежності від їх ролей в комп'ютерній грі?
- Яким є вплив візуальних та звукових ефектів на сприйняття гравцем ігрового середовища?
- Які інструментальні засоби використовуються для створення ігрових персонажів, об'єктів та ресурсів?
- В чому полягає взаємодія між ігровими ресурсами, об'єктами та персонажами ігрового середовища?

Практичне заняття 1. Створення ігрової ідеї та концепції.

Анотація

Практичне заняття присвячене створенню ідеї та концепції комп'ютерної гри при роботі в команді. Виконується малими групами студентів, студенти попередньо розподіляються по групах по 4-5 осіб. Отримана в результаті виконання практичної роботи ігрова концепція використовується надалі для виконання практичних та лабораторних робіт з дисципліни та для виконання групових проєктів.

Мета практичного заняття

Вироблення та презентація загальної ідеї та концепції комп'ютерної гри.

Очікувані результати

Студенти мають отримати навички:

- генерації загальної ідеї комп'ютерної гри з врахуванням способів та інструментів її проектування та розповсюдження;
- аналізу особливостей та переваг запропонованої ідеї гри задля виявлення кола потенційних гравців та обґрунтування актуальності;
- розробки концепції комп'ютерної гри з врахуванням аналізу цільової групи.

Контрольні запитання

- В чому полягає сутність ідеї комп'ютерної гри?
- З чого складається концепція комп'ютерної гри?
- Як необхідно окреслювати та втілювати ідею комп'ютерної гри?
- В чому полягає послідовність створення концепції комп'ютерної гри?
- Як окреслити переваги запропонованої концепції комп'ютерної гри?
- Як знайти коло потенційних гравців?
- Як обґрунтувати актуальності комп'ютерної гри?
- Як необхідно презентувати концепцію комп'ютерної гри для пошуку інвестора?

Практичне заняття 2. Розробка ігрової моделі.

Анотація

Практичне заняття присвячене командній розробці ігрової моделі, виконується малими групами студентів за концепціями гри, запропонованими групами на попередньому практичному занятті.

Мета практичного заняття

Формування навичок створення ігрової моделі, починаючи від сюжету, персонажів, рівнів та правил гри, та закінчуючи створенням ігрового сценарію.

Очікувані результати

Студенти мають отримати навички:

- storytelling, створення сюжету та персонажів гри;
- визначення та подання правил гри та рівнів;
- визначення сцен та виявлення шляхів проходження;
- опису взаємодії персонажів;
- побудови та подання сценарію гри;.

Контрольні запитання

- В чому полягає розробка ігрової моделі?
- Як створюють сюжет гри?
- Що таке Storytelling?
- Що таке персонаж та які типи персонажів бувають?

- Як створюють персонажі?
- Що таке рівень гри?
- Чим лінійна гра відрізняється від нелінійної?
- Які є способи подання правил гри?
- Як оцінюються ігрові досягнення гравця?
- В чому полягає нарахування бонусів та штрафних санкцій?
- Як виявляють взаємозалежності між персонажами в комп'ютерній грі?
- Як виявляють всі можливі шляхи проходження гри?
- Які є способи подання сценаріїв гри?
- З чого складається сценарій гри?
- Що таке Storyboard?

Практичне заняття 3. Розробка та дизайн ігрового середовища.

Анотація

Практичне заняття присвячене командній розробці ігрового середовища, виконується малими групами студентів за концепцією гри та ігровою моделлю, запропонованими групами на попередніх практичних заняттях.

Мета практичного заняття

Формування навичок розробки та опрацювання дизайну ігрового середовища, ігрових персонажів, об'єктів та ресурсів з використанням сучасних візуальних та звукових ефектів.

Очікувані результати

Студенти мають отримати навички:

- проектування ігрового середовища;
- опрацювання дизайну ігрових персонажів та об'єктів з врахуванням їх ігрових ролей та динаміки їх подання у відповідності до сценарію гри;
- створення ігрових ресурсів з використанням сучасних графічних, анімаційних, відео- та звукових ефектів з врахуванням їх взаємодії;
- синхронізації ігрових персонажів та подій за сценарієм гри, забезпечуючи динаміку ігрового середовища.

Контрольні запитання

- Що таке ігрове середовище?
- Що є персонажем гри?
- Що є ресурсом гри?
- Що є об'єктом гри?
- За якими принципами виконується дизайн персонажів?
- Які інструменти використовують для створення ігрових ресурсів?

- Які ефекти можуть бути використані для подання ігрових ресурсів?
- Як забезпечується динаміка гри?
- Які різновиди графіки використовуються в комп'ютерних іграх та для вирішення яких задач?
- Яким є взаємозв'язок ігрової моделі та ігрового середовища?

Лабораторна робота №1. Знайомство з середовищем розробки.

Анотація

Лабораторна робота зорієнтована на вивчення студентами техніки і прийомів роботи з середовищем візуальної розробки комп'ютерних ігор WIMI5. Лабораторна робота розрахована на 4 академічні години.

Мета лабораторної роботи

Вивчити функціональні можливості та засвоїти навички і прийоми роботи в середовищі візуальної розробки комп'ютерних ігор WIMI5.

Очікувані результати

Студент має отримати:

- знання функціональних можливостей та розуміння технології роботи в середовищі візуальної розробки комп'ютерних ігор WIMI5;
- вміння орієнтуватися у інтерфейсі середовища візуальної розробки комп'ютерних ігор WIMI5 та навички використання доступного інструментарію розробки ігор.

Контрольні запитання

- Охарактеризуйте середовище візуальної розробки WIMI5.
- Поясніть функціональні можливості середовища WIMI5.
- Охарактеризуйте можливості роботи з ігровими об'єктами та ресурсами.
- Поясніть призначення вікон інтерфейсу середовища WIMI5.
- Поясніть призначення елементів інтерфейсу користувача середовища WIMI5.
- Поясніть структуру ігрового проекту, що створюється середовищем WIMI5.
- Назвіть сферу застосування середовища WIMI5.

Список джерел інформації по модулю В.1.

1. Shell, J. The Art of Game Design: A book of Lenses [Text] / J. Shell. – A K Peters/CRC Press; 2 edition, 2014. – 600 p.
2. Fullerton, T. Game Design Workshop: A Playcentric Approach to Creating Innovate Games [Text] / T. Fullerton. – A K Peters/CRC Press; 3 edition, 2014. – 535 p.
3. Dille, F., Platten, J. Z. The Ultimate Guide to Video Game Writing and Design [Text] / F. Dille, J.Z. Platten. – Lone Eagle, 2008. – 272 p.
4. Кapp, K. The Gamification of Learning and Instruction Fieldbook: Ideas into Practice [Text] / K. Kapp. – Pfeiffer, 2013. – 480 p.
5. Novak D. Game Development Essentials: An Introduction. – 2004.
6. Разработка игр под Android [Электронный ресурс] / Марцио Джером Ди. - Санкт- Петербург : Питер, 2014. - 224 с. : ил. - Режим доступа: ЭБС "Айбукс". - Неогранич. доступ. - ISBN 978-5-496-01080-1 : Б. ц.
7. Разработка игр на языке JavaScript [Электронный ресурс] / С. А. Беляев. - Москва : Лань", 2016. - Режим доступа: ЭБС "Издательство Лань". - Неогранич. доступ. - ISBN 978-5-8114-1980-7 : Б. ц.
8. ПСИХОЛОГИЯ И ПЕДАГОГИКА ИГРЫ [Электронный ресурс] : учебник и практикум / Смирнова Е.О., Рябкова И.А. - М. : Издательство Юрайт, 2016. - 223 с. - (Бакалавр. Академический курс). - Режим доступа: "ЭБС Юрайт". - 5 доступов. - ISBN 978-5-9916-6807-1 : 560.00 p.
9. Дашко Ю.В., Заика А.А. “Основы разработки компьютерных игр” - М.: “Форум” 2009.-350с.
10. Рейнбоу В., “Компьютерные игры”. Энциклопедия. – С.: “Питер” 2005.-732с.
11. Роллингз, Эндрю. “Проектирование и архитектура игр”: пер. с англ./ Э. Роллингз, Д. Моррис. - М.: Вильямс, 2006.- 1040с.

10.МОДУЛЬ М4.2. ВІЗУАЛЬНА РОЗРОБКА КОМП'ЮТЕРНИХ ІГОР

Опис модуля

Галузь знань:	12 «Інформаційні технології».
Спеціальність:	121 «Інженерія програмного забезпечення»
Рівень:	бакалавр
Назва дисципліни:	Основи розробки комп'ютерних ігор
Назва модуля:	Візуальна розробка комп'ютерних ігор
Семестр:	4
Кредитних одиниць:	2,0
Академічних годин:	60,
з них лекцій	– 8 год.,
лабор. робіт	– 18 год.,
семінари	– 2 год.
самостійної роботи	– 32 год.

Викладач: д.т.н., професор Шерстюк В.Г.

Вступ

Модуль «Візуальна розробка комп'ютерних ігор» є базовим для формування вмінь та навичок студента щодо застосування інструментарію візуальної розробки ігрових додатків.

Модуль включає навчальний матеріал щодо особливостей візуальної розробки комп'ютерних ігор з використання середовища WIMI5, починаючи від створення простих ігрових елементів і до реалізації складних динамічних ігрових сценаріїв.

Мета модуля

Метою модуля є формування у студентів системи знань, умінь і навичок в галузі візуальної розробки ігрових додатків за допомогою середовища WIMI5.

Результати навчання

Знання та їх використання

У випадку успішного оволодіння матеріалами даного модуля студент буде знати базові поняття організації розробки ігрового додатку у середовищі WIMI5; основні інструменти середовища розробки WIMI5 та способи їх використання; основні поняття програмної моделі гри та правила реалізації потоку логіки гри; доступні ігрові ресурси та особливості їх використання в сценарії гри; особливості застосування шаблонів для побудови ігор різних типів; основні засоби монетизації, тестування, налагодження, розгортання та публікації комп'ютерних ігор в середовищі WIMI5.

Дослідницькі навички

У разі успішного вивчення модуля студент буде вміти створювати креативні ігрові моделі й додатки засобами WIMI5.

Спеціальні вміння

У разі успішного вивчення модуля студент буде вміти: правильно обирати інструменти розробки у відповідності до поставлених вимог та використовувати їх під час розробки гри; коректно застосовувати при розробці гри доступні ігрові ресурси й об'єкти, а також об'єкти програмної моделі гри; реалізовувати потоки логіки гри; інтегрувати в проєкт JavaScript код для програмування подій та логіки гри; застосовувати інструменти монетизації, тестування, налагодження, розгортання, публікації та розповсюдження комп'ютерних ігор за допомогою WIMI5.

Соціальні вміння

У разі успішного вивчення модуля студент буде вміти працювати в складі професійної команди розробників, усвідомлюючи свої персональні обов'язки та відповідальність за результати роботи.

Особисті якості

У разі успішного вивчення модуля студент розвине такі особисті якості:

- креативний спосіб мислення при генерації ідей стосовно розробки комп'ютерної гри
- самостійно вирішувати питання, що відносяться до створення комп'ютерних ігор;
- впевнено обґрунтовувати рішення та презентувати проєкти;
- систематизувати професійні знання; аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційні ресурси з урахуванням сучасних досягнень науки і техніки.

Перелік компетентностей та результати навчання

Загальні (універсальні) компетентності	<i>ЗК-4, ЗК-5, ЗК-6</i>
Спеціальні (фахові) компетентності	<i>ФК-1, ФК-2, ФК-3, ФК-5, ФК-6</i>
Програмні результати навчання	<i>ПРН-4, ПРН-7, ПРН-8, ПРН-10</i>

Календарний план семестру і структура модуля М4.2.

Місце модуля М4.2 в структурі дисципліни

Номер	Модулі дисципліни	Тиждень навчання
1	М4.1 Проектування комп'ютерних ігор	1-4
2	М4.2 Візуальна розробка комп'ютерних ігор	5-8
3	М4.3 Виконання групового фінального проєкту	9-16

Інформаційне наповнення модуля М4.2

Номер тижня	Зміст
5	Створення ігор у середовищі WIM15
6	Редактор сюжету та редактор логіки у WIM15
7	Візуальне програмування у WIM15
8	Тестування, розгортання та публікація комп'ютерної гри

Порядок слідування занять по тижням

Тиждень навчання				
5	Лекція 1	Лабор. 1	Лабор. 1	Лабор. 1
6	Лекція 2	Лабор. 2	Лабор. 2	Лабор. 2
7	Лекція 3	Лабор. 3	Лабор. 3	Лабор. 3
8	Лекція 4	Семінар		

Форми навчання

Навчальний процес полягає у проведенні навчальних занять аудиторно або дистанційно з використанням платформи Moodle (або іншої) у форматі лекцій, лабораторних робіт та семінару, у виконанні студентами самостійної роботи та контрольних заходів.

Аудиторна робота включає 8 годин лекцій, 18 годин лабораторних робіт та 2 години прикінцевого семінару.

Самостійна робота студентів в обсязі 32 годин передбачає підготовку до навчальних занять і контрольних заходів та здійснюється у вільний від аудиторних навчальних занять час.

Контактні години передбачають індивідуальні консультації та он-лайн контроль студентів з використанням платформи Moodle.

Порядок проведення атестації

Модуль включає лекції, лабораторну роботу та практичні заняття.

Поточне оцінювання студента передбачає оцінювання викладачем виконання студентом лабораторних робіт, а також участі студента у обговоренні виконання лабораторних робіт на прикінцевому семінарі.

Підсумкове оцінювання модуля передбачає виконання модульної контрольної роботи.

Графік проведення оцінювання модуля М4.2

Номер тижня	Оцінювання
5	Поточна оцінка за виконання лабораторної роботи №1
6	Поточна оцінка за виконання лабораторної роботи №2
7	Поточна оцінка за виконання лабораторної роботи №3
8	Поточна оцінка за участь у обговоренні результатів виконання лабораторних робіт на прикінцевому семінарі
8	Підсумкова оцінка за модульну контрольну роботу

Метод оцінювання модуля

Модуль оцінюється за рейтинговою системою за 20 бальною шкалою. Рейтингова оцінка модуля є накопичувальною та включає всі поточні оцінки та підсумкову оцінку модульної контрольної роботи.

Вид завдання	Оцінка
Поточна оцінка за виконання лабораторної роботи №1	4
Поточна оцінка за виконання лабораторної роботи №2	4
Поточна оцінка за виконання лабораторної роботи №3	4
Поточна оцінка за участь у обговоренні результатів виконання лабораторних робіт на прикінцевому семінарі	4
Підсумкова оцінка модульної контрольної роботи	4

Максимальна сумарна поточна оцінка за всі практичні та лабораторну роботи в межах навчального модуля складає 16 балів.

Оцінювання виконання та захисту студентом лабораторної роботи виконується за критеріями, наведеними в Додатку 1.

Оцінювання участі студентів у обговоренні результатів виконання лабораторних робіт на прикінцевому семінарі виконується за критеріями, наведеними в Додатку 3.

Всі набрані студентом бали підсумовуються (максимально студент може отримати 20 балів), штрафні бали віднімаються (максимально мінус 4 бали).

Сумарна оцінка (від 0 до 20 балів) є індивідуальною рейтинговою оцінкою засвоєння студентом навчального модуля М4.2.

Оцінювання результатів тестування

Модульна контрольна робота проводиться наприкінці 8-го тижня навчання (останнього тижня модуля М4.2) у форматі тестування за допомогою платформи дистанційного навчання Moodle (або іншої). Тест складається з низки теоретичних питань, що охоплюють всі теми навчального модуля М4.2.

Модульна контрольна робота оцінюється у 4 бали.

Навчальна програма та матеріали модуля М4.2

Лекція 1. Створення ігор у середовищі WIM15.

Анотація

Лекція знайомить з інструментарієм візуальної розробки HTML5-ігор WIM15, його організацією та можливостями, інструментами розробки – редактором логіки й редактором сцен. Розглядаються також питання шаблонів ігор, їх безпосереднього застосування шляхом налаштування, або адаптації шаблонів для створення нових ігрових проектів.

Мета лекції

Ознайомити студентів з інструментарієм візуальної розробки ігор WIM15, забезпечити розуміння основних елементів середовища розробки та їх взаємозв'язків, принципів роботи в середовищі та можливостями його окремих інструментів.

Очікувані результати

Студент має:

- знати та вміти використовувати основні інструменти середовища розробки: редактор сцен, редактор логіки, dashboard, налаштування;
- знати та розуміти особливості застосування шаблонів для побудови ігор різних типів шляхом адаптації або налаштування;
- знати різні аспекти реалізації життєвого циклу в середовищі розробки та вміти використовувати їх під час розробки гри;
- знати та розуміти основні принципи створення ігор різного типу із застосуванням інструментів середовища WIM15.

Контрольні запитання

- З чого складається середовище WIM15?
- Що таке редактор логіки?
- Якими є можливості редактора логіки?
- Що таке редактор сцен?
- Якими є можливості редактора сцен?
- Що таке чорна скринька?
- Що таке сцена?
- Як налаштувати логіку гри?
- Що таке шаблон гри?
- В чому полягає налаштування шаблону гри?
- В чому полягає адаптація шаблону гри?
- Як створюється проект гри «з нуля»?
- Як розробити візуалізацію гри?

Лекція 2. Редактор сюжету та редактор логіки у WIM5.

Анотація

Лекція знайомить з двома основними інструментами візуальної розробки ігор. Розглядаються принципи використання цих інструментів, основні панелі та вкладки, елементи логіки гри та ігрових ресурсів, можливості налаштування ресурсів та логіки гри.

Мета лекції

Ознайомити студентів з інструментами візуальної розробки комп'ютерних ігор, пояснити особливості їх налаштування та використання. Розкрити особливості застосування елементів логіки гри та ігрових ресурсів в ігрових додатках, ознайомити студентів з технологією розробки гри за допомогою середовища візуальної розробки WIM5.

Очікувані результати

Студент має:

- знати візуальний інтерфейс редактора сюжету та розуміти принципи його використання;
- знати візуальний інтерфейс редактора логіки та розуміти принципи його використання;
- знати доступні елементи логіки гри та вміти коректно їх застосовувати при розробці гри;
- знати доступні ігрові ресурси та особливості їх використання в сценарії гри, вміти правильно обирати інструменти розробки у відповідності до поставлених вимог.

Контрольні запитання

- Що таке ігровий ресурс?
- Що таке спрайт?
- Як організовані ігрові ресурси на робочому столі?
- З яких елементів складається логіка гри?
- Для чого призначений менеджер ресурсів?
- Що таке сцена?
- Як організовані переходи між сценами?
- Що таке менеджер ресурсів?
- Якими є елементи логічної схеми?
- Що таке компоненти з'єднання?
- Які панелі має логічна схема?

- Якими є основні типи чорних скриньок?
- Що таке логічне дерево?
- Для чого призначений менеджер активів?

Лекція 3. Візуальне програмування у WIMI5.

Анотація

Лекція знайомить з особливостями візуального програмування у середовищі WIMI5. Розглядається внутрішня структура та особливості програмування основного елементу логіки гри – чорної скриньки, реалізація потоку логіки гри. Роз’яснюються особливості чорної скриньки CodeRunner та використання JavaScript коду для програмування подій та логіки гри.

Мета лекції

Ознайомити студентів з принципами та методами візуального програмування за допомогою JavaScript при розробці ігор в середовищі WIMI5 для реалізації потоку логіки гри.

Очікувані результати

Студент має:

- знати та розуміти основні поняття програмної моделі гри: активатори, тригери, властивості, параметри, колекції, конектори, посилання;
- розуміти особливості використання елементів BlackBox та CodeRunner та вміти їх використовувати в ігровій моделі;
- знати та розуміти принципи та правила реалізації потоку логіки гри, вміти їх використовувати в ігрових проектах;
- вміти інтегрувати в проект JavaScript код для програмування подій та логіки гри.

Контрольні запитання

- Що таке конектор і як він використовується?
- Що таке тригер та в яких випадках він активується?
- Що таке властивість та в яких об’єктах вона присутня?
- Що таке параметр та які його види бувають?
- Які види вихідних конекторів Ви знаєте?
- Що таке колекція параметрів та як отримати доступ до її елементів?
- Що таке посилання та як воно реалізується?
- Які елементи реалізуються при завантаженні гри? Рівня?
- Що таке потік логіки гри?
- Які є типи чорних скриньок?
- Як реалізуються посилання на властивість параметра?
- Що можна налаштувати в елементі CodeRunner?

- Як інтегрувати JavaScript код у ігровий додаток?

Лекція 4. Тестування, розгортання та публікація комп'ютерної гри.

Анотація

Лекція знайомить з особливостями реалізації віртуальних товарів та управління ними, тестування й налагодження ігрового додатку, його розгортання та публікації засобами середовища WIM15.

Мета лекції

Ознайомити студентів з інструментами тестування й налагодження ігрового додатку, розгортання та публікації в середовищі WIM15.

Очікувані результати

Студент має:

- знати основні принципи монетизації комп'ютерних ігор та вміти їх застосовувати;
- знати особливості тестування й налагодження ігрових додатків в середовищі WIM15 та вміти їх застосовувати в ігрових проектах;
- знати особливості розгортання та публікації ігрових додатків в середовищі WIM15 та вміти їх застосовувати в ігрових проектах;
- вміти використовувати інструментарій середовища WIM15 для розробки та розповсюдження ігрових додатків.

Контрольні запитання

- Що таке віртуальні товари?
- Як створити віртуальні товари?
- Як налаштувати конфігурацію віртуальних товарів?
- Що таке віртуальна чорна скринька?
- Як організовано віртуальний магазин?
- За допомогою якого інструменту публікують ігрові проекти?
- Як протестувати ігровий проект?
- Як розгорнути ігровий додаток?
- Як публікують ігровий додаток?
- Як використовувати сайт WIMI Labs?
- Як створити завантажуваний файл проекту?
- Як опублікувати гру HTML5 в інтернет-магазині Google Chrome?

Лабораторна робота №1. Візуальна розробка елементів ігрового середовища.

Анотація

Лабораторна робота орієнтована на оволодіння студентами навичок візуальної розробки як низькорівневих елементів ігрового середовища (зображення, спрайти, звуки), так і високорівневих елементів (табло з підрахунком балів, ігрова сесія).

Лабораторна робота розрахована на 6 академічних годин і виконується малими робочими групами студентів.

Мета лабораторної роботи

Засвоїти навички і прийоми візуальної розробки елементів ігрового середовища засобами інструментарію WIMI5.

Очікувані результати

Студент має отримати:

- знання інструментів середовища візуальної розробки ігор WIMI5 та розуміння особливостей їх застосування для розробки елементів ігрового середовища;
- вміння орієнтуватися у інтерфейсі середовища візуальної розробки комп'ютерних ігор WIMI5 та навички застосування його інструментів для розробки елементів ігрового середовища.

Контрольні запитання

- Поясніть, які інструменти і як саме можуть бути використані для візуальної розробки елементів ігрового середовища.
- Поясніть функціональні можливості середовища WIMI5 щодо реалізації графічних, анімаційних та звукових ігрових об'єктів.
- Охарактеризуйте можливості роботи з ігровими об'єктами та ресурсами.
- Поясніть способи розв'язання задач взаємозв'язку ігрових об'єктів у WIMI5.

Лабораторна робота №2. Візуальна розробка елементів ігрової моделі.

Анотація

Лабораторна робота орієнтована на оволодіння студентами навичок візуальної розробки складних елементів ігрової моделі (руху та зіткнень ігрових об'єктів).

Лабораторна робота розрахована на 6 академічних годин і виконується малими робочими групами студентів.

Мета лабораторної роботи

Засвоїти навички і прийоми візуальної розробки елементів ігрового моделі засобами інструментарію WIMI5.

Очікувані результати

Студент має отримати:

- знання інструментів середовища візуальної розробки ігор WIMI5 та розуміння особливостей їх застосування для розробки складних елементів і сценаріїв ігрової моделі;
- вміння орієнтуватися у інтерфейсі середовища візуальної розробки комп'ютерних ігор WIMI5 та навички застосування його інструментів для розробки складних елементів і сценаріїв ігрової моделі.

Контрольні запитання

- Поясніть, які інструменти і як саме можуть бути використані для візуальної розробки елементів ігрової моделі.
- Поясніть функціональні можливості середовища WIMI5 щодо реалізації сценаріїв взаємодії ігрових об'єктів.
- Охарактеризуйте можливості реалізації руху та зіткнень ігрових об'єктів.
- Поясніть способи розв'язання задач побудови потоку логіки гри засобами WIMI5.

Лабораторна робота №3. Монетизація та публікація ігрового проекту.

Анотація

Лабораторна робота орієнтована на оволодіння студентами навичок монетизації та публікації ігрових додатків.

Лабораторна робота розрахована на 6 академічних годин і виконується малими робочими групами студентів.

Мета лабораторної роботи

Засвоїти навички і прийоми монетизації, розгортання і публікації ігрових додатків засобами WIMI5.

Очікувані результати

Студент має отримати:

- знання інструментів монетизації, розгортання та публікації середовища візуальної розробки ігор WIMI5 та розуміння особливостей їх практичного застосування для розроблених ігрових додатків;
- вміння орієнтуватися у інтерфейсі середовища візуальної розробки комп'ютерних ігор WIMI5 та навички застосування його інструментів монетизації, розгортання та публікації ігрових додатків.

Контрольні запитання

- Поясніть, які інструменти монетизації і як саме можуть бути використані для ігрових додатків.
- Поясніть, які інструменти розгортання і як саме можуть бути використані для ігрових додатків.
- Поясніть, які інструменти публікації і як саме можуть бути використані для ігрових додатків.
- Поясніть функціональні можливості середовища WIMI5 щодо монетизації, розгортання та публікації комп'ютерних ігор.
- Охарактеризуйте можливості розповсюдження розроблених ігрових додатків за допомогою WIMI5.

Прикінцевий семінар. Презентація та обговорення виконаних лабораторних робіт.

Анотація

Прикінцевий семінар присвячено презентації студентами результатів виконання завдань лабораторних робіт №1-№3 та їх обговоренню у відкритій загальній дискусії. Наприкінці семінару студенти оцінюють друг друга.

Мета семінару

Взаємна оцінка студентами отриманих ними в результаті виконання лабораторних робіт вмінь і навичок розробки ігрових додатків з використанням середовища візуальної розробки WIMI5.

Очікувані результати

Студент має отримати:

- вміння критично ставитися до результатів особистої роботи;
- вміння проводити самооцінку виконаних робіт;
- вміння обґрунтувати певний спосіб діяльності при виконанні лабораторних робіт;
- вміння презентувати та аргументувати свої результати, обговорювати їх перед аудиторією;

- вміння самостійно формулювати та доносити до співрозмовника свої думки, ясно і конкретно їх висловлювати, слухати та розуміти співрозмовника, використовувати прийоми ділового спілкування;
- вміння дотримуватись етичних норм поведінки, ділового та партнерського спілкування;
- навички формулювання зауважень щодо функціональних та нефункціональних обмежень середовища розробки.

Список джерел інформації по модулю М4.2.

1. Shell, J. The Art of Game Design: A book of Lenses [Text] / J. Shell. – A K Peters/CRC Press; 2 edition, 2014. – 600 p.
2. Fullerton, T. Game Design Workshop: A Playcentric Approach to Creating Innovate Games [Text] / T. Fullerton. – A K Peters/CRC Press; 3 edition, 2014. – 535 p.
3. Novak D. Game Development Essentials: An Introduction. – 2004.
4. <http://wimi5.com/learn/video-tutorials/>
5. <http://wimi5.com/learn/how-to/>
6. <http://developer.wimi5.com/doc/latest/>

11.МОДУЛЬ М4.3. ВИКОНАННЯ ГРУПОВОГО ФІНАЛЬНОГО ПРОЕКТУ

Опис модуля

Галузь знань:	12 «Інформаційні технології».
Спеціальність:	121 «Інженерія програмного забезпечення»
Рівень:	бакалавр.
Назва дисципліни:	Основи розробки комп'ютерних ігор
Назва модуля:	Виконання групового фінального проекту
Семестр:	4
Кредитних одиниць:	1
Академічних годин:	30, з них семінар – 2 год., самостійної роботи – 28 год.
Викладач:	д.т.н., професор Шерстюк В.Г.

Вступ

Модуль «Виконання групового фінального проекту» є заключним.

Модуль включає виконання студентами, працюючими в малих групах, групового фінального проекту, що передбачає розробку завершеного прототипу комп'ютерної гри, використовуючи ігрові концепції та моделі, створені під час виконання практичних завдань першого навчального модуля (М4.1), та застосовуючи інструментарій розробки, навички використання якого отримуються студентами під час виконання лабораторних занять другого навчального модуля (М4.2).

Мета модуля

Метою модуля є формування у студентів навичок командної розробки крос-платформних ігрових додатків із застосуванням інструментарію візуальних середовищ розробки, розвиток творчих здібностей, потенціалу креативного і логічного мислення студентів.

Результати навчання

Знання та їх використання

У випадку успішного оволодіння матеріалами даного модуля студент буде володіти основами проектування, розробки, підтримки та просування комп'ютерних ігор; особливості реалізації циклу розробки та технології розробки комп'ютерної гри; обов'язки та відповідальність учасників процесу розробки гри, їх ролі та необхідні професійні компетенції.

Дослідницькі навички

У разі успішного вивчення модуля студент отримає навички реалізації креативних концепцій гри, працюючи у команді та використовуючи сучасний інструментарій візуальної розробки комп'ютерної гри.

Спеціальні вміння

У разі успішного вивчення модуля студент буде вміти проектувати комп'ютерні ігри різних жанрів; реалізовувати базові елементи циклу розробки типової комп'ютерної гри; використовувати сучасний інструментарій для візуальної розробки ігрових додатків; оволодіє навичками командної розробки простих крос-платформних ігрових додатків.

Соціальні вміння

У разі успішного вивчення модуля студент буде вміти працювати в складі професійної команди розробників, усвідомлюючи свої персональні обов'язки та відповідальність за результати роботи.

Особисті якості

У разі успішного вивчення модуля студент розвине такі особисті якості:

- креативний спосіб мислення під час практичної реалізації концепції та проекту комп'ютерної гри;
- самостійно вирішувати питання, що стосуються розробки комп'ютерних ігор;
- приймати обґрунтовані технічні рішення під час розробки;
- систематизувати професійні знання; аналізувати, цілеспрямовано шукати і вибирати необхідні для розробки гри інформаційні ресурси з урахуванням сучасних досягнень науки і техніки.

Перелік компетентностей та результати навчання

Загальні (універсальні) компетентності	<i>ЗК-4, ЗК-5, ЗК-6</i>
Спеціальні (фахові) компетентності	<i>ФК-1, ФК-2, ФК-3, ФК-5, ФК-6</i>
Програмні результати навчання	<i>ПРН-4, ПРН-7, ПРН-8, ПРН-10</i>

Календарний план семестру і структура модуля М4.3.

Місце модуля М4.3 в структурі дисципліни

Номер	Модулі дисципліни	Тиждень навчання
1	М4.1 Проектування комп'ютерних ігор	1-4
2	М4.2 Візуальна розробка комп'ютерних ігор	5-8
3	М4.3 Виконання групового проекту	9-16

Інформаційне наповнення модуля М4.3

Номер тижня	Зміст
9-14	Виконання фінального групового проекту
15	Подання звіту про виконання фінального групового проекту
16	Презентація та обговорення фінального групового проекту на заключному семінарі

Порядок слідування занять по тижням

Тиждень навчання	
9	Виконання групового проекту
10	Виконання групового проекту
11	Виконання групового проекту
12	Виконання групового проекту
13	Виконання групового проекту
14	Виконання групового проекту
15	Виконання та подання групового проекту
16	Семінар Презентація результатів виконання фінального групового проекту

Форми навчання

Навчальний процес полягає у виконанні студентами, які працюють в малих групах, групового фінального проекту, своєчасному поданні звіту про виконання проекту з використанням платформи Moodle (або іншої), презентації та обговорення результатів виконання проекту.

Виконання проекту планується як самостійна робота студента у обсязі 28 годин.

Аудиторна робота включає 2 години заключного семінару, на якому студенти мають презентувати та обговорювати результати виконання групового фінального проекту.

Контактні години передбачають індивідуальні консультації та он-лайн контроль студентів з використанням платформи Moodle.

Порядок проведення атестації

Модуль передбачає лише підсумкове оцінювання за результатами виконання групового фінального проекту, що проводиться під час заключного

семінару на 16-му тижні навчання, де студенти мають презентувати та обговорити у студентському середовищі результати виконання проекту.

Графік проведення оцінювання модуля М4.3

Номер тижня	Оцінювання
15	Подання групового фінального проекту
16	Оцінювання групового фінального проекту

Метод оцінювання модуля

Модуль оцінюється за рейтинговою системою за 20 бальною шкалою.

Вид завдання	Оцінка
Оцінка групового фінального проекту	20

Оцінювання результатів виконання, презентації та обговорення студентами групового фінального проекту виконується за критеріями, наведеними в Додатку 3.

За кожний тиждень прострочення подання фінального групового проекту знімається 2 бали (максимально 10 балів).

Всі набрані студентом бали підсумовуються (максимально студент може отримати 20 балів), штрафні бали віднімаються (максимально мінус 10 балів).

Сумарна оцінка (від 0 до 20 балів) є індивідуальною рейтинговою оцінкою засвоєння студентом навчального модуля М4.3.

Навчальна програма та матеріали модуля М4.3

1. Виконання групового фінального проекту

На першому тижні навчальних занять за дисципліною відбувається рандомне формування малих груп студентів (чисельністю по 4-5 осіб), кожна з яких буде виконувати фінальний груповий проект.

Впродовж навчального модуля М4.1. групи студентів виконують завдання на практичних заняттях, що дозволяє їм визначитися з проектом комп'ютерної гри, створюючи ігрову концепцію й ігрову модель, а потім моделюючи ігрове середовище.

Впродовж навчального модуля М4.2. групи студентів виконують лабораторні роботи, засвоюючи певні навички розробки комп'ютерної гри за допомогою візуального середовища WIMI5.

Потім впродовж навчального модуля М4.3. студенти, використовуючи отримані на попередньому модулі навички, реалізують прототип комп'ютерної гри за ігровою концепцією та моделлю, отриманими в результаті виконання першого модуля.

Подання групового фінального проекту

Звіт за результатами виконання проекту, виконаний у відповідності до методичних рекомендацій, надається викладачеві через платформу Moodle (або іншу використовувану університетом) не пізніше передостаннього тижня навчання (п'ятнадцятого).

Продовження терміну подання проекту можливе лише при наявності поважної причини, передбаченої Положенням про організацію навчального процесу в університеті.

Презентація та обговорення групового фінального проекту

На останньому, шістнадцятому тижні групи студентів на заключному семінарі презентують звіт за результатами проекту, обговорюють їх та отримують підсумкову оцінку. Для цього кожна група студентів готує електронну презентацію у відповідності до методичних рекомендацій.

2. Заключний семінар

Анотація

Заключний семінар присвячено презентації студентськими командами та обговоренню виконаного групового фінального проекту, що має реалізувати завершений прототип ігрового додатку, виконаний за допомогою середовища візуальної розробки WIM15, у відповідності до ігрової концепції та моделі, отриманими в результаті виконання першого модуля.

Мета семінару

Оцінити прийняті малими групами ігрові концепції й моделі та коректність їх реалізації впродовж виконання проекту, а також працездатність ігрового додатку, його дизайн та юзабіліті.

Очікувані результати

Студент має отримати:

- вміння критично ставитися до результатів особистої роботи;
- вміння проводити самооцінку виконаних робіт;
- вміння обґрунтувати певний спосіб діяльності при виконанні практичних завдань;
- вміння презентувати та аргументувати свої рішення, обговорювати в групі результати роботи;
- вміння самостійно формулювати та доносити до співрозмовника свої думки, ясно і конкретно їх висловлювати, слухати та розуміти співрозмовника, використовувати прийоми ділового спілкування;
- вміння дотримуватись етичних норм поведінки, ділового та партнерського спілкування;

- навички оцінки ігрових додатків на підставі виявлених недоліків та помилок, що були допущені на етапах розробки ігрового додатку;
- навички формулювання зауважень щодо функціональних та нефункціональних вимог до гри;
- навички визначати шляхи подальшого розвитку і вдосконалення ігрового додатку з урахуванням результатів обговорення.

ЛЕКЦІЯ 1

СТВОРЕННЯ ІГОР У СЕРЕДОВИЩІ WiMi5

Онлайн-редактор дозволяє створювати, публікувати та монетизувати ігри HTML5.

За допомогою інструмента створення та перетягування WiMi5 можливо легко створювати крос-платформні HTML5-ігри.

Ви зможете публікувати свої ігри на багатьох ринках та швидко залучити гравців для ігор. Ви також можете почати монетизувати свою гру та отримувати доходи за допомогою кількох платіжних шлюзів, включених у платформу WiMi5.

Ця лекція покаже, як використовувати платформу WiMi5.

Спочатку загальний огляд WiMi5.

ОГЛЯД ПЛАТФОРМИ WiMi5.

СИСТЕМНІ ВИМОГИ

Як платформа на основі HTML5 потрібен будь-який сучасний веб-браузер для використання WiMi5.

Рекомендується використовувати Chrome, але будь-який інший веб-переглядач, що підтримує HTML5, теж працюватиме.

ЗАГАЛЬНИЙ ОГЛЯД

WiMi5 є єдиною схемою, яка дозволяє розробникам відеоігор **створювати, публікувати та монетизувати ігри HTML5** одним інструментом, використовуючи лише одну платформу. Ці 3 процеси інтегровані в один і той же інструментарій для розробників (рис. 1).

Після запуску WiMi5 Ви побачите Інформаційну панель, де зможете керувати своїми проектами. На панелі інструментів ви можете редагувати та створювати нові проекти або публікувати їх.

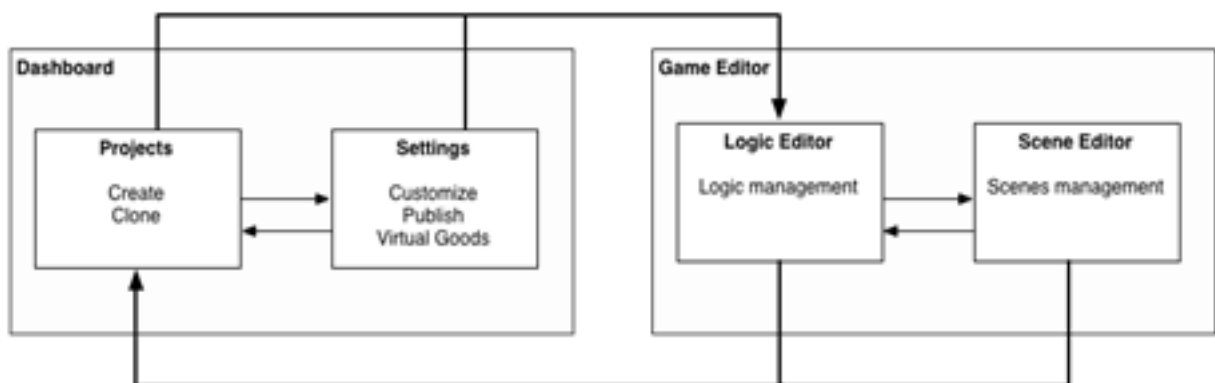


Рисунок 1. Процеси, що інтегровані до інструментарію

Якщо Ви редагуєте існуючий проект або створюєте новий, то будете перенаправлені до інструмента створення. За допомогою цього потужного та інтуїтивно зрозумілого інструменту ви можете працювати в двох різних інструментах: Редакторі сюжетів та Редакторі логіки.

Якщо вирішите опублікувати свій проект, Ви зможете зробити його публічним або приватним (наприклад, для тестування).

Наразі Ви можете лише опублікувати свій проект у зоні лабораторії WiMi5.

Незабаром Ви зможете опублікувати свої ігри HTML5 в **ігровій арені WiMi5**, у Веб-магазині Chrome, Firefox Marketplace та інших.

DASHBOARD

В даний час ви можете створити нову гру, редагувати існуючі або клонувати проект із сховища шаблонів ігор (DashBoard) (рис. 2).

У кожній грі є набір кнопок у нижній частині, щоб керувати своїми налаштуваннями, перейти до редактора, поділитися ним у соціальних мережах або видалити його.

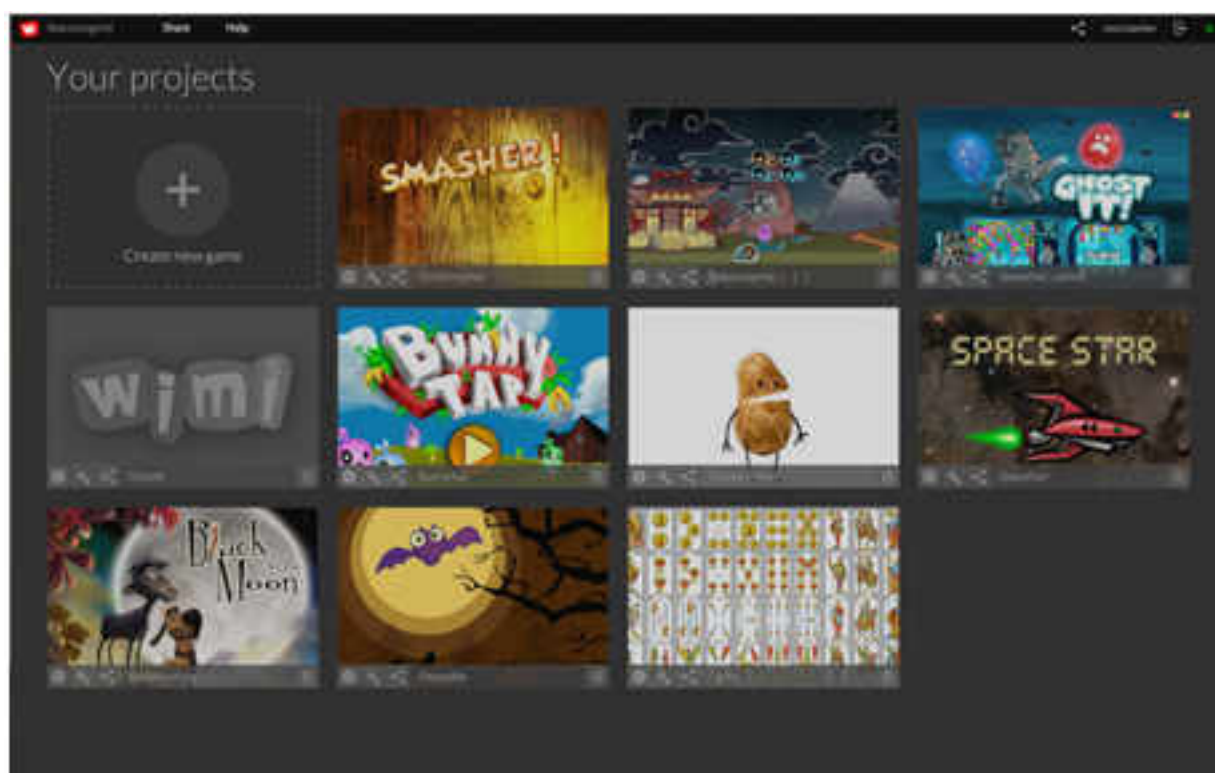


Рисунок 2. Сховище шаблонів ігор

Ви також можете двічі натиснути на нього, оскільки це ярлик, щоб перейти до перегляду параметрів.

У налаштуваннях Ви можете опублікувати свою гру, керувати віртуальними товарами, контролювати статистику тощо (рис. 3).



Рисунок 3. Налаштування шаблонів ігор

ОГЛЯД РЕДАКТОРА ЛОГІКИ

Коли Ви відкриваєте або створюєте новий проект з інформаційної панелі, Ви будете заходити на робочий стіл Logic Editor.

З будь-якого настільного ПК ви можете створити логіку своєї гри.

Це ядро ваших ігор HTML5, де ви зможете налаштувати всі дії та поведінку для ігрових елементів вашої.

У редакторі логіки ви знайдете Blackboxes, або чорні скриньки, які є основною функцією в WiMi5. Ви можете використовувати чорні скриньки для створення візуальних сценаріїв у грі.

Чорна скринька - це скринька, яка містить елемент логіки, який можна підключити до інших чорних скриньок, що формують складну поведінку.

На вкладці «Чорні скриньки» ви знайдете всі чорні скриньки, розташовані в декількох категоріях. Наприклад, ви знайдете чорну скриньку під назвою *Перемістити* для анімації елементів в категорії *анімації*.

Якщо ви хочете використовувати цю чорну скриньку, потрібно лише:

- перетягнути її до логічної діаграми;
- заповнити необхідні вхідні та вихідні параметри;
- підключити її активатори;
- активувати й налаштувати її властивості.

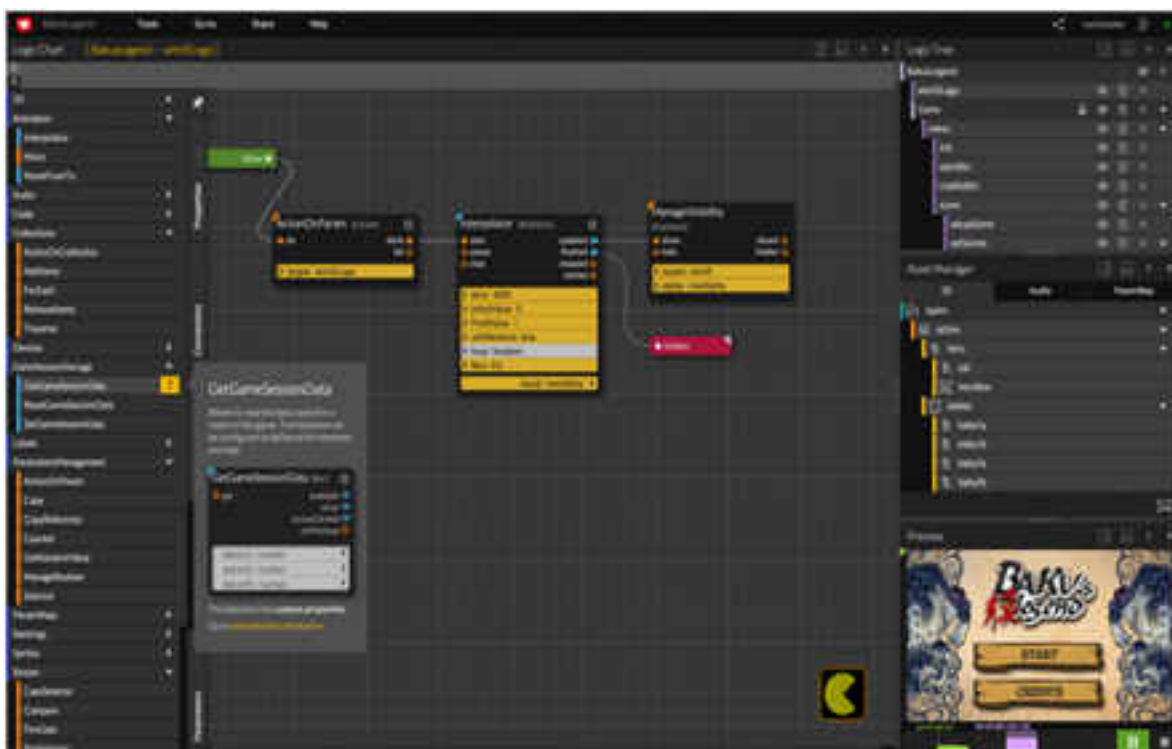


Рисунок 4. Робочий стіл редактора логіки

Всі елементи в редакторі можна перетягувати, так що ви також можете перетягнути активи з *Asset Manager* і помістити їх у параметри Blackboxes, або ви також можете перетягнути параметри з вкладки *Parameters*.

Поки що мова йшла про підключення чорних скриньок та зміну значень властивостей, тому програмування нам було не потрібно.

WiMi5 має потужний візуальний сценарій, який дозволяє створювати всю логіку гри без написання будь-якого коду.

У будь-якому випадку, необхідно знати принаймні деякі основні концепції програмування, щоб відчувати себе комфортно з інструментом розробки WiMi5.

ОГЛЯД РЕДАКТОРА СЦЕНАРІЇВ

Ви можете отримати доступ до редактора сценаріїв, натиснувши кнопку, розташовану в нижньому правому куті панелі керування ресурсами на робочому столі "Logic Editor".

У цьому перегляді ви зможете керувати всіма функціями, пов'язаними з візуальним створенням декількох сцен вашої гри (рис. 5).

Ви можете розбити свою гру на сцени, щоб організувати свій процес створення.

У режимі перегляду сценаріїв ви можете завантажувати та керувати своїми зображеннями та звуковими ресурсами, які ви будете використовувати у своїх HTML5-іграх.



Рисунок 5. Робочий стіл редактора сценаріїв

УПРАВЛІННЯ МАКЕТОМ РОБОЧОЇ ОБЛАСТІ.

WiMi5 – це дуже гнучкий інструмент, який можна налаштувати відповідно до потреб.

Ви можете переміщати, змінювати розмір, видаляти або створювати панелі, щоб створити робоче середовище, яке вам сподобається.


Просто використовуйте кнопки, розташовані у верхньому правому куті кожної панелі.

Ви також можете перетягнути їх і обміняти позиції між двома панелями (рис. 6).



Рисунок 6. Обмін між панелями

ОБМІН ПАНЕЛЯМИ РЕДАКТОРА СЦЕН ТА РЕДАКТОРА ЛОГІКИ

Коли ви перебуваєте в *Logic Editor*, можете побачити кнопку в правому нижньому кутку *Диспетчера Asset* панелі () , що дозволяє перейти до редактору сцен.

Редактор сцени є робочою областю, в якій ви можете налаштувати кожну сцену вашої гри. Ви можете повернутися до *редактора логіки*, написнувши кнопку, розташовану у верхньому лівому куті.

АДАПТАЦІЯ ШАБЛОНІВ WiMi5 ДЛЯ СТВОРЕННЯ ВЛАСНИХ ІГОР

WiMi5 пропонує кілька **ігрових шаблонів** та інших інструментів, з яких можна запусити новий **проект HTML5**. Ми подивимся, як ви можете **змінити графіку** одного з цих шаблонів, щоб персоналізувати гру з власною графікою і, таким чином, мати свою власну гру.

Для цього ми будемо змінювати шаблон до гри **WiMi Bird**, який є клоном знаменитої гри Flappy Bird (рис. 7).

Ми збираємося використовувати деякі зображення, які мають однаковий розмір і характеристики оригіналів, що є найпростішим способом змінити шаблон гри.

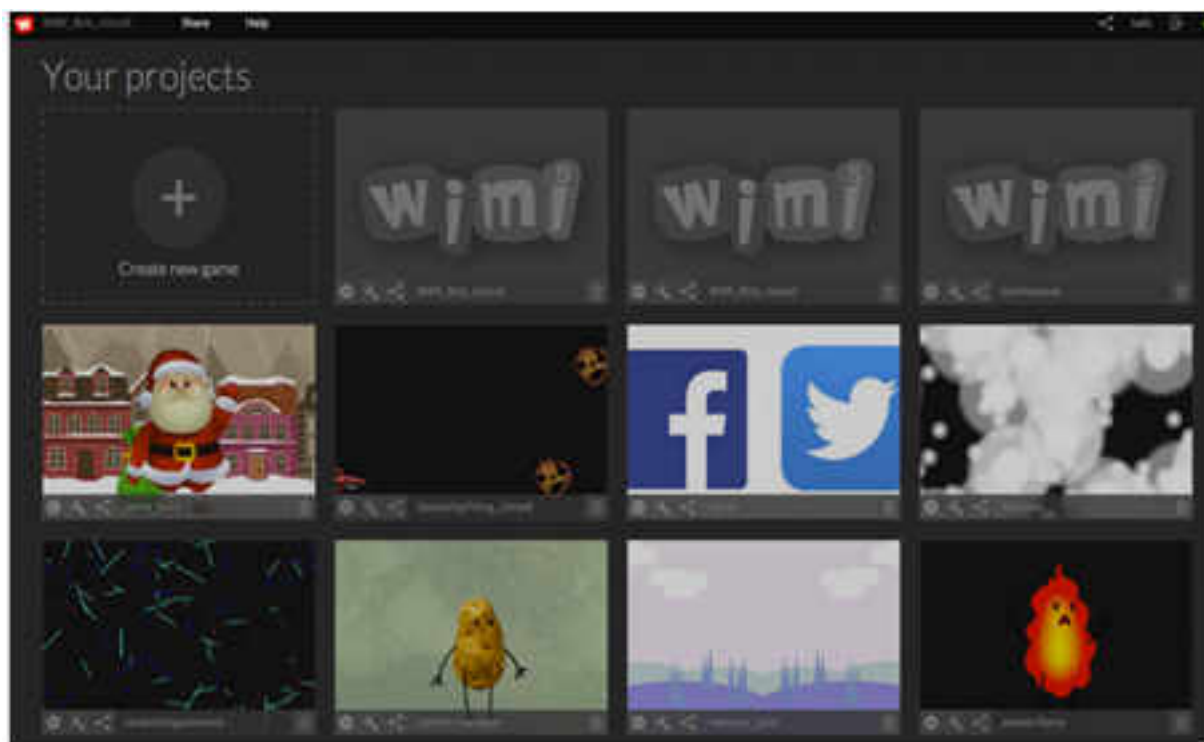


Рисунок 7. Шаблон гри Flappy Bird

Першим кроком є створення нового проекту на основі цього шаблону.

Для цього ввійдіть в WiMi5 і перейдіть прямо на Інформаційну панель. Тут ви побачите всі свої проекти та кнопку «Створити нову гру».

Коли ви натискаєте цю кнопку, ви побачите різні варіанти, в основному, створення нової гри з нуля або використання існуючого шаблону.

Виберіть шаблон "WiMi5 Bird", який знаходиться в кінці списку (рис. 8).

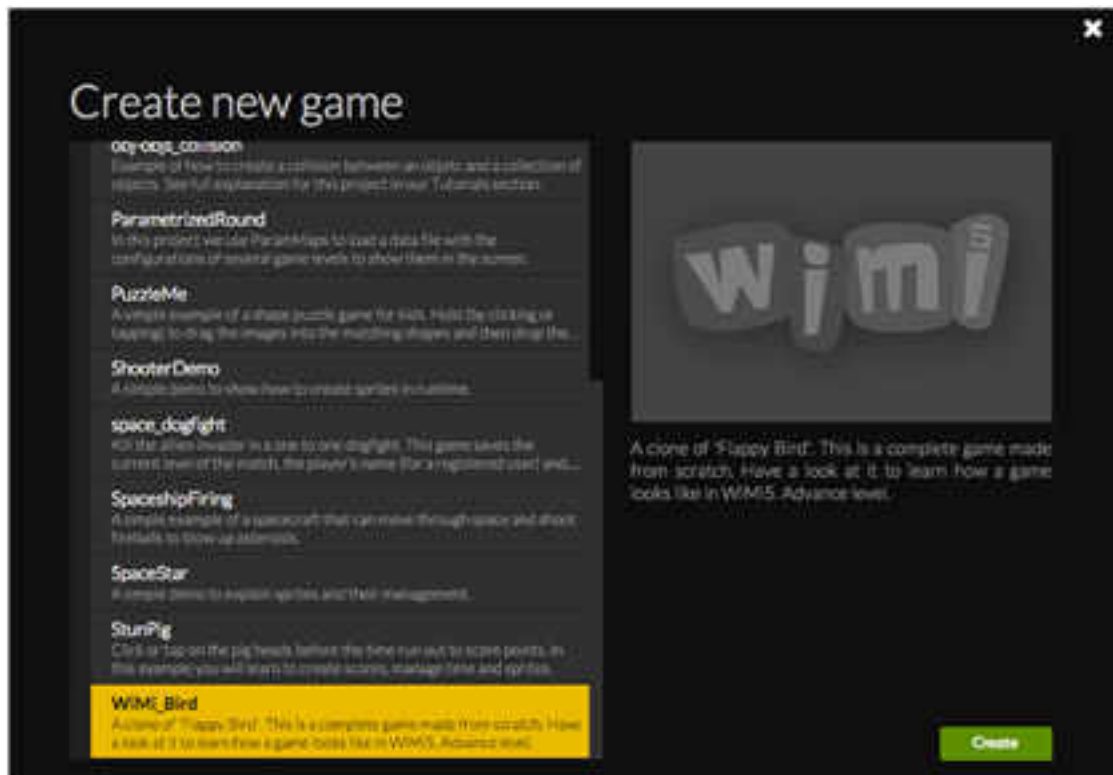


Рисунок 8. Вибір шаблону гри WiMi5 Bird

Коли ви створюєте проект, вам буде запропоновано налаштування проекту, де ви зможете налаштувати декілька його характеристик, а також керувати віртуальними товарами, статистикою або магазинами, де можна буде опублікувати проект.

Прямуємо до редактора, щоб змінити гру.

Коли ви отримуєте доступ до проекту, спочатку ви побачите вікно **редактора логіки**, де ви можете перевірити логіку, створену для цієї гри, за допомогою візуального сценарію (рис. 9).

Натискання на "Редактор сюжетів" з меню "Робочий стіл" приведе вас до **редактора сцени**, де ви зможете змінити графіку гри (рис. 10).

Але перш ніж почати змінювати зображення, вам потрібно знати характеристики оригіналів, особливо їх розмір. Щоб дізнатись, потрібно використовувати **інструмент для налагодження**, який постачається в більшості сучасних веб-переглядачів. У цьому випадку ми використовуємо Chrome.

Першим кроком є запуск оригінальної гри в браузері. Для цього потрібно **розгорнути** проект та провести **тест**, щоб знайти загальнодоступну URL-адресу, доступ до якої можна отримати за допомогою вашого веб-переглядача.

Тому поверніться до логічного редактора та в меню "Інструменти", виберіть "Розгорнути", а потім "Тестувати".

Тепер у вас є URL, де ви можете побачити оригінальну гру (рис. 11).

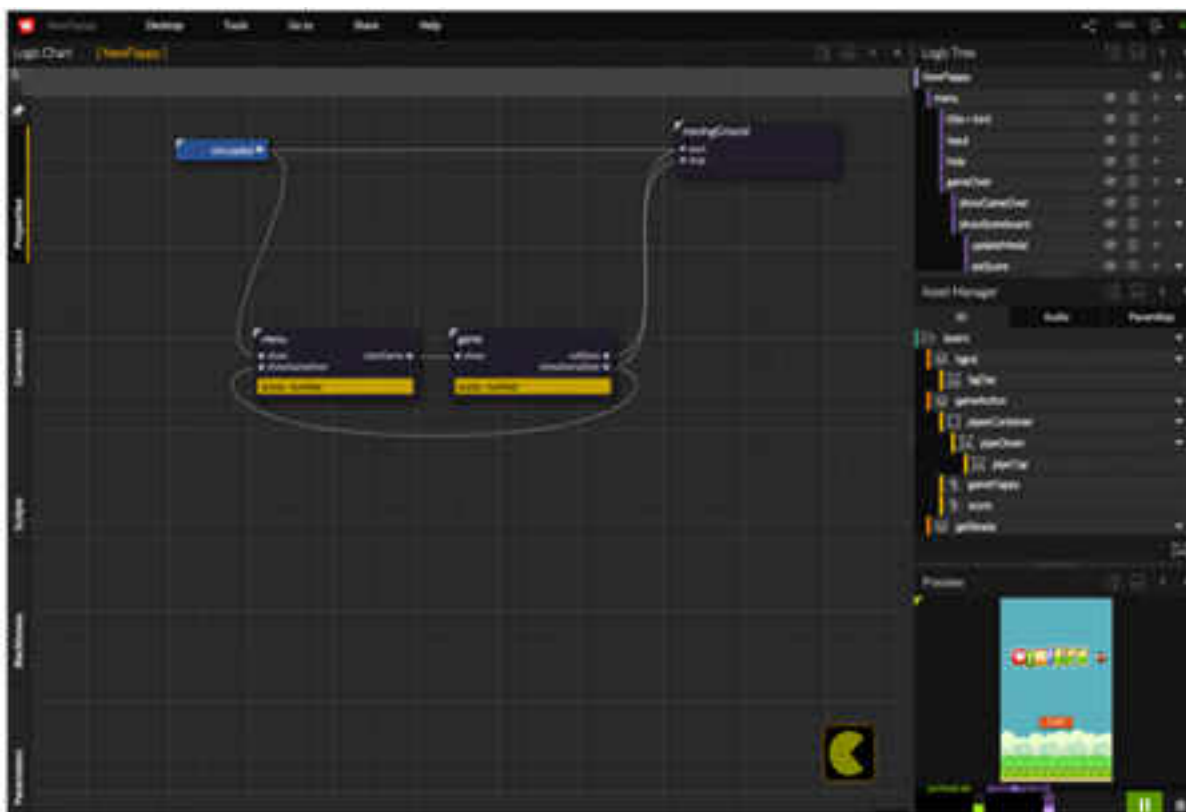


Рисунок 9. Редагування логіки шаблону гри

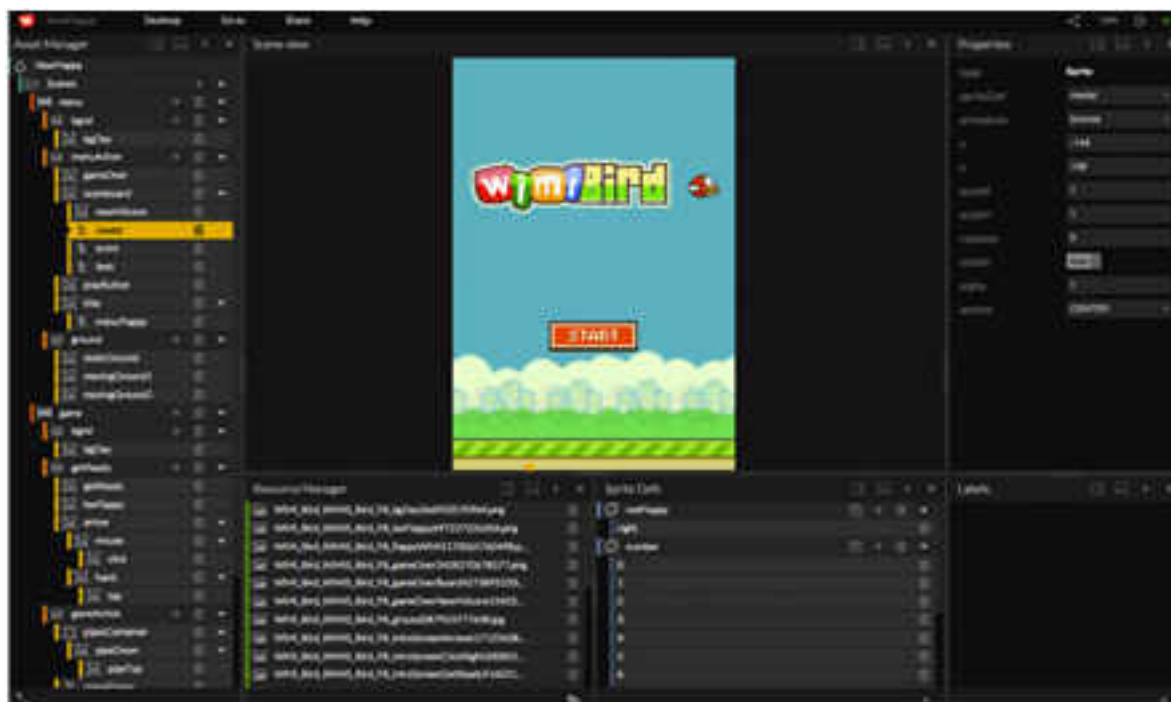


Рисунок 10. Редагування сюжету шаблону гри

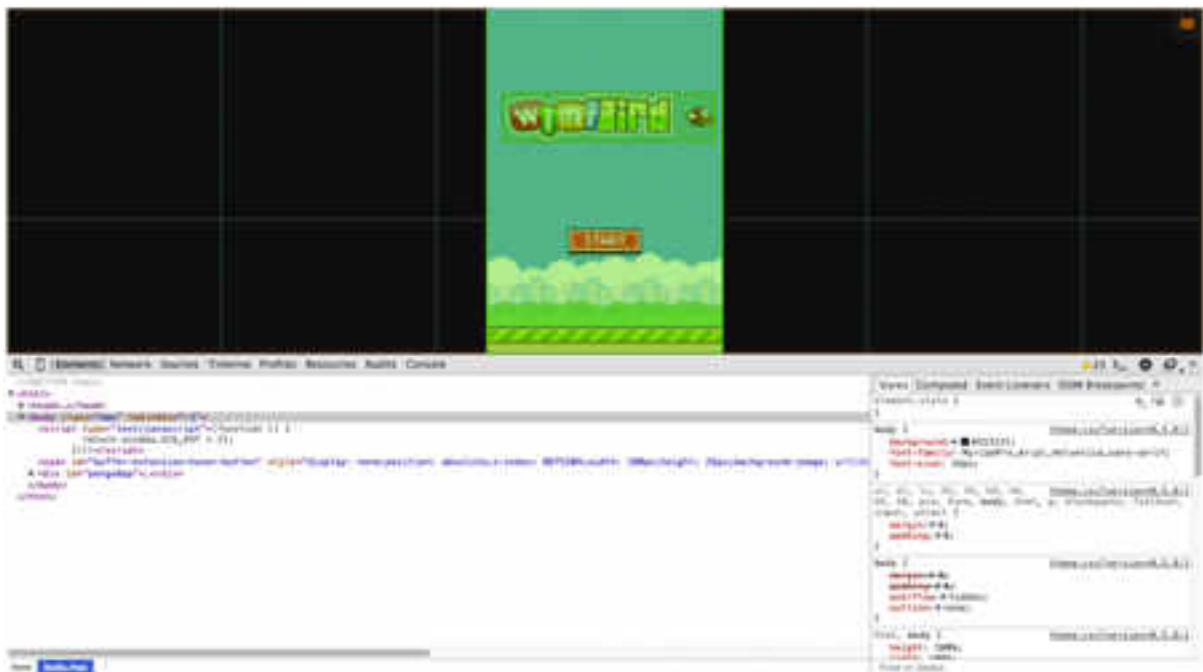


Рисунок 11. Тестування гри

Щоб переглянути графіку, яка використовується в грі, відкрийте **інструменти розробки** в "Перегляд> Розробник> Засоби розробки" в Chrome for Mac або "Меню> Додаткові інструменти> Інструменти для розробників" у Chrome для Windows.

Таким чином, ви можете бачити вихідний код гри, а також інші параметри (рис. 12).

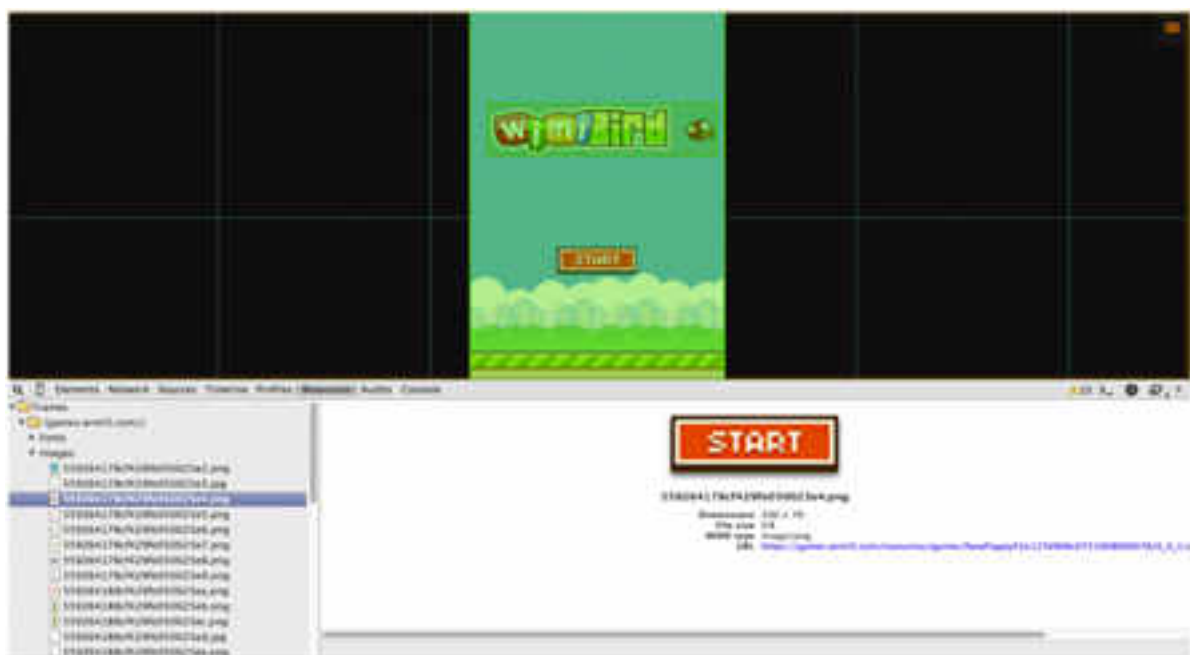


Рисунок 12. Перегляд параметрів гри



Рисунок 14. Перегляд подробиць зображень

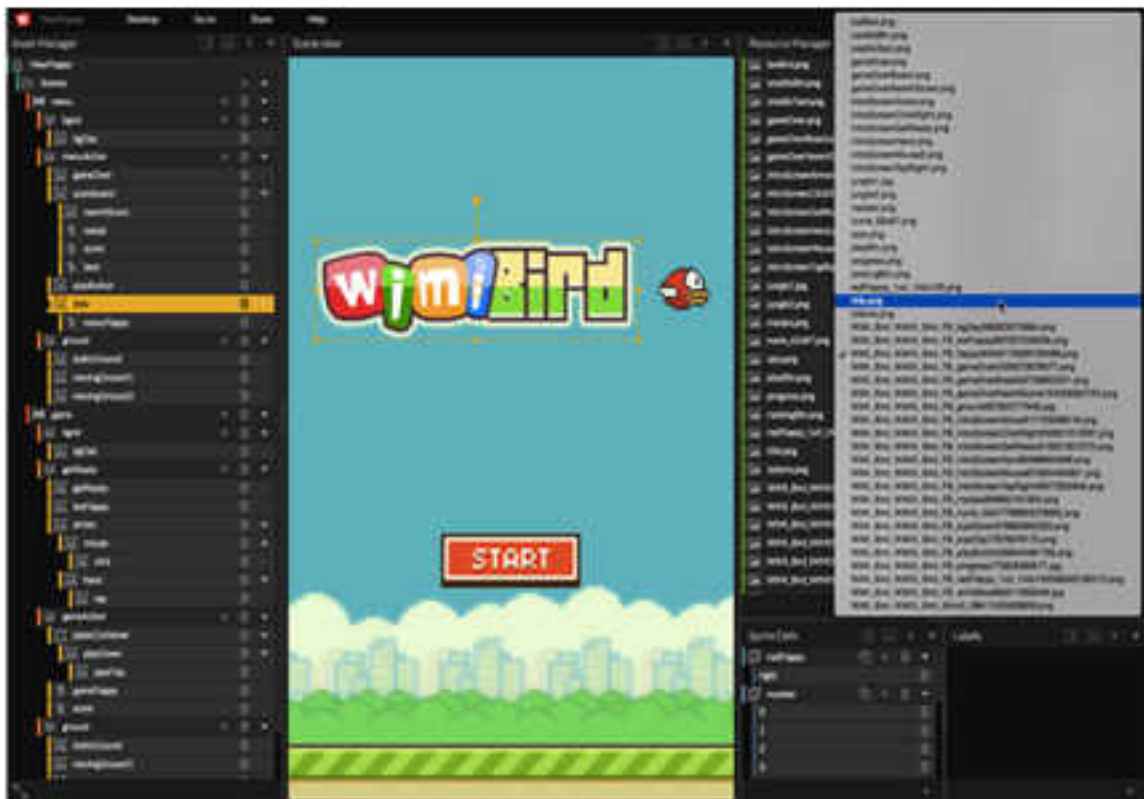


Рисунок 15. Створення пакету зображень

У **редакторі сюжетів** все, що вам потрібно зробити, це просто замінити старі зображення на нові. Отже, **виберіть** будь-яке зображення, яке вам подобається, наприклад, назву гри.

На панелі "Властивості" є властивість "зображення", яка визначає, які зображення використовуються для цього ресурсу.

Якщо клацнути на **спадному меню** праворуч, ви можете **вибрати** зображення, яке ви хочете використовувати, і воно автоматично з'явиться в «Scene View» (рис. 16).

Якщо ви "розгорнете" та "перевірите" в **логічному редакторі**, ви можете побачити, що в грі тепер є нові зображення.

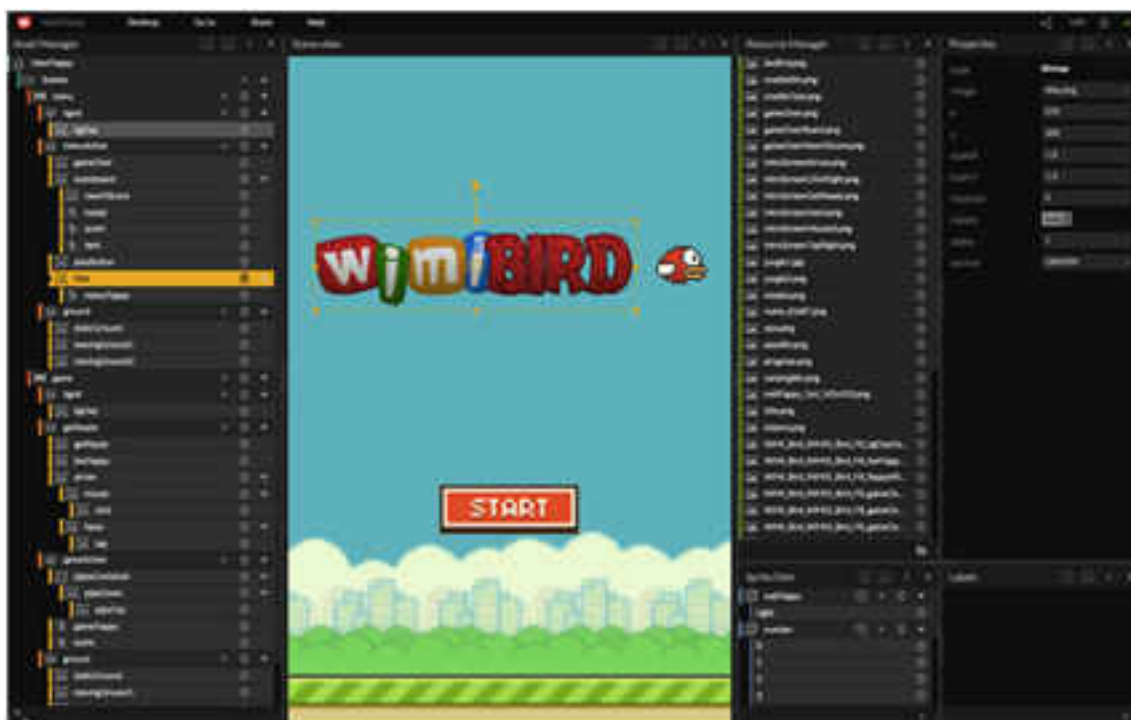


Рисунок 16. Перегляд доступних зображень

Тепер це просто **повторення** процесу з усіма зображеннями, які хочете змінити.

Тепер ми збираємося подивитися, як замінити **Sprites**, оскільки процес трохи відрізняється.

У випадку спрайту ви повинні змінити графіку безпосередньо на панелі "Sprite Defs".

Наприклад, якщо ви хочете замінити спрайт "redFlappy", клацніть на ньому, і ви побачите, що на вкладці "Властивості" ви можете вибрати еталонне зображення (рис. 17).

Просто, **вибравши** нове зображення у випадяючому меню, на панелі перегляду «Сцена» з'явиться нова **анімація**.

Для цього вам доведеться зберегти розмір кожного кадру, визначеного в оригіналі.



559264178cf429fe050025e5.png

Dimensions: 422 x 103

File size: 9.1 KB

MIME type: image/png

URL: https://games.wmi5.com/resources/games/NewFlappy52e1217d900d372156800079/r0_0_1/res/559264178cf429fe050025e5.png

Рисунок 17. Еталонне зображення спрайту

Якщо нова анімація була дещо іншою, вам доведеться відкоригувати значення цієї анімації у властивостях анімації на панелі «SpriteDefs» (рис. 18); в цьому прикладі анімація "правильна".

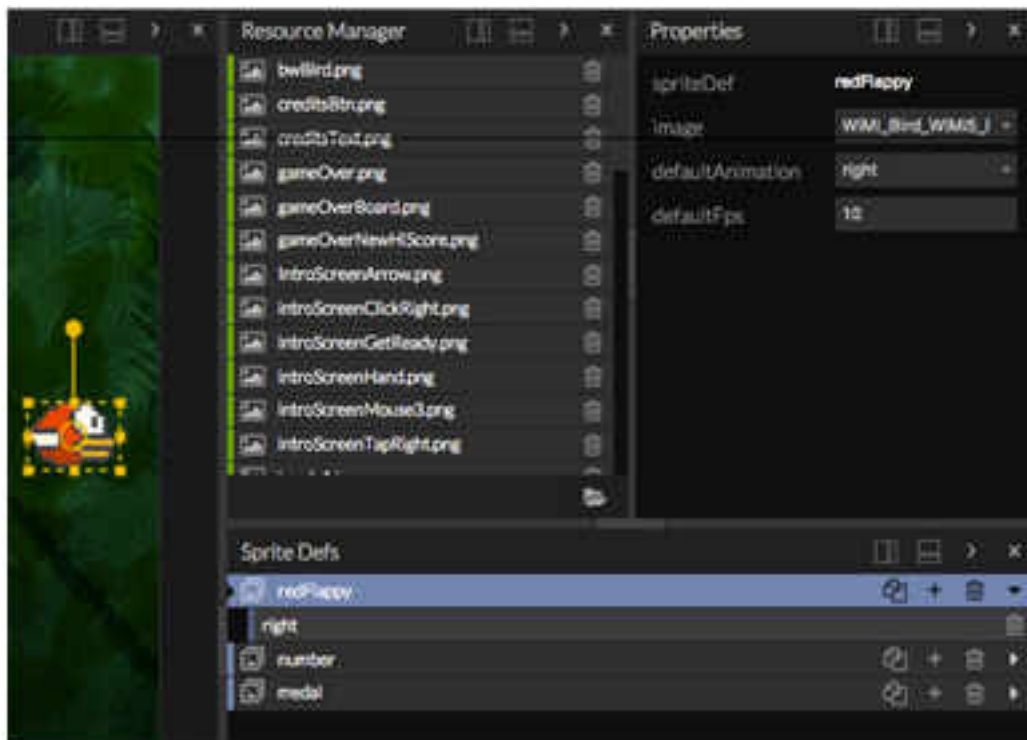


Рисунок 18. Редагування анімації спрайту

І з цим ми пояснили кроки, необхідні для **зміни** та **персоналізації** шаблону гри WiMi5. Як видно, цей процес досить легкий, тому ви можете персоналізувати та створювати власні проекти. Пам'ятайте, що публікація цих ігор також дуже проста.

НАЛАШТУВАННЯ ШАБЛОНІВ

[Шаблон "Карты"](#)

Цей проект розроблено для того, щоб служити шаблоном для ваших власних картіжних ігор (рис. 19).



Рисунок 19. Шаблон Карты

Цей проект можна знайти на інформаційній панелі редактора WiMi5 під розділом "Освіта" при створенні нового проекту.

Карты пам'яті – це гра для поліпшення пам'яті для людей у віці від 5 до 99 років.

Гра полягає в тому, щоб розвернути всі карти, які обличчям донизу до закінчення часу. Для цього треба натискати на карти попарно, підбираючи підключені карти. Якщо ви зрозумієте це неправильно, обидва картки повернуться обличчям донизу. Ви виграєте гру, коли ви перевертаєте всі карти в стільки раундів, скільки вказано в раунді у верхньому лівому куті.

Цю гру можна легко персоналізувати. Все, що вам потрібно зробити, це виконати кілька простих кроків, описаних нижче.

[Шаблон "Відповідність"](#)

Цей шаблон розроблено, щоб слугувати проектом для вашої власної гри відповідних понять (рис. 20).



Рисунок 20. Шаблон Відповідність

Ви можете знайти його на інформаційній панелі редактора WiMi5 у розділі "Освіта" під час створення нового проекту. Зокрема, це полягає в тому, що приєднуються проблеми зі своїми відповідями, спрямовані на 5-8-річних.

Ця гра була розроблена таким чином, щоб тема гри могла бути легко модифікована – просто змінити зображення, які використовуються для створення пар (рис. 21).

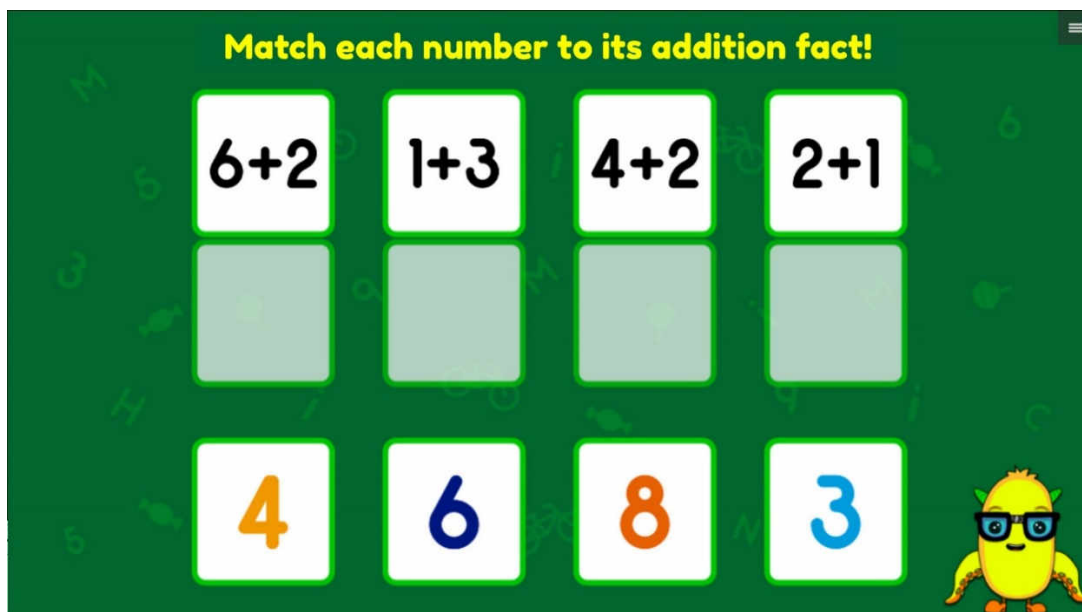


Рисунок 21. Створення пар

Шаблон "Hangman"

Цей проект розроблено для того, щоб служити шаблоном для власних ігор типу Hangman (рис. 22).



Рисунок 22. Шаблон Hangman

Ви можете знайти його на інформаційній панелі редактора WiMi5 під розділом "Освіта" під час створення нового проекту.

Це класична гра-ханж: ви повинні вгадати слова, підбираючи окремі літери, перш ніж персонаж повісить вас (рис. 23).

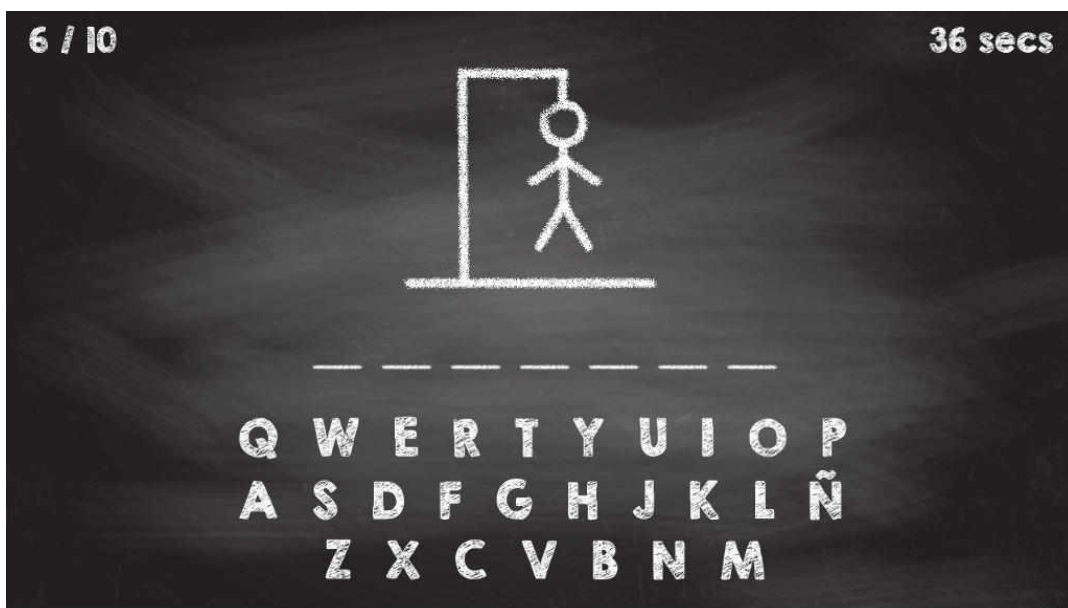


Рисунок 23. Класична гра-ханж

Шаблон "Sequences"

Цей проект розроблений таким чином, щоб слугувати шаблоном для створення ігор, в яких користувачеві потрібно поставити послідовність елементів у порядку (рис. 24), призначену для дітей віком від 5 до 11 років.

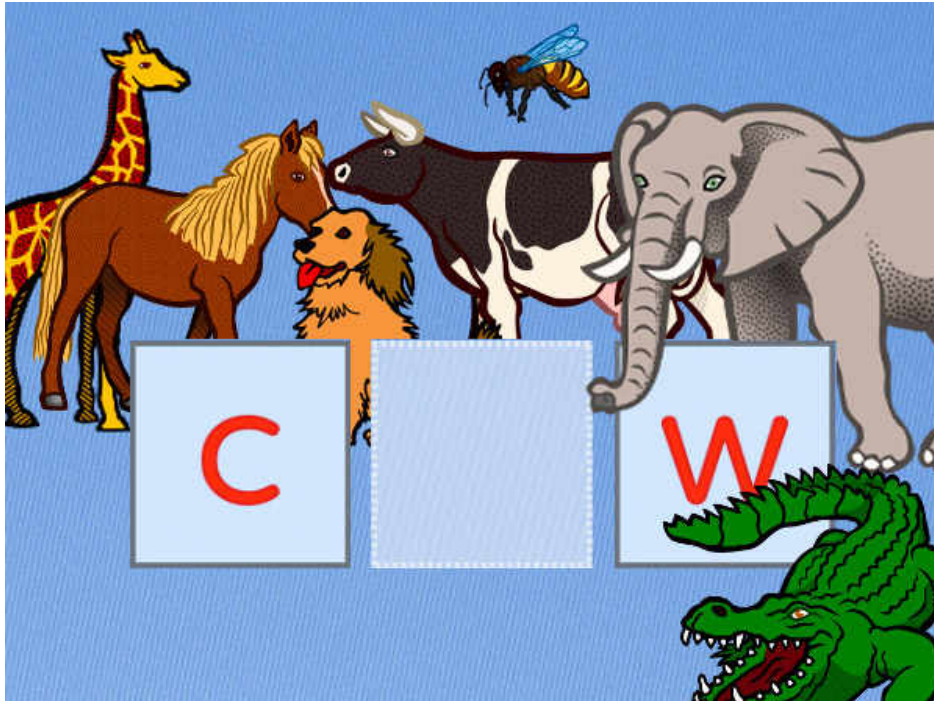


Рисунок 24. Шаблон Sequences

Ви можете знайти цей проект на інформаційній панелі редактора WiMi5 під Розділ "Освіта" при створенні нового проекту (рис. 25).

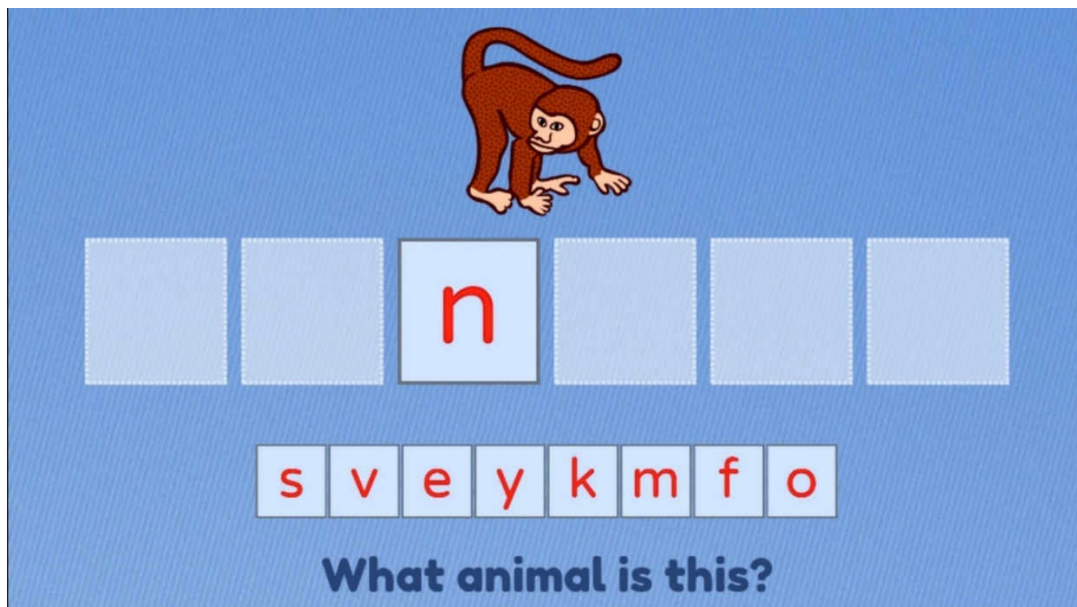


Рисунок 25. Гра Тварини

[Шаблон "Вікторина"](#)

Цей проект розроблений для того, щоб слугувати шаблоном для вашого власного ігрового запитання (рис. 26).

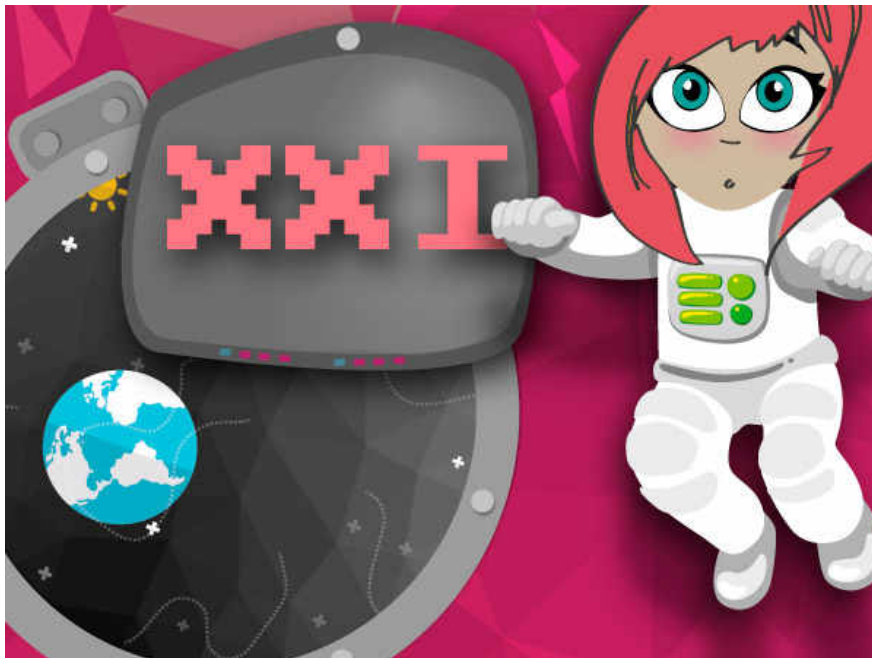


Рисунок 26. Шаблон Вікторина

Ви можете знайти цей проект на інформаційній панелі редактора WiMi5 під розділом "Освіта" при створенні нового проекту.

Навчальний посібник розділений на різні розділи залежно від функціональності гри, яку ви хочете змінити.

[Шаблон "Word search"](#)

Цей проект розроблений для того, щоб слугувати шаблоном для власних ігор пошуку слів (рис. 27).

Ви можете знайти цей проект на інформаційній панелі редактора WiMi5 у розділі "Освіта" під час створення нового проекту.



Рисунок 27. Шаблон Пошук слів

Пошукова гра слів має два варіанти: звичайна, та де метою є пошук серії слів (рис. 28).



Рисунок 28. Гра Пошук серії слів

ЛЕКЦІЯ 2

РЕДАКТОР СЮЖЕТУ ТА РЕДАКТОР ЛОГІКИ У WIMİ5

РЕДАКТОР СЮЖЕТНОГО ПЕРЕГЛЯДУ

Як ми пояснили в першій лекції, робочий стіл редактора сценаріїв є одним з двох основних переглядів інструмента створення WiMi5.

В цій лекції ми розпочнемо з панелей, розташованих у вікні редактора сцени (рис. 29).



Рисунок 29. Панелі редактора сцени

МЕНЕДЖЕР РЕСУРСІВ

На цій панелі ви можете створювати та керувати сценами та шарами, необхідними для вашого проекту (рис. 30).

Щоб створити ігрові об'єкти у вибраному шарі диспетчера активів, вам потрібно перетягнути ресурси зображень з панелі «Зображення» в режим перегляду сцен. Зробивши це, ви побачите нові екземпляри об'єктів в дереві диспетчера ресурсів.

Клацаючи піктограму з трьома крапками, розташованими у кожному елементі, можна виконувати певні операції над ним, наприклад видалити його, клонувати його або створювати інші дочірні елементи (нові шари, порожні 2D-об'єкти або тексти залежно від типу елемента, до якої застосовується ця операція).

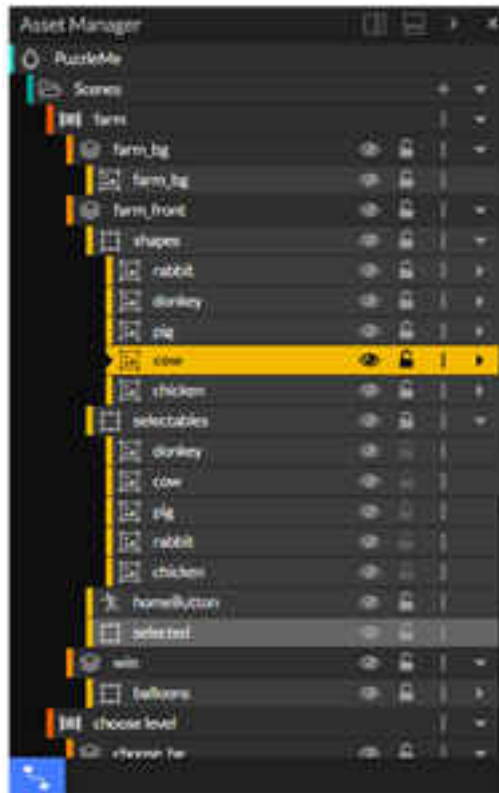


Рисунок 30. Панель Менеджера ресурсів

Ви також можете заблокувати елемент, а в дитячій грі - за допомогою кнопки блокування приховати його в режимі перегляду сцени за допомогою значка очі.

Можна переміщати шари і змінювати свої позиції, перетягнувши їх і перетягнувши їх.

Для розташування шарів ви повинні знати, що нижній шар відповідає верхній позиції у вікні перегляду сцени. Отже, якщо ви хочете, щоб зображення було видно в передній частині екрана, ви повинні мати свій шар об'єкта внизу диспетчера ресурсів.

ВИГЛЯД СЦЕНИ

На цій панелі ви можете перетягувати і викинути свої графічні ресурси, щоб створити сцену.

Ви можете рухати, змінювати розмір або обертати будь-який з них, щоб скласти чудову сценку гри.

Якщо ви бажаєте більш точно змінити елемент гри, ви завжди можете використовувати панель властивостей, щоб змінити вручну значення його властивостей.

Якщо вибрано будь-який елемент сцени, його можна видалити або клонувати, написнувши на нього правою клавішею миші.

Розташований у верхньому лівому куті панелі інструмент для встановлення опорної сітки може допомогти вам легко розташувати об'єкти сцени (рис. 31).

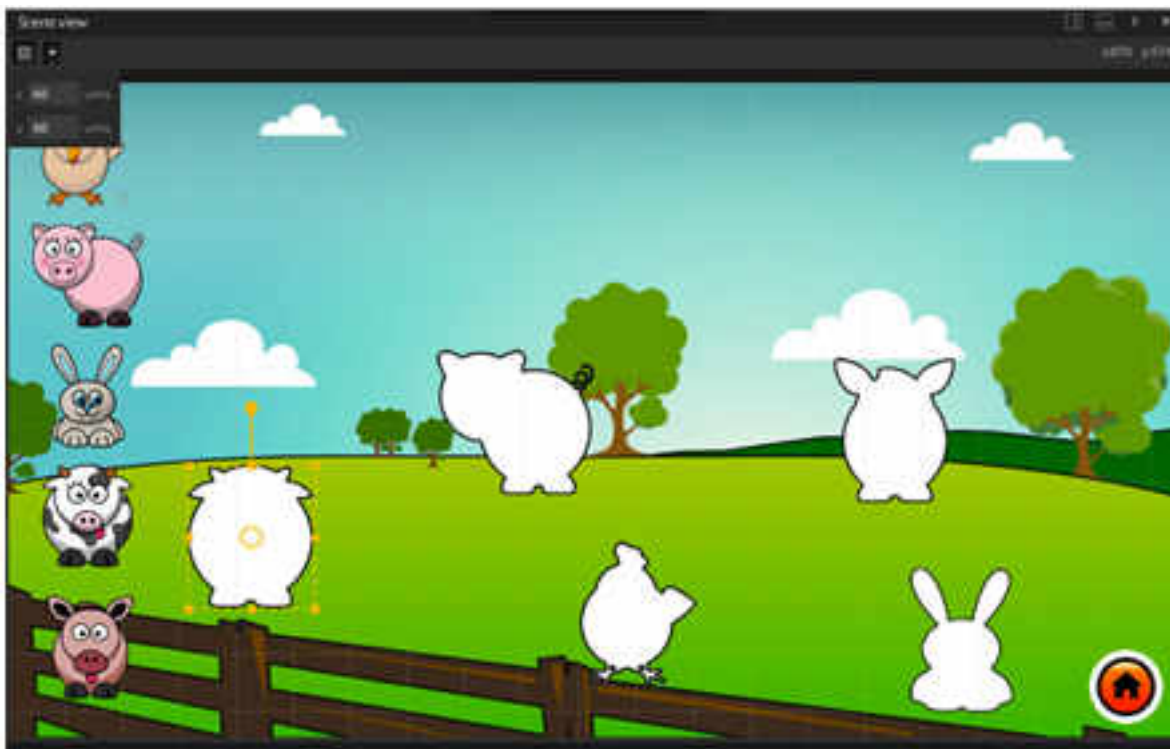


Рисунок 31. Інструмент опорної сітки

ПАНЕЛЬ ЗОБРАЖЕНЬ

Ця панель служить для завантаження зображень або звуків до вашого локального місця зберігання WiMi5. Для цього використовуйте кнопку у нижньому правому куті панелі або просто перетягніть їх з файлової системи через вікно редактора (рис. 32).

Буде показано спливаюче вікно, в якому відобразатимуться доступні зображення та аудіо ресурси. У вас немає обмежень щодо розширень файлів для ваших зображень або звуків. Просто пам'ятайте про конкретні розширення файлів, які підтримують веб-переглядачі. У вашому локальному обладнанні WiMi5 існує обмеження в розмірі 100 Мб.

За допомогою цієї панелі ви можете завантажувати зображення та звуки у ваш хмарний накопичувач та розташувати їх у папках.



Рисунок 32. Панель зображень

Ці ресурси будуть доступні для всіх ваших проектів. Щоб використовувати ці ресурси у вашому проекті, потрібно вибрати їх та натиснути кнопку «Додати до проекту» (рис. 33).



Рисунок 33. Додавання ресурсів до проекту

ПАНЕЛЬ СПРАЙТІВ

У цій панелі ви можете легко створювати анімовані об'єкти (рис. 34).



Рисунок 34. Панель спрайтів

Вам знадобиться зображення зі спрацьовуванням, завантажене на панель «Зображення». Щоб розпочати створення спрайту, просто натисніть на кнопку, розташовану в нижній частині панелі, або перетягніть зображення з панелі зображень на цю панель. Роблячи це, ви створите новий спрайт – зображення з анімацією за замовчуванням.

Ви можете змінити ім'я спрайта, двічі клацнувши на ньому.

Кнопка з 3-ма точками в елементі дозволяє видалити і клонувати спрайт або створити в ньому нові анімації.

Після створення спрайту ви можете налаштувати різні його властивості на панелі властивостей (зображення, анімація за промовчанням тощо).

Вибравши кожну анімацію, ви також можете налаштувати його властивості, якщо це необхідно (висота, ширина, кадри / кадри, номер кадру тощо) на панелі Властивості.

ПАНЕЛЬ ВЛАСТИВОСТЕЙ

За допомогою цієї панелі ви можете налаштувати всі параметри ваших графічних ресурсів, розташованих на панелі перегляду сцени (рис. 35).



Рисунок 35. Панель властивостей

Ви можете налаштувати параметри для положення x , y , масштаб, видимість, альфа тощо.

Ви також можете визначити деякі функції шару, такі як видимість або альфа.

Нарешті, ви також можете керувати мітками для кількох груп зображень або анімацій, які ви хочете використовувати.

ПАНЕЛЬ МІТОК

На цій панелі ви створюєте та керуєте мітками (рис. 36).

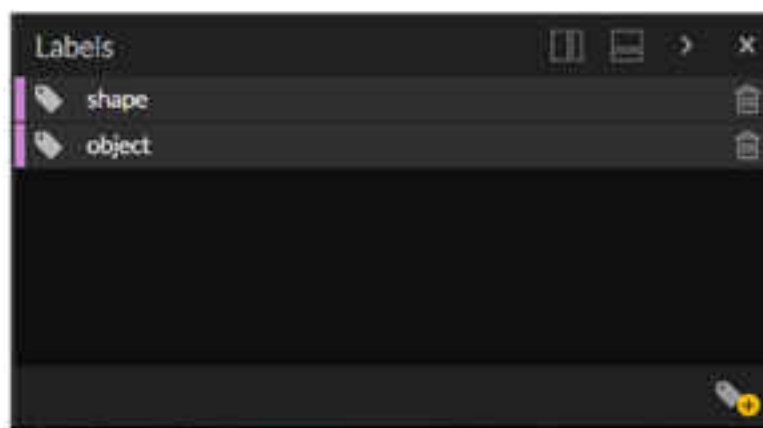


Рисунок 36. Панель міток

За допомогою міток ви можете групувати набір ресурсів і використовувати цю групу для будь-яких цілей.

Наприклад, у вас може бути група Blackboxes для використання об'єктів залежно від їх міток.

ПЕРЕГЛЯД ЛОГІЧНОГО РЕДАКТОРА

Коли ви відкриваєте або створюєте проект з інформаційної панелі, ви будете входити на робочий стіл Logic Editor (рис. 37).

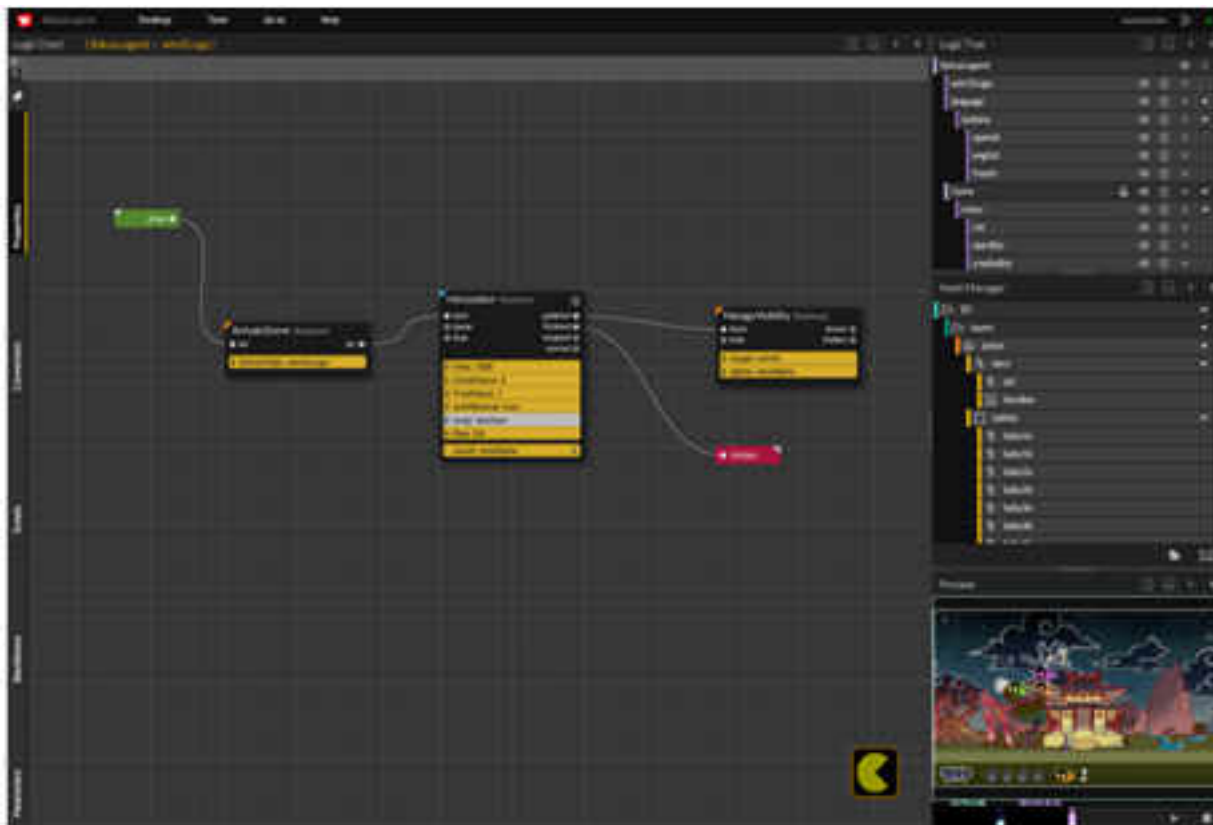


Рисунок 37. Редактор логіки

У цьому макеті робочої області ви можете створити логіку вашої гри.

Ця робоча область є основою ваших ігор HTML5, тут ви зможете розробити всі дії та поведінку для елементів вашої гри.

У вас буде кілька панелей, що містять функції, які дозволяють створювати та керувати сценаріями, поведінкою та діями вашої гри.

Давайте поглянемо на ці панелі.

ЛОГІЧНА СХЕМА

У цій панелі (рис. 38) ви зможете створити всі логічні сценарії для вашої гри. Тут ви можете перетягувати Blackboxes, що містять дії, правила та поведінку для вашої гри.

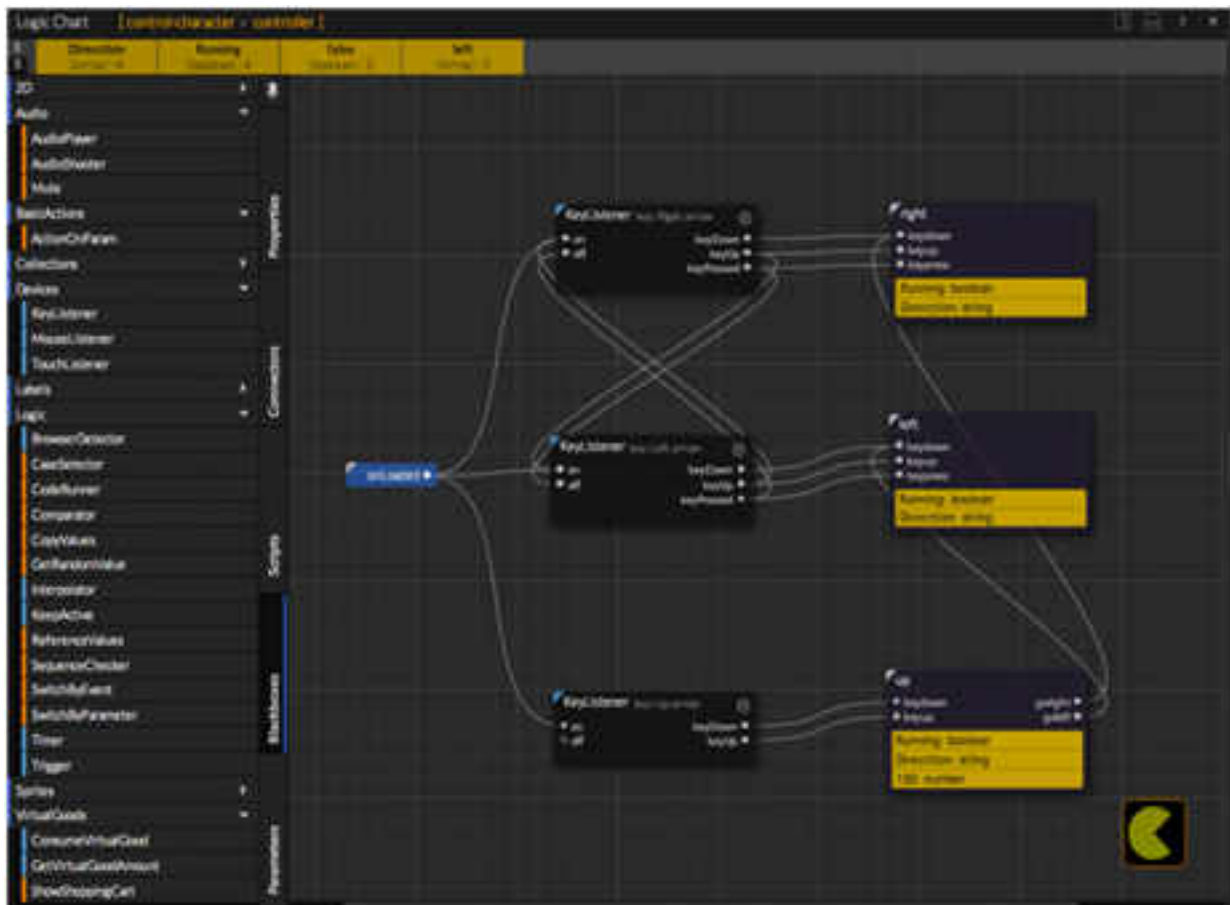


Рисунок 38. Панель Логічна схема

Ви можете переглядати скрипти, двічі клацнувши всередині або за межами них, щоб ввести скрипт, щоб побачити його Blackboxes або перейти на верхній рівень (рис. 39).



Рисунок 39. Перегляд скрипта

Ви також можете перетягувати компоненти з'єднання, параметри та властивості, які вам потрібно буде створити за допомогою сценаріїв, що заповнять необхідні параметри Blackboxes та підключати їх до інших (рис. 40). На логічній схемі ви знайдете кілька кнопок, вкладок та панелей. Ми проаналізуємо найважливіші.

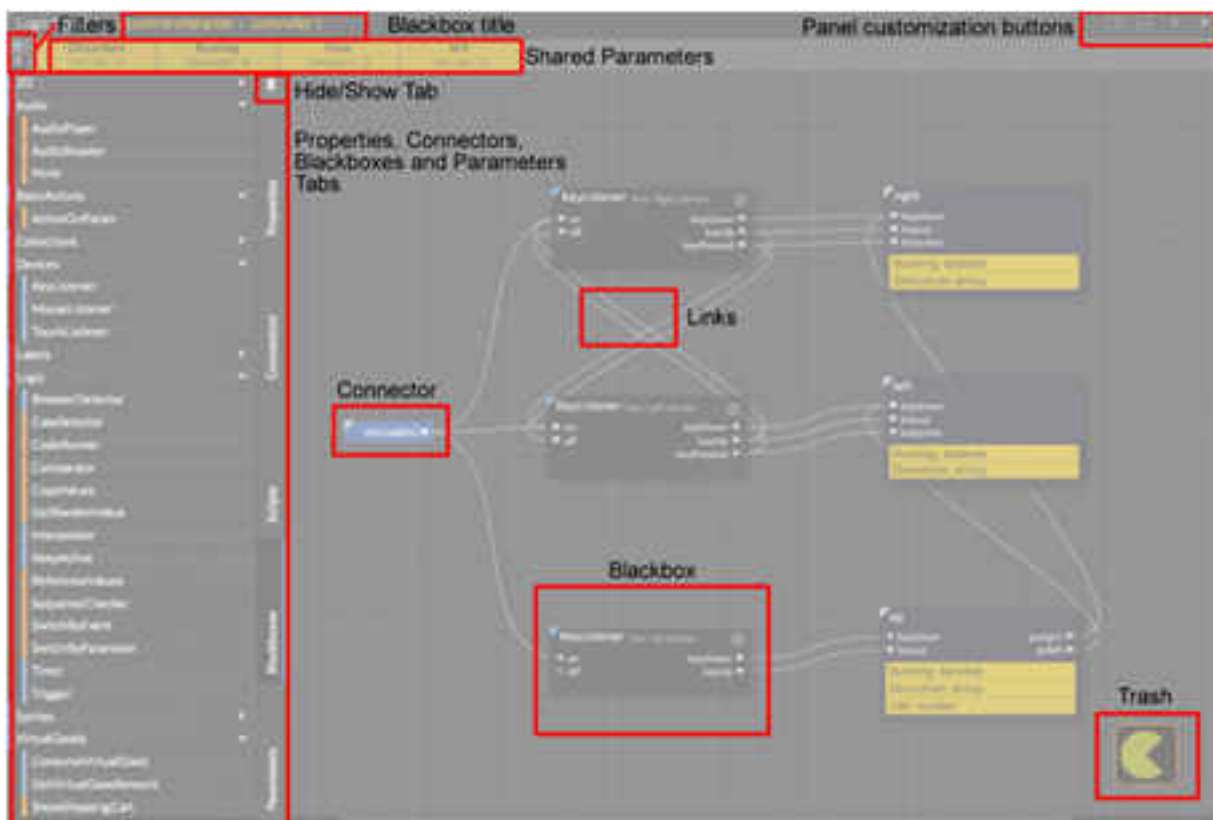


Рисунок 40. Редагування скрипта перетягуванням

ВЛАСТИВОСТІ, З'ЄДНУВАЧІ, ЧОРНІ ВІКНА ТА ПАРАМЕТРИ.

Ці вкладки показують інформацію та ресурси, необхідні для створення логіки вашої гри HTML5. Залежно від вибраної вкладки ви знайдете інший вид інформації.

Ви можете включити кнопку "Показати / сховати вкладку", щоб виправити видимість цих вкладок, або, якщо ви вимкнете її, вкладки будуть відображатися та приховані в залежності від положення миші.

ВКЛАДКА ВЛАСТИВОСТІ

Ця вкладка показує властивості вибраного елемента в логічній діаграмі (рис. 41).

Наприклад, якщо ви виберете MouseListener Blackbox, ви побачите властивості цього Blackbox (ім'я та відомості про кнопки або функції миші).

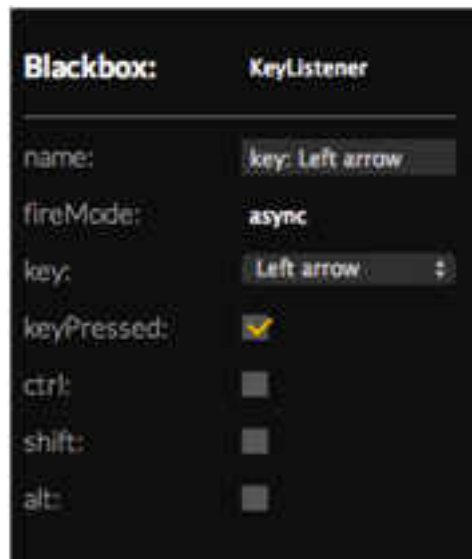


Рисунок 41. Вкладка Властивості

Якщо ви виберете параметр String, ви побачите властивості цього параметра, такі як ім'я, значення, тип, випадки, де він використовується тощо. Ви можете заповнити значення даних для Параметрів на цій вкладці.

ВКЛАДКИ З'ЄДНУВАЧІВ

Коннектори – це події, які ви можете використовувати для підключення сценаріїв (рис. 42).



Рисунок 42. Коннектор

Коли ви перетягуєте коннектор у скрипт, він відображається на верхньому рівні як активатор або тригер у вікні скрипту залежно від характеру роз'єму. Після того, як з'єднання з'єднано, потік програмування буде продовжувати виконувати наступний сценарій.

В даний час у вас є 3 типи роз'ємів: onLoaded, OutputConnector і InputConnector, які можна перейменувати.

У цих зображеннях ви можете побачити зразок з'єднувачів (1 і 2) усередині сценарію та на верхньому рівні.

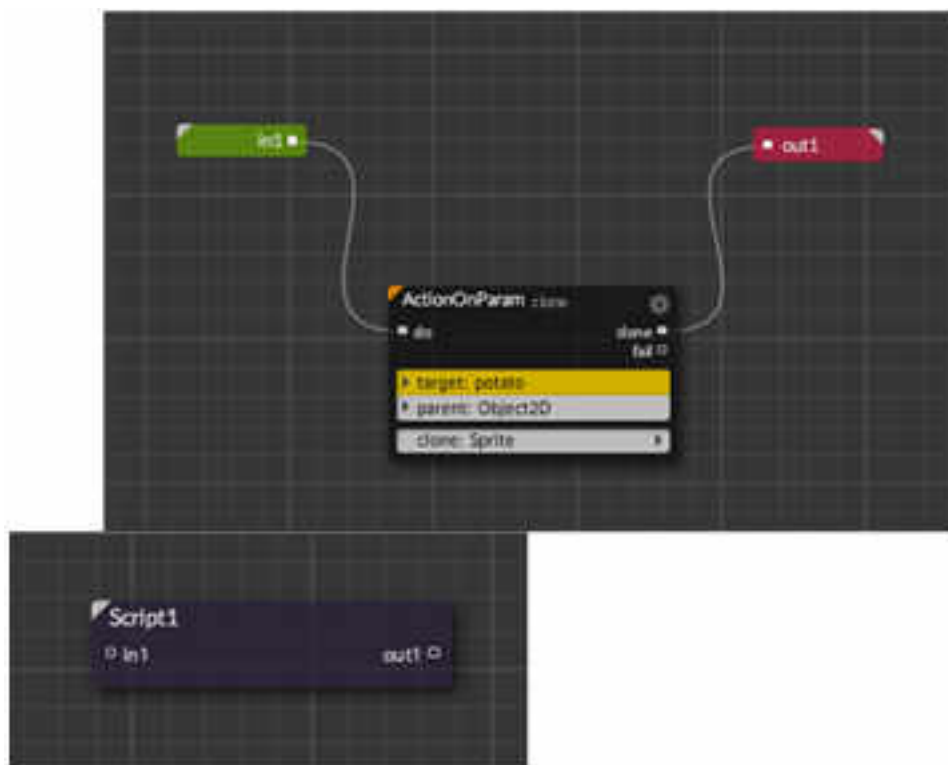


Рисунок 43. Використання коннекторів

ВКЛАДКА BLACKBOXES

На цій вкладці (рис. 44) ви знайдете всі доступні Blackboxes для створення логіки вашої гри HTML5.

Чорні шухляди об'єднуються в 7 категорій. Кожна з цих категорій включає в себе чорні вікна, які ви можете перетягнути з вкладки та падіння на панелі логічної діаграми.

Категорії такі:

- **2D.** Ці чекбоксы пов'язані з управлінням графічними ресурсами, які використовуються у ваших іграх. Ви можете перетворити, клонувати, видаляти об'єкти, керувати текстами, сценами тощо.
- **Аудіо.** Завдяки цим Blackboxes ви можете керувати своїми звуковими ресурсами.
- **Основні дії.** Він містить ActionOnParam Blackbox, один з найпотужніших Blackboxes.
- **Колекції.** Колекційні чорні вікна керують групами або наборами параметрів. Вони зазвичай використовуються для керування масивами.
- **Пристрої.** Ці "чорні ящики" керують входом від декількох пристроїв, як миша, клавіатура або торкаючись екранів.
- **Етикетки.** За допомогою цих "чорних ящиків" ви можете керувати своїми ресурсами, позначеними під певними мітками. Це колективний вид управління ресурсами.

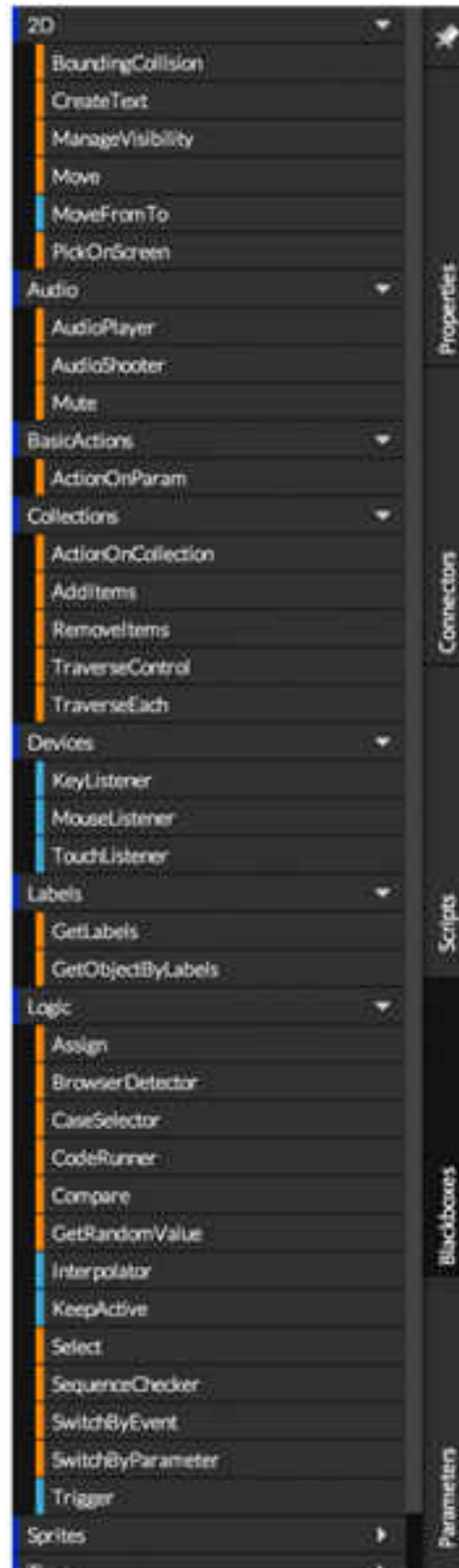


Рисунок 44. Вкладка Blackboxes

- **Логіка.** Логіка Blackboxes дозволяє створювати логічні операції, як порівняння, послідовності, випадкові операції тощо. Ви також можете використовувати спеціальний Blackbox, ScriptRunner Blackbox, який

дозволяє писати код JavaScript, якщо у вас є особлива потреба, яку звичайні чорні вікна не дозволяють.

- **Sprite.** Це спеціальні чорночечки для керування Sprites, які ви можете створювати у режимі перегляду сценаріїв.
- **Віртуальні товари.** Вони керують віртуальними товарами гри, щоб монетизувати їх.

ВКЛАДКА ПАРАМЕТРІВ


На вкладці Параметри (рис. 45) ви можете перетягнути будь-який параметр і скинути на будь-який параметр Output або Input Blackbox.

Перед переміщенням параметра слід перевірити, якщо цільовий Blackbox підтримує той самий тип параметра, який ви обрали.



Рисунок 45. Вкладка параметрів

Якщо тип параметра, який ви перетягуєте через цільовий вхід / параметр Blackbox, відповідає потрібному типу параметра вводу / виводу Blackbox, параметр стає жовтим, і ви можете його скинути.

Ви також можете використовувати колекції замість простих параметрів, перетягнувши правою піктограмою кожного параметра .

ОБЛАСТЬ СПІЛЬНИХ ПАРАМЕТРІВ

Якщо ви хочете використовувати параметри однаково між кількома скриптами, ви можете поставити ці параметри у верхній частині логічної діаграми (рис. 46).

Таким чином, ці параметри будуть доступні на інших рівнях логічної діаграми.

Ви також побачите ці параметри у Blackboxes у верхній частині Вашої логічної діаграми.

Ви можете використовувати кнопки E і S, розташовані у лівій частині цієї області, щоб відфільтрувати вхідні параметри та вихідні параметри сценарію, з яким ви працюєте.



Рисунок 46. Область спільних параметрів

ПАНЕЛЬ ДЕРЕВА ЛОГІКИ

Детально пояснивши панель «Логічна діаграма», ми повертаємося до основних панелей в режимі перегляду логічного редактора.

На панелі логічного дерева (рис. 47) ви можете створити структуру для організації візуальних сценаріїв, розташованих на логічній схемі.

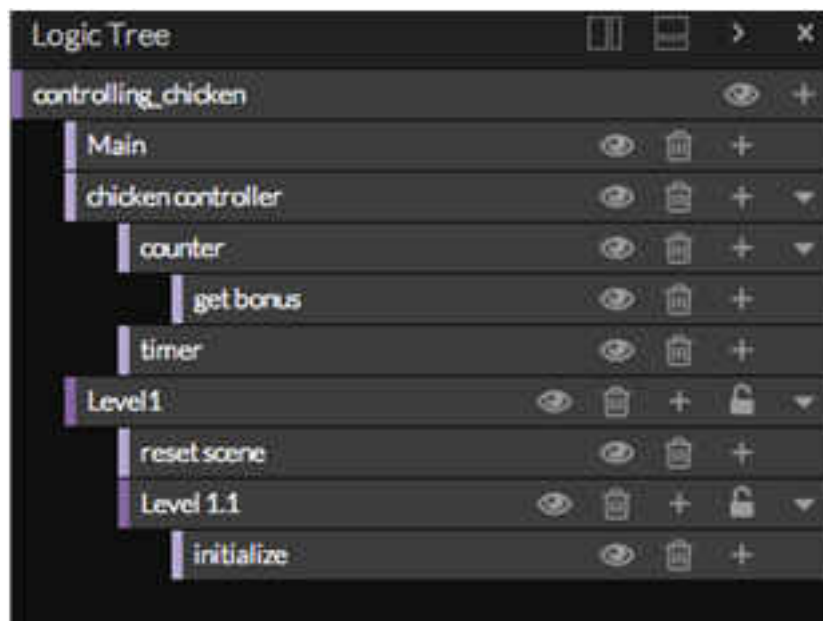


Рисунок 47. Панель дерева логіки

Це своєрідна структура шарів, яка дозволяє вам легко керувати візуальними скриптами, з якими працюєте.

Ви можете створити новий "скрипт", щоб створити нову групу Blackboxes або створити новий "Level", щоб містити кілька сценаріїв.

Це також спосіб керування сценами або рівнями відеоігор.

Ви можете створювати рівні, щоб завантажувати або вивантажувати візуальні скрипти, а також ресурси (зображення та звуки), які використовуються на цих рівнях.

ПАНЕЛЬ МЕНЕДЖЕРА АКТИВІВ

За допомогою цієї панелі (рис. 48) ви можете візуалізувати структуру своїх активів (зображення та звуки, а також шари та сцени), які створені в редакторі сцен.

З цієї панелі ви зможете перетягнути ці об'єкти в параметри вводу/виводу Blackboxes, наприклад, якщо вам потрібно визначити параметр типу 2D Object у своєму Blackbox.

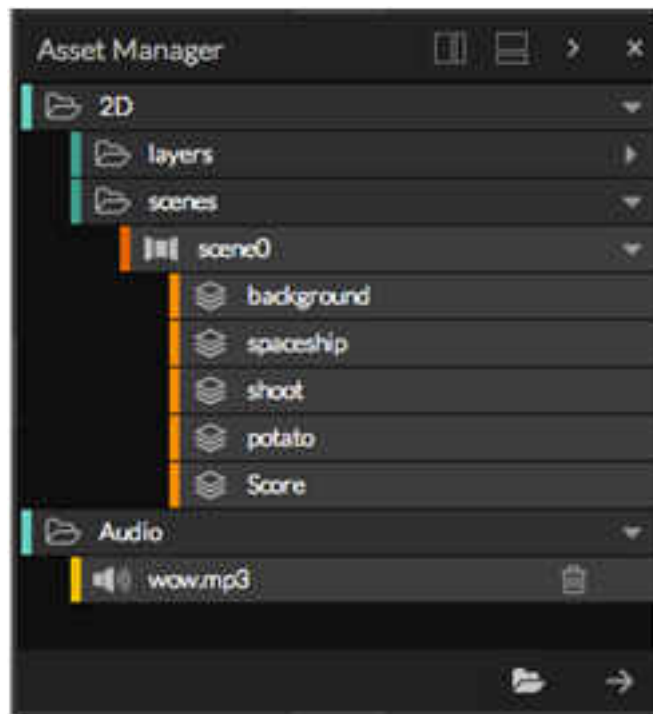


Рисунок 48. Панель менеджера активів

У цій панелі у вас 2 кнопки в нижній області.

Перша кнопка дозволяє перейти до редактора сцени.

Друга кнопка дозволяє завантажувати, завантажувати та списувати звукові ресурси.

ПАНЕЛЬ ПОПЕРЕДНЬОГО ПЕРЕГЛЯДУ

У цій панелі (рис. 49) ви можете використовувати 2 кнопки: відтворити для попереднього перегляду проекту та зупинити попередній перегляд.

Коли попередній перегляд працює, ви побачите зелену лінію, що оточує панель, а також фіолетову, коли попередній перегляд зупинено.

Ви повинні зупинити попередній перегляд, якщо хочете змінити деякі параметри у своїх "чорних скриньках".

У нижньому лівому кутку є також 2 кольорові діаграми, в яких відображаються кадри в секунду (фіолетовий) та пам'ять, використані в проекті (червоний).

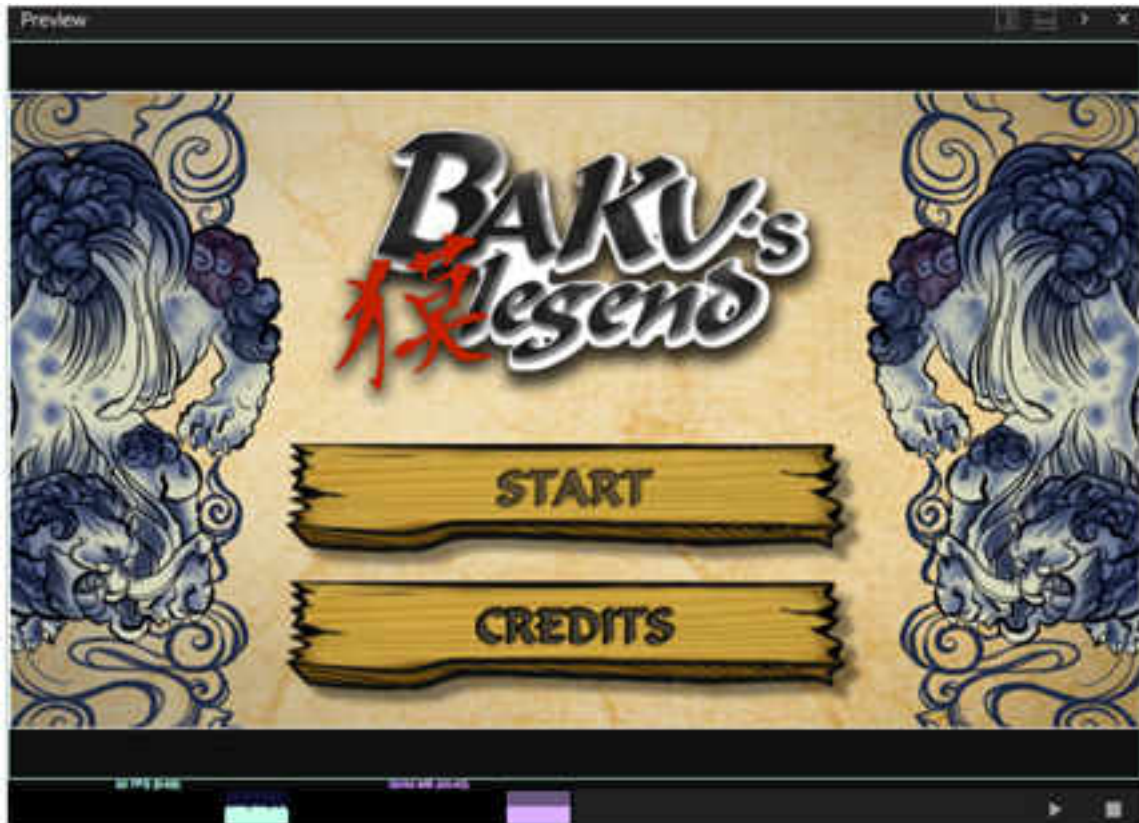


Рисунок 49. Панель попереднього перегляду

Можливо, ви можете сплутати цю панель зі сценою з області редагування сцен. Вони є різними.

У цій панелі ви можете попередньо переглянути всю гру, включаючи всі сцени та, що важливіше, включаючи візуальний сценарій, який застосовується до ваших графічних та звукових ресурсів.

У іншій панелі, редактор сцен використовується для проектування та складання сцен.

ЛЕКЦІЯ 3

ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ У WiMi5

ВІЗУАЛЬНІ ЕЛЕМЕНТИ ПРОГРАМУВАННЯ

У цій лекції пояснюються елементи, які ми збираємося використати для створення гри за допомогою візуального програмування WiMi5.

Також пояснюються основні концепції та механізми редактора WiMi5.

ЧОРНА СКРИНЬКА (Blackbox)

Це основний складовий елемент логіки гри, який візуально відображається як скринька. Він являє собою процес, який може бути запущений за бажанням та в поєднанні з іншими для створення більш складних процесів.

Складається з двох типів елементів:

- **роз'єми**. Роз'єми використовуються для керування виконанням панелі Blackbox. Вони підрозділяються на:
 - **вхідні роз'єми**. Вхідні роз'єми використовуються для запуску або зупинки виконання *Blackbox* панелі (на зображенні буде ввімкнено та вимкнено *TouchListener*, який, як можна здогадатися, використовується для запуску та зупинки Blackbox, а "виконує" *CreateText*)
 - **вихідні роз'єми**. Вихідні роз'єми служать для інформування інших чорних скриньок про те, що відбулася значна подія (на зображенні нижче буде "touch 1 start", "touch 1 end", touch 1 move "*TouchListener*" і "*finished*" з *CreateText*). Таким чином, якщо ми зв'яжемо вихідні роз'єми Blackbox з іншими вхідними роз'ємами Blackbox, ми створимо нитку, яка в кінцевому підсумку буде нашою логікою гри.

Вихідні роз'єми, у свою чергу, поділяються на два типи в залежності від того, як вони поведуться:

- **синхронні**. Вони відрізняються від інших, оскільки вони круглі та помаранчеві. Вихідний роз'єм синхронний, коли він запускається відразу після активації вхідного роз'єму Blackbox. На слайді можна побачити, що вихідний роз'єм "done" з *CreateText* синхронний, що означає, що коли вхідний роз'єм "do" активовано, відразу після процесу Blackbox він запустить "done". Якщо всі вихідні роз'єми є синхронними, ми кажемо, що чорна скринька синхронна, відрізняючи її в помаранчевий колір у верхньому лівому куті.
- **асинхронний**. Вони відрізняються тим, що вони сині і мають форму сльози. Можна побачити, що "touch 1 start", "touch 1

end", "touch move" *TouchListener* є асинхронними. Це означає, що коли включено вхідний роз'єм "включено", жоден вихідний роз'єм не буде негайно спрацьовувати, але буде спрацьовувати *Blackbox*, який керує цим процесом. Є такі ситуації, коли *TouchListener*, керуючи діями, які не є зовсім послідовними, працює як сенсорний екран. Тобто, *BlackBox* не може передбачити, коли користувач торкнеться екрана. Асинхронні вихідні роз'єми використовуються для тих ситуацій, коли вихід або не може бути визначений апіорі, або виходи відрізняються з інших причин. Якщо чорна скринька містить принаймні один такий асинхронний вихідний роз'єм, ми говоримо, що чорна скринька є асинхронною, відрізняючи її синім кольором у верхньому лівому куті.

- **параметри.** Параметри – це контейнери даних, які використовуються для управління інформацією, яку використовує та виготовляє чорна скринька. Є два типи параметрів:
 - **вхідні параметри.** Для деяких чорних скриньок потрібні дані для запуску. Наприклад, якщо ви хочете клонувати символ у грі, чорна скринька повинна знати, що це символ для клонування. Спосіб інформувати чорну скриньку про це використовує вхідний параметр, який буде депонувати символ для клонування. Тоді чорна скринька може читати і клонувати його.
 - **вихідні параметри.** Вихідні параметри *Blackbox* використовуються для зберігання інформації, яка виробляється в результаті виконання. Однак майте на увазі, що *Blackbox* може мати лише вхідні параметри або лише вихідні, або мають обидва. На рис. 50 можна побачити, що *BlackBox TouchListener* має вихідні параметри, які в цьому випадку призначені для *чорної скриньки* для введення контактних координат користувача кожного разу, коли він торкається екрана.



Рисунок 50. Вихідні параметри *BlackBox TouchListener*

ПАРАМЕТРИ

Чорні скриньки, що використовують контейнер параметрів, використовуються для визначення його дій або для надання результату його дій.

Ці параметри можуть мати власні властивості або значення, які можуть використовуватися в нових параметрах, які будуть використовуватися як параметри для чорних скриньок.

Вони є жовтими в чорних скриньках.

КОЛЕКЦІЇ

Колекції – це списки параметрів для використання в "чорних скриньках" (рис. 51). У них є послідовність, що містить параметри, та індекс, який вказує, який саме параметр з послідовності маємо на увазі. Ось чому в Колекції може бути використаний простий параметр, що базується на значенні індексу.

Їхній колір у чорних скриньках оранжевий.

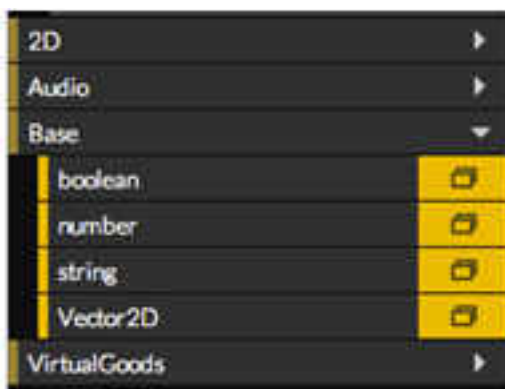


Рисунок 51. Колекції

РОЗ'ЄМ

Вхідна або вихідна точка сценарію (рис. 52).

Вони приєднані до Blackboxes для визначення потоку виконання. Коннектор OnLoaded дуже важливий: це вхідна точка для виконання будь-якого сценарію після завантаження рівня, що асоційований з ним.



Рисунок 52. Роз'єми, або коннектори

Замовлення на виконання виконується на основі внутрішньої ієрархії сценаріїв рівня, виконуючи спочатку першого потомку і всіх його нащадків, після цього виконує другого потомку і всіх його нащадків тощо.

Вхідні з'єднувачі зелено-блакитні для onLoaded і вихідні з'єднувачі червоні.

ПОСИЛАННЯ

Посилання – логічне об'єднання між виходом Connector або Blackbox (тригер) з іншим з'єднувачем або активатором Blackbox, який він запускає (рис. 53).

Він представлений лінією, яка об'єднує елементи.

Ця властивість "order call" визначає порядок запуску для одного виходу Blackboxes (тригер) або Connector.



Рисунок 53. Посилання

СЦЕНАРІЙ

Блок, що містить чорні скриньки та скрипти з їх об'єднаннями та використаними параметрами (рис.54).

Вони визначають нові виконавчі одиниці сценарію (сцени).



Рисунок 54. Сценарій

РІВЕНЬ

Завантажувані та незавантажувані скрипти, що використовуються для управління завантаженням та вивантаженням використаних активів (рис. 55).

Значок замка вказує, як залежить рівень від завантаження чи розвантаження скриптів.

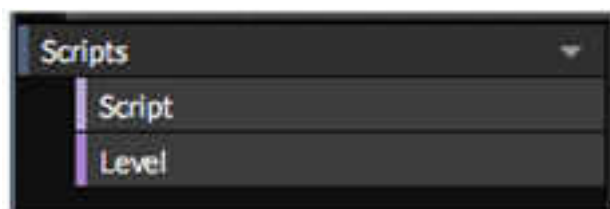


Рисунок 55. Рівні

ОГЛЯД ЕЛЕМЕНТІВ, ЩО ВИКОРИСТОВУЮТЬСЯ В РЕДАКТОРІ WIM5 ПРИ ЗАВАНТАЖЕННІ

Початок кожного виконання сценарію залежить від закінчення завантаження рівня.

Цей процес завершується, коли завантажується останній актив рівня. У цей момент запускаються заряджені рівні Connectors, виконуючи спочатку рівень OnLoaded, потім першу фазу скриптів onLoaded і всіх його нащадків, потім другий фазу сценарію onLoaded із своїми нащадками тощо.

АКТИВАТОРИ (ВХОДИ) І ТРИГЕРИ (ВИХОДИ)

Чорні скриньки мають деякі входи та виходи, які є об'єднаними елементами для ланцюга виконання.

Вхідні дані відповідають за активацію внутрішніх функцій Blackboxes.

Результати - це реакція Blackboxes на тригери виконання завдяки активації їхніх входів.

Таким чином, спосіб побудови сценарію ініціює введення та виведення, що формує логічний потік дій.

ЦИКЛ "ООНОВЛЕННЯ"

Виконання завжди відбувається в циклах оновлення, називається `requestAnimationFrame` у JavaScript.

Це відбувається в момент оновлення відображення та безпосередньо перед ним, так що забезпечується значення та стан елементів для відображення.

Таким чином, як `onLoaded Connector`, так і постійні `Blackboxes` запускають їх вихід у момент візуалізації гри (`requestAnimationFrame`), безпосередньо перед викликом відображення використовуваних рушіїв.

ЛОГІЧНИЙ ПОТІК

Шлях до побудови візуального сценарію в PengoJS – це поєднання подій (виходів-тригерів) з діями (вводами-активаторами) елементів, які називаються "чорними вікнами", на основі яких ці дії викликають нові події (рис. 56).

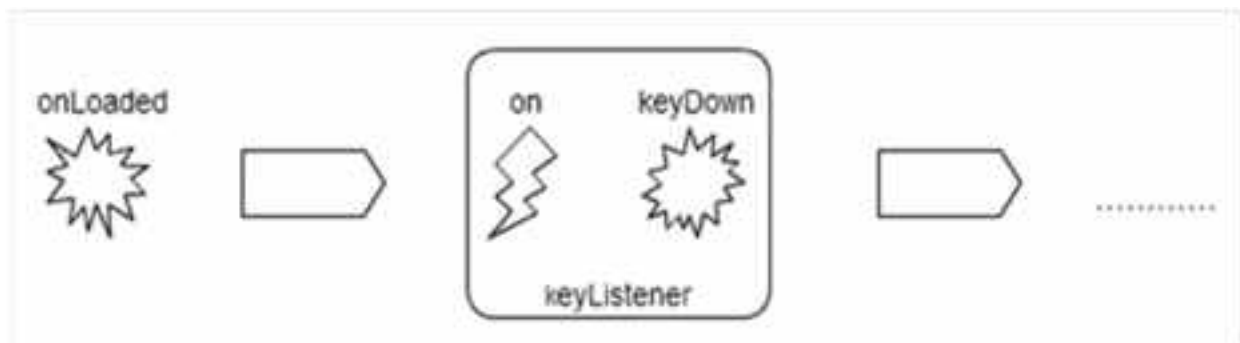


Рисунок 56. Побудова візуального сценарію

Дуже важливо знати, яка акція йде раніше, і що йде далі. Таким чином, важливо пам'ятати, що все виконання відбувається, коли задіяна активація (активатор) ініціює подію (тригер). Через це, коли подія обробляє різні активації для запуску, кожна з них буде глибоко виконана секундно.

Таким чином, в той час як активація викликає нові події, і вони запускають нові активації, наступні активації не спрацьовують.

Ось чому виконання `Visual Scripting` є глибоким. Якщо з певної події (тригера) обробляються різні активації, виконання замовлення визначається по властивості "Call order", починаючи з 0 (рис. 57).

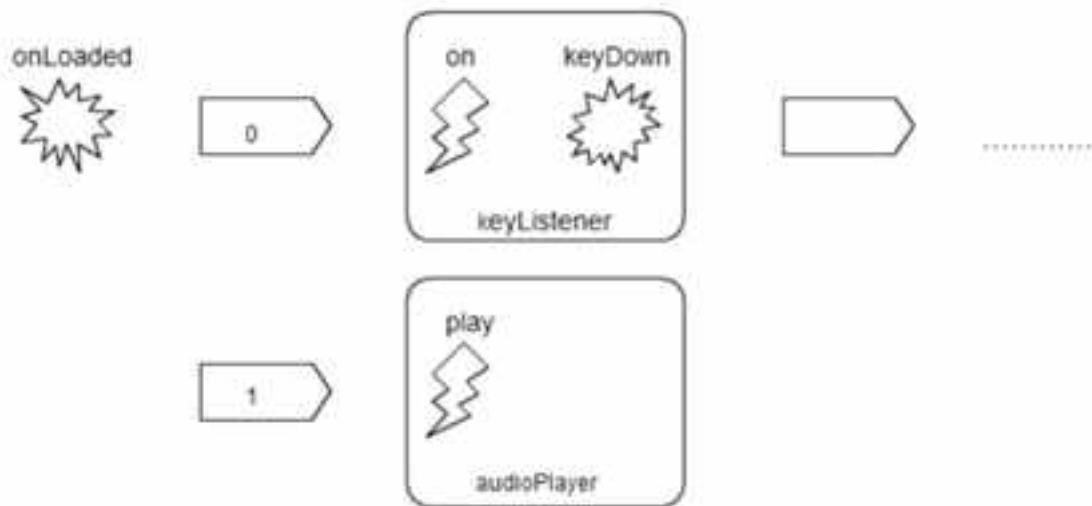


Рисунок 57. Глибоке виконання скриптів

Приклад (рис. 58): Коли рівень, на якому завершується завантаження наступного сценарію, активізує сценарій onLoaded Connector.

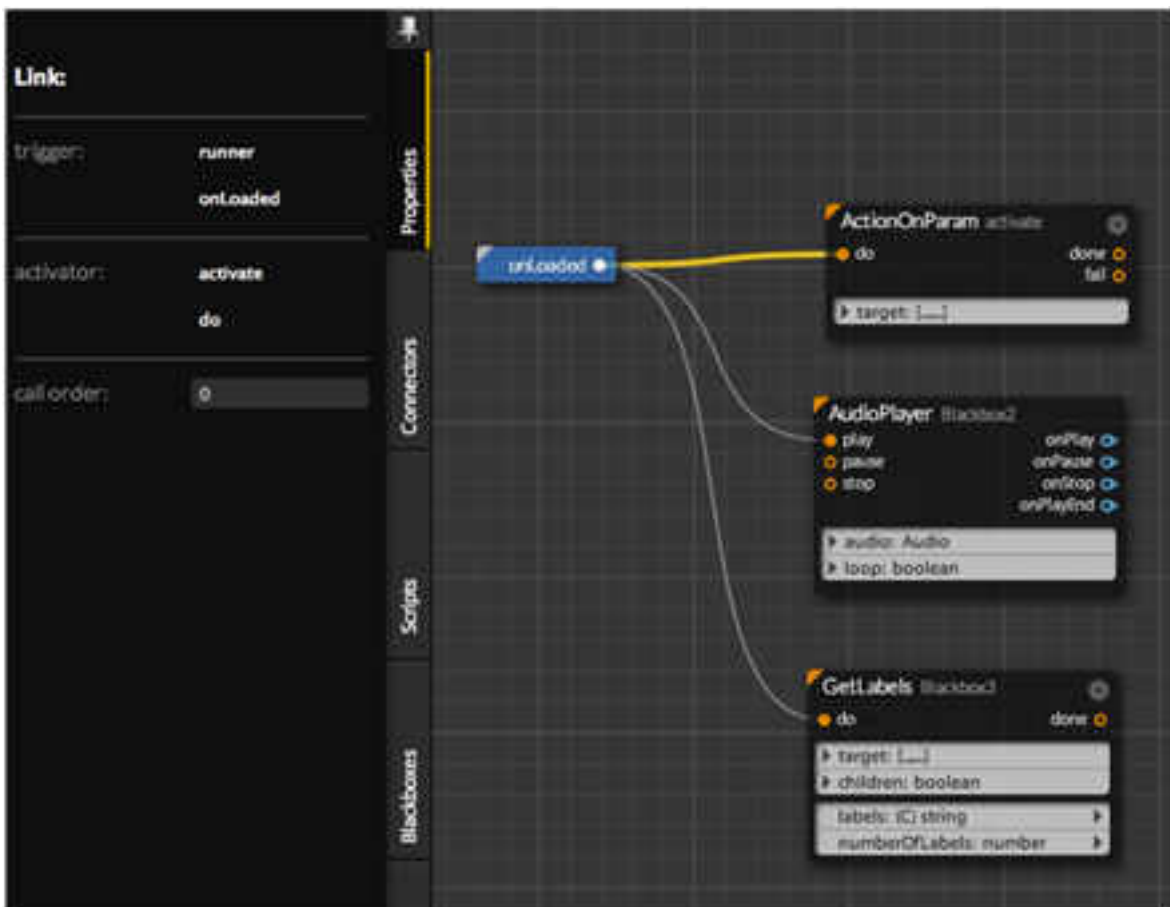


Рисунок 58. Гра Пошук серії слів

Це викликає активацію, підключену до Connector. Спочатку запускається активація активації BlackBox ActivateScene через те, що порядок дзвінків посилань становить 0, який є першим, і всім тригерним ланцюжком, який він виробляє.

Після цього буде знята наступна активація, для якої "Посилання" має "порядок дзвінків" 1.

ОГЛЯД BLACKBOXES

Чорні скриньки – це елементи, які виконують дію і засновані на них, ініціюють виконання нових дій.

Існує два типи чорних скриньок – негайні, які безпосередньо запускаються при залученні, та стійкі, виконання яких залишається стійким, запускаючи дії негайно і затримуючись, але завжди на цикли оновлення.

Залежно від Blackbox, можна ввести вхід, вихід та номер параметра. Цей тип Blackbox ототожнюється з кнопкою + після назви типу Blackbox.

Встановлення виконується за допомогою спадного меню, яке з'являється, клацнувши правою кнопкою миші на Blackbox (рис. 59).

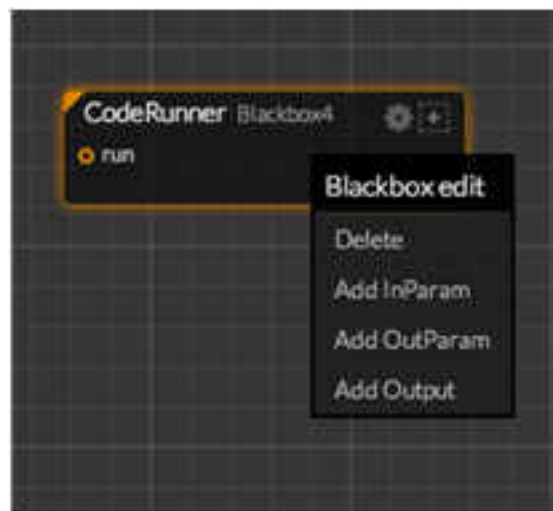


Рисунок 59. Чорні скриньки – встановлення параметрів

Є також чорні скриньки з властивостями налаштування, які визначають його функцію (рис. 60). Вони відзначаються за допомогою шестерні після типу Blackbox.

Доступ до цих властивостей виконується так само, як і з іншими елементами, вибираючи їх та зберігаючи їх на панелі властивостей.



Рисунок 60. Властивості налаштування чорної скриньки

ВИКОРИСТАННЯ ПАРАМЕТРІВ

Параметри – це елементи, що використовуються в Blackboxes для введення параметрів дії виконання, що дають значення або об'єкти дії (рис. 61).

Вони можуть також бути іншими параметрами властивостей або навіть списком параметрів, який використовує елемент у заданому індексі.

Під час вибору призначеного параметра Blackbox вказується тип класу інкапсуляції параметрів (значення, ресурс, посилання або колекція), екземпляри, скрипти, в яких він використовується (у скриптах), тип, його ім'я та відкритий параметр інкапсульованої властивості (вартість).

Параметри використовуються при перетягуванні їх до чорних скриньок з панелі ресурсів (менеджер об'єктів), або перетягуванні їх з панелі елементів (бічна панель інструментів) на вкладці Параметри (рис. 62).

Вони також можуть бути призначені для з'єднувачів Params Blackboxes за допомогою випадуючого меню, що з'являється, клацнувши правою кнопкою миші на них, якщо вони є унікальними.



Рисунок 61. Параметри

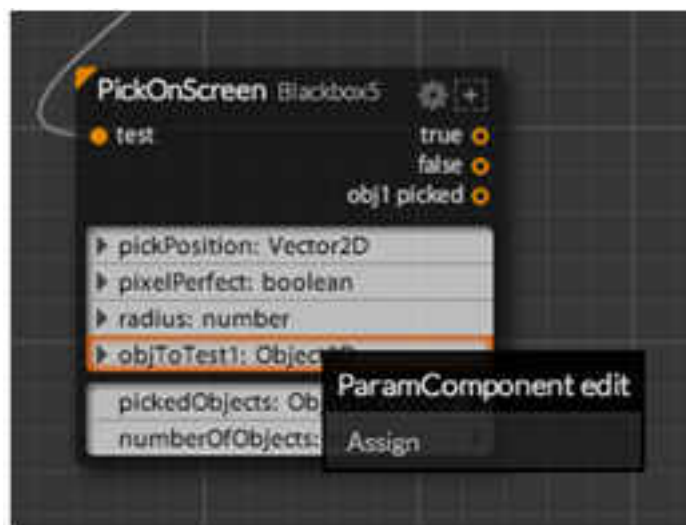


Рисунок 62. Перетягування параметрів

ПАНЕЛЬ СПІЛЬНИХ ПАРАМЕТРІВ

На логічній схемі у кожному скрипті є верхня панель, де показані параметри. У цій панелі відображаються ім'я, тип і кількість параметрів, які діляться між скриптами.

В базі є дві вкладки (рис. 63), відкритих (S) і доступних (E) параметрів. Залежно від цього показані різні параметри.

На вкладці S відображаються параметри, використані в реальному сценарії та в інших. Це автоматично, тому параметри не можна перетягувати та скинути вручну.



Рисунок 63. Панель спільних параметрів

На вкладці E відображаються параметри, що не відображаються за сценарієм. Додавання здійснюється шляхом перетягування та видалення призначеного параметра або видалення їх, витягуючи їх з вкладки.

КОЛЕКЦІЇ, СПИСКИ ПАРАМЕТРІВ

Певний список параметрів типу (рис. 64).


Вони створені перетягуванням Колекції з панелі елементів (бічна панель інструментів) на вкладці параметрів, з потрібного типу значка колекції 



Рисунок 64. Списки параметрів та колекції

Вибір колекції покаже параметри, які вона має. У цьому випадку це параметр з класу Collection, кількість використань, що використовуються, названі екземпляри, що містять параметри типу, ім'я, кількість параметрів, названий елемент, відкриті параметри, названий номер елемента, і відображається параметр значення.

Вручну вилучають відкритий елемент зі списку через значок кошика.

ПАРАМЕТР PARAM, ПОСИЛАННЯ НА ВЛАСТИВІСТЬ ПАРАМЕТРА

Це параметр, що містить значення як властивість іншого Param (рис. 65).

Створення відбувається шляхом перетягування потрібної властивості для використання в якості параметра для Blackbox. Це буде посилання на оригінальну властивість, тому, якщо змінюється початкове значення, то також буде змінюватися і вона, і навпаки.

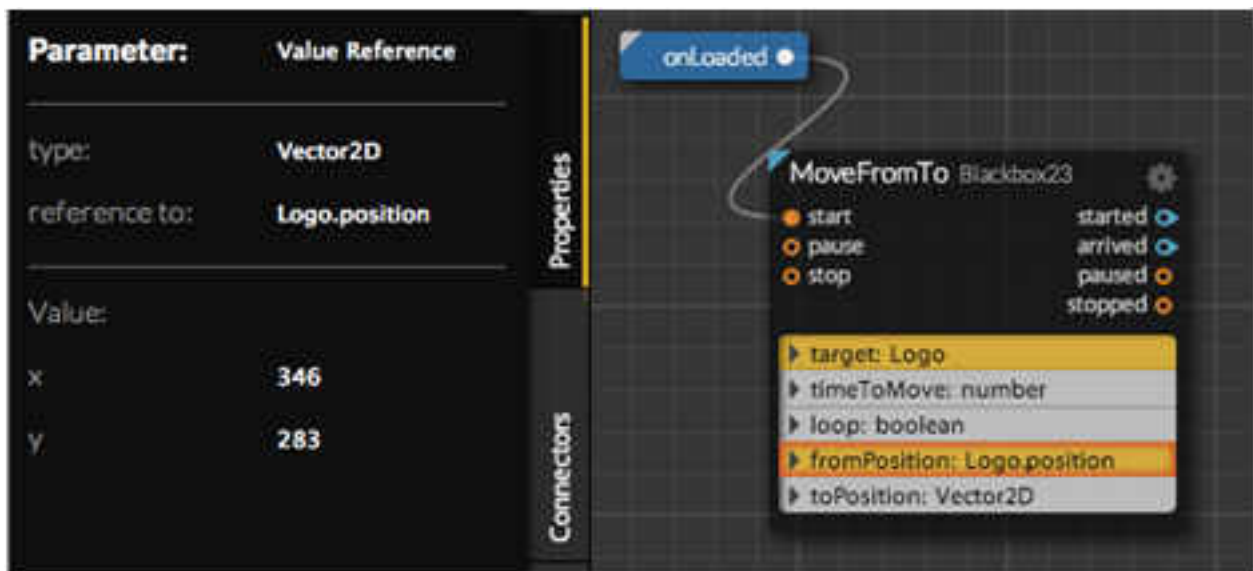


Рисунок 65. Посилання на властивість параметра

ПОДІЯ ПОСИЛАННЯ

Посилання – це лінія, яка об'єднує подію (вихідний / тригер / роз'єм) та активацію (вхід / активатор / роз'єм), що представляє їх ланцюжок (рис. 66).

Вона має властивість "call order", яка визначає порядок викликів на входи від виведення події.

Він створюється шляхом натискання на виході та на вході.

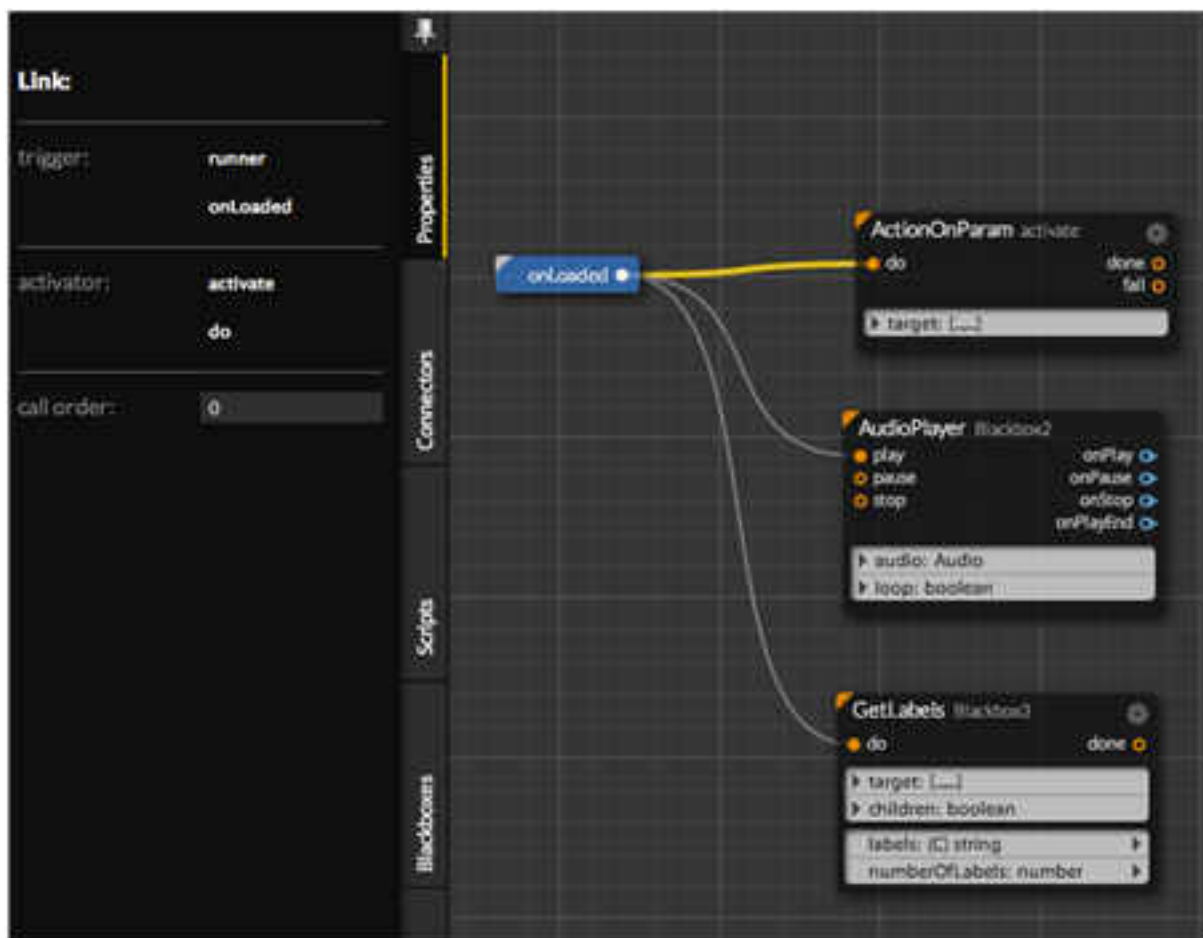


Рисунок 66. Подія посилання

CODERUNNER: ДОДАЙТЕ СВІЙ ВЛАСНИЙ КОД JAVASCRIPT

Всі чорні вікна формуються за допомогою вхідних та вихідних параметрів, активаторів та тригерів. Крім того, деякі з них мають властивості, які можуть налаштувати чорні скриньки з різною поведінкою.

В більшості випадків є чорна скринька або їх комбінація, яка може вирішити поведінку, яку необхідно досягти. Але що трапляється, коли необхідно зробити щось особливе, не передбачене чорними ящиками?

Відповідь – BlackBox CodeRunner.

CodeRunner BLACKBOX є дуже потужним, оскільки дозволяє створити свою власну чорну скриньку.

Є чотири речі, які можна налаштувати:

- вхідні параметри
- вихідні параметри.
- тригери
- вихідний код

Почніть перетягування **CodeRunner BlackBox** з *коду* розділу у вкладку **Blackboxes**. Потім клацніть правою кнопкою миші.

Ви повинні побачити спливаюче меню, подібне до наступного (рис. 67).

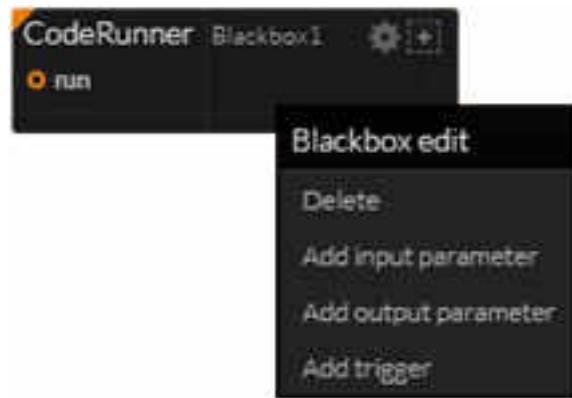


Рисунок 67. Спливаюче меню Скриньки коду

Під час створення параметрів та тригерів елементи меню відрізняються, що показує більше можливостей. Наприклад, якщо ми створимо новий параметр введення, а потім клацнемо чорну скриньку правою кнопкою, побачимо наступне спливаюче вікно (рис. 68).

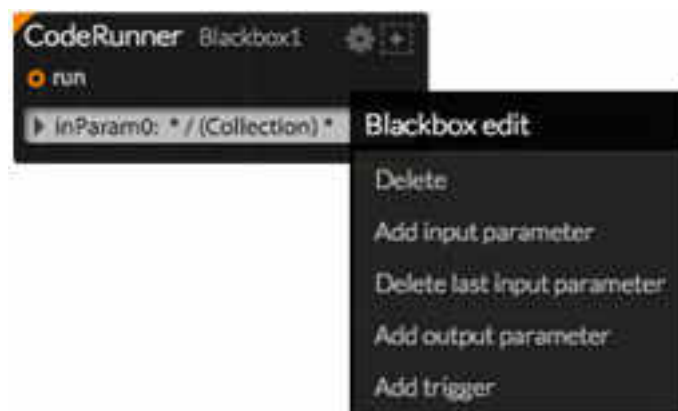


Рисунок 68. Створення параметрів скриньки коду

Зауважимо, що існує нова функція, яка називається **Видалити останній параметр введення**.

Коли ви створюєте більше елементів, наприклад, новий тригер, ви побачите більше варіантів для їх вирішення (рис. 69).

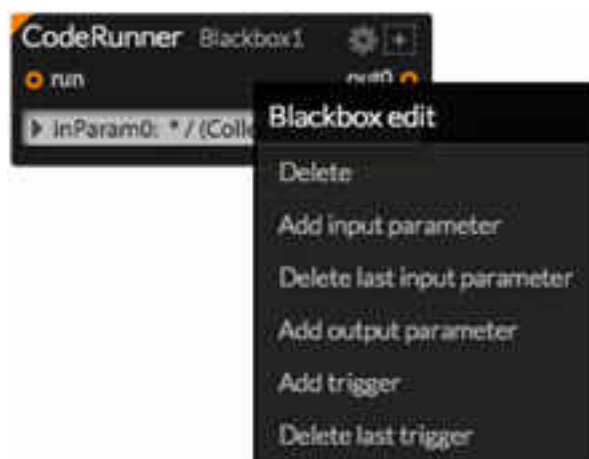


Рисунок 69. Додавання тригерів

Давайте створимо дуже простий приклад, щоб побачити, як працює *CodeRunner*. Уявіть собі, що ми хочемо представити параболічний постріл у нашій грі. В даний час немає шаблонів або скриньок, які займаються фізикою, тому ми повинні створювати власну.

По-перше, перетягніть *CodeRunner* BlackBox з коду розділу в вкладку *Blackboxes*.

Наступним кроком є налаштування параметрів і тригерів, які нам потрібні.

Щоб розрахувати положення (x, y) в будь-який момент часу (t) в параболічному знімку, ми повинні використовувати наступне параболічне рівняння:

$$x = vx * t$$

$$y = vy * t - 0.5 * G * (t ^ 2) \quad // G = gravity.$$

де vx і vy - швидкість обох осей і розраховується наступним чином:

$$vx = velocity * \cos(angle)$$

$$vy = velocity * \sin(angle)$$

Ми повинні врахувати, що формула має дві частини. Спочатку нам потрібна швидкість в обох осях x і y. І тоді ми повинні застосувати вище наведену формулу проходження часу. Щоб уникнути перерахунку, vx і vy в кожній ітерації давайте розіб'ємо логіку на двох CoderRunners.

Отже, у першому нам потрібні наступні вхідні параметри (рис. 70):

- **швидкість**: початкова швидкість пострілу.
- **кут**: кут знімка

І в результаті ми хочемо, щоб вихідний параметр містив швидкість в x (vx) і y (vy). Тому створіть ці вхідні та вихідні параметри.



Рисунок 70. Створення параметрів пострілу

Переименуйте назви параметрів та тригерів, щоб ви легко запам'ятали їх. Для цього натисніть на них і змініть їх назви на вкладці **Властивості**.

Тепер створіть код.

Клацніть чорний ящик, а потім зелений **Редагувати ...** на вкладці **Властивості**.

Ця дія відкриває редактор Javascript в Інтернеті, де ви можете написати код, який буде виконуватися кожного разу, коли **запускається** активатор **запуску**.

Вставте наступний код.

```
var angle = inParams.angle,  
velocity = inParams.velocity, radians;  
radians = ( angle / 180 ) * Math.PI;  
outParams.velocity.x = velocity * Math.cos( radians );  
outParams.velocity.y = -velocity * Math.sin( radians );  
outEvents.done();
```

Ви повинні отримати щось подібне (рис. 71).

У редакторі Javascript ви маєте доступ до таких об'єктів:

- Математика. - це об'єкт Javascript Math для виконання математичних розрахунків.
- Час. - час, що минув від моменту початку виконання гри.
- deltaTime. - час, що минув з останнього ігрового циклу.
- inParams. - об'єкт Javascript, що містить вхідні параметри, визначені розробником. Ви можете побачити їх у розділі **параметрів входу** на вкладці **Властивості**. У цьому прикладі: **швидкість** і **кут**.
- outParams. - об'єкт Javascript, що містить вихідні параметри, визначені розробником. Ви можете побачити їх у розділі **параметрів вихідних даних** на вкладці **Властивості**. У цьому прикладі: **швидкість**.
- Triggers. - об'єкт Javascript, що містить тригери, визначені розробником. Ви можете побачити їх у розділі **Тригери** на вкладці **Властивості**. У цьому прикладі: **виконано**.

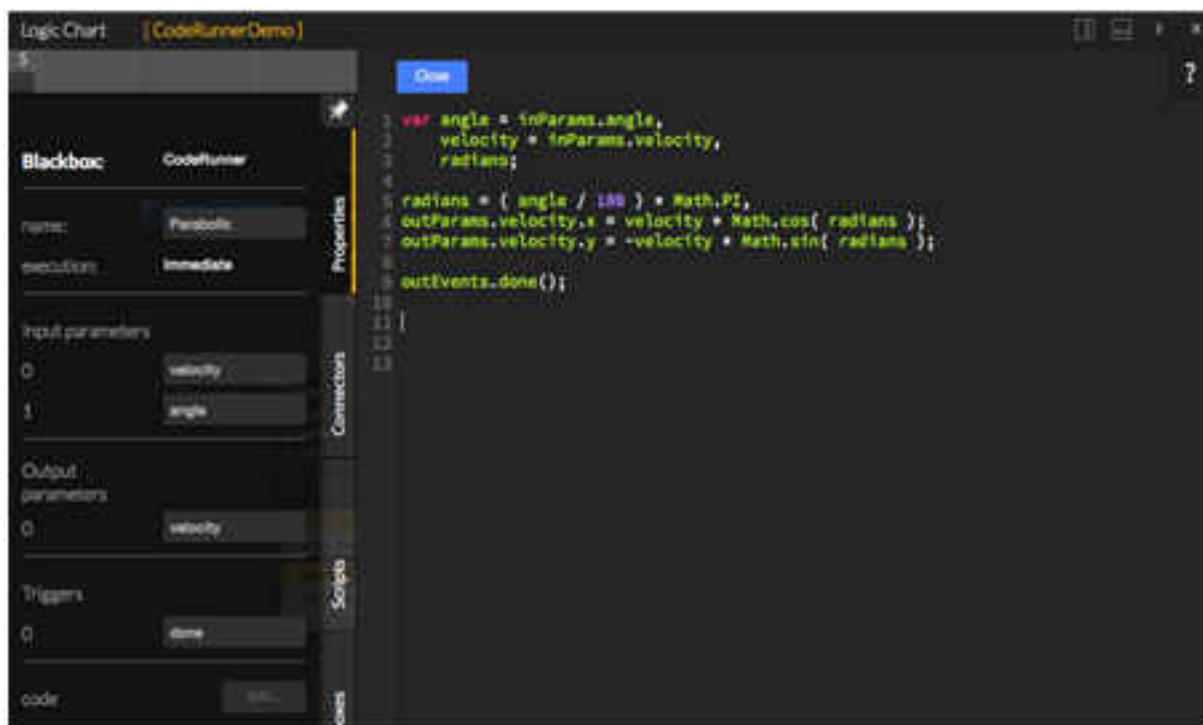


Рисунок 71. Результат запуску коду

Коли ви знаєте, як отримати доступ до параметрів вводу та виводу, а також як виконати тригери, написання коду – це дуже добре.

Давайте подивимося докладно на код.

```
var angle = inParams.angle,
    velocity = inParams.velocity, radians;
```

Спочатку ми створюємо три змінні.

У *кутовій* змінній ми зберігаємо значення вхідного параметра *кута*.

Параметр доступний за допомогою об'єкта *inParams*. Ми повторюємо процес з параметром введення *швидкості*.

Нарешті ми створюємо змінну *радіани*, щоб використовувати його пізніше.

```
radians = ( angle / 180 ) * Math.PI;
```

Тоді ми перетворюємо кут з градусів на радіани, тому що об'єкт *математики* вимагає проходження кутів у радіанах.

```
outParams.velocity.x = velocity * Math.cos( radians );
outParams.velocity.y = -velocity * Math.sin( radians );
```

За допомогою цих двох рядків ми обчислимо швидкість в x і в y та прив'яжемо їх до властивостей x і *швидкості* у вихідного параметра.

Як згадувалося раніше, ми отримуємо доступ до параметра виведення швидкості через об'єкт `outParams`. Зверніть увагу, що цей параметр є *Vector2D*, тому за замовчуванням він має властивості x і y .

Пізніше, коли ми створимо змінну для збереження цього значення, ми повинні пам'ятати, що треба використовувати *Vector2D*, інакше *CodeRunner* не спрацює.

```
outEvents.done();
```

Нарешті, ми закликаємо до пускового *done()* з допомогою *outEvents* об'єкта. У цьому випадку тригери не є властивостями Javascript, а функціями. Ось чому помічайте це за допомогою круглих дужок.

Тепер, коли ми налаштували наш *CodeRunner*, нам потрібно призначити деякі параметри для деяких значень. Важливо зазначити, що всі параметри в *CodeRunner* *нетиповані*. Це означає, що ви не можете створювати змінну з подвійним клацанням, оскільки Blackbox не знає, який тип змінної потрібно створити.

Отже, нам доведеться перетягнути змінну з іншої Blackbox або з вкладки *Parameters*. Давайте використовувати останню, так що клацніть на вкладці *Параметри*, а потім відкрийте *основний* розділ.

Перетягніть змінну *числа* в параметр вводу *швидкості* та іншу змінну *номера* в параметр введення *кута*.

Нарешті, перетягніть *Vector2D* у вихідний параметр *швидкості*. Будьте обережні, щоб не плутати між входом *швидкості* та вихідними параметрами, оскільки вони мають різний тип.

Ви повинні отримати чорну скриньку, подібну до наступної (рис. 72).



Рисунок 72. Результуюча чорна скринька

Значення *швидкості* та *кута* змінили до 100 і 75 відповідно. А також перейменовано в *Vector2D* змінну *XYVelocity*.

Чудово. У нас вже є наш перший *CodeRunner*.

Створимо другий. Повторіть кроки, зроблені вище, доки ви не отримаєте наступну чорну скриньку (рис. 73).