

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ МІЖНАРОДНИХ ВІДНОСИН
КАФЕДРА КОМП'ЮТЕРНИХ МУЛЬТИМЕДІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____ С.М. Лобода

" ____ " _____ 2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР
ЗА СПЕЦІАЛІЗАЦІЄЮ "ТЕХНОЛОГІЇ ЕЛЕКТРОННИХ
МУЛЬТИМЕДІЙНИХ ВИДАНЬ"

**Тема: "Інтерактивне навчальне середовище лазерного стрілецького тренажеру
на основі ігрового рушія"**

Виконавець: студент 215М групи Денисюк Віталій Олегович

Керівник: доктор педагогічних наук, професор Лобода Світлана Миколаївна

Нормоконтролер: _____ ст. викладач Таран Віктор Миколайович

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет міжнародних відносин

Кафедра комп'ютерних мультимедійних технологій

Спеціальність: 186 "Видавництво та поліграфія"

Освітньо-професійна програма: "Технології електронних мультимедійних видань"

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ С.М. Лобода

" ____ " _____ 2019 р.

ЗАВДАННЯ

на виконання дипломної роботи

Денисюка Віталія Олеговича

1. Тема роботи "Інтерактивне навчальне середовище лазерного стрілецького тренажеру на основі ігрового рушія" затверджена наказом ректора від 02.10.2019 р. №2255/ст.
2. Термін виконання роботи: 7 жовтня 2019 р. по 28 січня 2020 р.
3. Вихідні дані роботи: теоретичні відомості до інформаційно-концептуальної моделі веб-ресурсів.
4. Зміст пояснювальної записки:
Розділ 1. Інтерактивне навчальне середовище
Розділ 2. Порівняння ігрових рушіїв
Розділ 3. Розробка інтерактивного навчального середовища лазерного стрілецького тренажеру на основі ігрового рушія
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: фотографії, ілюстрації, презентація *Power Point*.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1	Уточнення постановки задачі	07.10.2019 р.	
2	Аналіз літературних джерел	08.10.2019 р. – 11.10.2020 р.	
3	Обґрунтування вибору рішення	14.10.2019 р. – 18.10.2019 р.	
4	Збір інформації	21.10.2019 р. – 25.10.2019 р.	
5	Аналіз даних та їх класифікація	28.10.2019 р. – 01.11.2019 р.	
6	Ознайомлення із загальними положеннями науково-популярних видань	04.11.2019 р. – 08.11.2019 р.	
7	Розробка технології створення інтерактивного навчального середовища	11.11.2019 р. – 29.11.2019 р.	
8	Створення інтерактивного навчального середовища в програмному середовищі <i>UNITY HUB</i>	03.12.2019 р. – 27.12.2019 р.	
9	Оформлення пояснювальної записки	30.12.2019 р. – 10.01.2020 р.	
10	Оформлення презентації	13.01.2020 р. – 17.01.2020 р.	
11	Отримання рецензій від опонентів	20.01.2020 р. – 24.01.2020 р.	
12	Підготовка до захисту в ДЕК	28.01.2020 р.	

7. Дата видачі завдання: "7" жовтня 2019 р.

Керівник дипломного проекту: _____ Лобода С.М.

Завдання прийняв до виконання: _____ Денисюк В.О.

РЕФЕРАТ

Пояснювальна записка до дипломного проекту "Інтерактивне навчальне середовище лазерного стрілецького тренажеру на основі ігрового рушія": 79 сторінок, 58 рисунків та 17 літературних джерел.

ІНТЕРАКТИВНЕ НАВЧАЛЬНЕ СЕРЕДОВИЩЕ, ЛАЗЕРНИЙ СТРІЛЕЦЬКИЙ ТРЕНАЖЕР, ІГРОВИЙ РУШІЙ, *UNITY*, ІНТЕРАКТИВНЕ ТЕСТУВАННЯ.

Об'єктом дослідження даного дипломного проекту є інтерактивне навчальне середовище та ігрові рушії.

Предметом дослідження є створення інтерактивного навчального середовища на основі ігрового рушія.

Метою дипломної роботи є створення інтерактивного навчального середовища для перевірки знань за допомогою лазерної стрілецької зброї.

Досліджено моделі змішаного навчання, розглянуто основні ігрові рушії, розроблено навчальне середовище на основі ігрового рушія.

Матеріали дипломної роботи рекомендується використовувати для перевірки знань студентів військових кафедр та спеціалізованих навчальних центрів з військовою підготовкою.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	7
ВСТУП	8
РОЗДІЛ 1. ІНТЕРАКТИВНЕ НАВЧАЛЬНЕ СЕРЕДОВИЩЕ.....	10
1.1. Класифікація моделей змішаного навчання.....	10
1.2. Модель " <i>Face-to-Face Driver</i> "	13
1.3. Модель " <i>Rotation</i> "	13
1.4. Модель " <i>Flex</i> "	15
1.5. Модель " <i>Online Lab</i> "	15
1.6. Модель " <i>Self-blend</i> "	16
1.7. Лазерні стрілкові тренажери.....	17
Висновки до розділу	27
РОЗДІЛ 2. ПОРІВНЯННЯ ІГРОВИХ РУШІЇВ	28
2.1. Логіка вибору ігрового рушія	28
2.2. <i>Unity3D</i>	28
2.3. <i>CryEngine 3</i>	31
2.4. <i>Unreal Engine 4</i>	31
2.5. <i>HeroEngine</i>	32
2.6. <i>Rage Engine</i>	33
2.7. <i>Phaser</i>	34
2.8. <i>Construct 2</i>	36
2.9. <i>Corona SDK</i>	38
2.10. <i>GameSalad</i>	40
2.11. <i>Lumberyard</i>	41

2.12. <i>LibGDX</i>	43
Висновки до розділу	46
РОЗДІЛ 3. РОЗРОБКА ІНТЕРАКТИВНОГО НАВЧАЛЬНОГО СЕРЕДОВИЩА ЛАЗЕРНОГО СТРІЛЕЦЬКОГО ТРЕНАЖЕРУ НА ОСНОВІ ІГРОВОГО РУШІЯ	47
3.1. Розробка інтерактивного навчального середовища лазерного стрілецького тренажеру на основі ігрового рушія.....	47
Висновки до розділу	75
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

ПМ – пістолет Макарова.

ПЯ – пістолет Яригіна.

ІЛТ – інтерактивний лазерний тир.

ЕМ – електронні мішені.

ЛЦВ – лазерний цілевказівник.

ММГ – Макети масогабаритні.

VR – Virtual Reality – Віртуальна реальність.

UE4 – Unreal Engine 4.

ІІ – штучний інтелект.

PC – Personal computer – Персональний комп'ютер.

PS3 – PlayStation 3.

PS4 – PlayStation 4.

SC2 – StarCraft 2.

NFS – Need For Speed.

ВСТУП

Актуальність теми. Сучасні умови життя та діяльності людей передбачають та вимагають постійної модернізації навчального процесу. Причиною цьому є стрімкий розвиток новітніх технологій. Створення нових способів передачі та перевірки знань, дозволяє швидше вчитись в будь-якій точці світу, незалежно від фізичних можливостей.

Саме через нові технології, процес створення нових методів навчання та перевірки знань це життєво важливі процеси для сучасних навчальних закладів.

В кожній країні є армія, яка її захищає. Окрім цього, дуже багато людей по всьому світу ходять на військові кафедри чи їх аналоги. Саме для перевірки їх знань й розроблена ця дипломна робота.

Об'єктом дослідження даної дипломної роботи є процес роботи лазерного стрілецького тренажеру на основі ігрового рушія.

Предметом дослідження є робота лазерного стрілецького тренажеру на основі ігрового рушія.

Метою дипломної роботи є демонстрація нової платформи для перевірки знань студентів за допомогою створення лазерного стрілецького тренажеру на основі ігрового рушія.

Наукова новизна отриманих результатів. Створено програму для тестування студентів на основі нової платформи.

Практичне значення отриманих результатів. Продемонстровано роботу платформи для навчання та тестування студентів. Практична значимість дипломної роботи була виявлена при перевірці знань випускників військової кафедри. Створено програму, в основу якої вкладено збільшення візуального матеріалу для забезпечення мультимедійної роботи.

Всі результати, представлені у дипломній роботі, отримані випускником особисто. Було проаналізовано сучасні ігрові рушії та їх можливості. Обрано ігровий рушій *Unity* як показник багато платформного продукту. На основі ігрового рушія було розроблено та впроваджено лазерний стрілецький тренажер.

Матеріали дипломної роботи рекомендується використовувати при перевірці знань випускників військових кафедр, військових гуртків та як тест для перевірки знань для допуску до стрільби.

Прогнозні припущення про розвиток об'єкту та предмету дослідження – застосування у сфері військових технологій для перевірки знань.

РОЗДІЛ 1

ІНТЕРАКТИВНЕ НАВЧАЛЬНЕ СЕРЕДОВИЩЕ

1.1. Класифікація моделей змішаного навчання

Сьогодні в галузі технічної вищої освіти на перший план виходить проблема застарілої матеріально-технічної бази вищих навчальних закладів – фахівці відмічають "закритість" новітніх розробок та їх високу собівартість, що в свою чергу робить їх недосяжними для навчальних закладів. Тож роботодавці відмічають, що молоді спеціалісти, які приходять на підприємства, мають досить широкі фундаментальні академічні знання, добре працюють з літературними джерелами, але досить складно адаптуються до умов виробництва, не завжди "легко перепрофілюються під іншу спеціалізацію", у багатьох з них відсутня розвинена самостійність, почуття самовизначення і творчості в професії.

Вирішення означених проблем вимагає проектування і впровадження в навчальний процес нових методичних систем навчання, які б доцільно поєднували традиційні і сучасні педагогічні технології, включаючи інформаційно-комунікативні. На думку М.І. Жалдака, вбудовування інформаційно-комунікаційних технологій у діючі дидактичні системи, гармонійне поєднання традиційних і комп'ютерно орієнтованих технологій навчання має не заперечувати і відкидати здобутки педагогічної науки минулого, а, навпаки, їх удосконалювати і посилювати, в тому числі і за рахунок використання досягнень у розвитку комп'ютерної техніки і засобів зв'язку. Вважається, що однією з сучасних педагогічних технологій, яка може задовільнити вимоги і сучасної системи вищої освіти, і роботодавців, є технологія змішаного навчання. Як показує аналіз існуючих моделей змішаного навчання, ключовим моментом в їх проектуванні є вибір правильного співвідношення всіх методів навчання, що використовуються, та встановлення оптимального поєднання традиційних технологій з технологіями електронного навчання.

Існує декілька класифікацій моделей змішаного навчання. Зокрема, С. Велединська наводить три моделі: модель підтримки традиційного

аудиторного навчання, збалансована модель, гнучка модель, елективна модель. В зарубіжній практиці традиційно виділяють шість моделей змішаного навчання: модель "*Face-to-Face Driver*", модель "*Rotation*", модель "*Flex*", модель "*Online Lab*", модель "*Self-blend*", модель "*Online Driver*". Три моделі змішаного навчання виділяє А. Антоненко: націлену на навички, націлену на працівника, націлену на компетенції.

Спираючись на результати проведеного дослідження і враховуючи особливості змішаного навчання розглядається модель змішаного навчання технічних дисциплін як методичну систему, яка побудована на основі оптимального поєднання трьох технологій: технологій традиційного навчання, технологій дистанційного навчання, технологій онлайн навчання і дає цілісне уявлення про зміст, внутрішню структуру, взаємозв'язок і взаємозалежність елементів процесу навчання технічних дисциплін. При цьому співвідношення перерахованих технологій навчання визначене на підставі досліджень, які проводилися вченими університету Ватерлоу – 30 % технологій традиційного навчання та 70% технологій дистанційного навчання. Особливістю моделі є створення електронного інтерактивного навчального середовища, яке сприятиме підвищенню ефективності навчання технічних дисциплін через посилення лабораторно-практичної складової навчального процесу.

При проектуванні електронного інтерактивного навчального середовища спиралися на метод комплексного системно-діяльнісного моделювання процесу навчання, який передбачає одночасне поєднання моделювання змісту навчання, моделювання навчально-пізнавальної діяльності студентів та моделювання взаємодії учасників навчального процесу. Моделювання змісту в цій системі є основним, особливості технічної дисципліни обумовлюють підпорядкування змістовного компоненту навчання діяльнісному компоненту, реалізація якого відбувається переважно в процесі виконання студентами лабораторних робіт. Таким чином основою моделі електронного інтерактивного навчального середовища є лабораторний практикум [1].

Перший досвід використання проєктованого інтерактивного навчального середовища, як базового складника моделі змішаного навчання технічних дисциплін, вказує на те, що в таких умовах ефективно забезпечується:

- підвищення ступеня автономності студентів;
- гнучкість моделі навчання;
- оптимальний розподіл часу між аудиторними заняттями, самостійною роботою студентів та онлайн спілкуванням;
- врахування індивідуальних особливостей студентів та їх очікувань;
- збільшення наочності навчання, яка реалізується через використання інфографіки та спеціальних прийомів у поданні теоретичного матеріалу (кольори, виділення, структурування тексту тощо);
- різноманітність способів і засобів педагогічної взаємодії; збільшення обсягу навчального матеріалу та підвищення якості його подання;
- можливість необмеженого доступу до навчальної інформації;
- забезпечення оперативного зворотного зв'язку (рис. 1.1).



Рис. 1.1. Схема інтерактивного навчального середовища

1.2. Модель "*Face-to-Face Driver*"

Традиційна аудиторія зазвичай працює наступним чином. Кожного дня учні збираються, а вчитель веде їх через курсову роботу. Тепер скажемо, що учень цього класу бореться з конкретною концепцією. Як цей студент може отримати допомогу, не перешкоджаючи потоку курсу для інших студентів. Модель "*Face-to-Face Driver*" може бути відповіддю.

Модель "*Face-to-Face Driver*" – це тип змішаного навчання, в якому технології використовуються поряд із традиційним навчанням для підвищення успіху в навчальних результатах. Навчання в класі в режимі віч-на-віч є основним способом викладання в цій моделі, але технологія використовується для доповнення навчання.

Щоб запозичити попередній приклад, студенту, який бореться, в класній кімнаті з "*Face-to-Face Driver*", можуть бути призначені онлайн-виправні завдання. Цей студент може отримати доступ до цієї додаткової допомоги через комп'ютер у класі, комп'ютерну лабораторію в корпусі або навіть з домашнього комп'ютера. Це дозволяє традиційний досвід роботи в класі з додатковим рівнем індивідуалізації в навчанні та навчанні [2].

1.3. Модель "*Rotation*"

Ротаційна модель навчання включає традиційне навчання віч-на-віч із навчанням в Інтернеті. У цьому графік часу поділяється і фіксується між цими двома процесами або він працює на розсуд вчителя для даного курсу. Діти в класі зазвичай відстежують як віч-на-віч, так і онлайн-навчання, а навчання в Інтернеті відбувається індивідуально. Студенти обертаються через навчання в Інтернеті, навчання в невеликій групі та завдання з олівця-ручки. Ця модель включає чотири підмоделі.

Ротація станції: модель повороту, в якій для заданого курсу або предмета студент обертається за встановленим графіком або на розсуд викладача однією онлайн-станцією до іншої, яка може бути діями, такими як невеликі групові

навчання, групові проекти та індивідуальне навчання. Він відрізняється від індивідуально-ротаційної моделі.

Модель обертання лабораторії: студент повертається до комп'ютерної лабораторії з цегли та розчину для онлайн-станції навчання.

Модель перегорнутого класу: У цьому випадку учні обертаються за фіксованим графіком або на розсуд вчителя через навчання в класі та онлайн-навчання після шкільних годин. Інтернет-навчання виступає основним джерелом вмісту, який викладається на парі наступного дня.

Індивідуально-ротаційна модель: У цій моделі обертання вчитель встановлює індивідуальні часові орієнтації учнів серед різних модальностей навчання. Він відрізняється від інших моделей обертання тим, що студенту не потрібно повертатись до кожної наявної станції.

Ротація станції, обертання в лабораторії та обертання в класі перераховуються як справді змішані або гібридні аудиторії, тоді як модель індивідуального повороту межує з більш типовим онлайн-класі. Ротації станцій, лабораторій та перевернутого класу вважаються змішаними або гібридними аудиторіями, оскільки вони відповідають чотирьом критеріям; вони представляють інтеграцію старих і нових стилів, вони розроблені з урахуванням традиційної основної навчальної програми з додаванням онлайн-вмісту, вони тримають учнів на своїх місцях у традиційних кладках, і вони не є простішими версіями класу але інтегровані аудиторії, де вчителю все ще потрібен досвід з традиційних стилів. Модель індивідуального повороту, хоча вона вважається змішаною аудиторією, дійсно наближається до онлайн-навчання. Навчальна програма побудована для особистості, тобто студенти могли самостійно працювати повністю в Інтернеті, якщо цей стиль їм підходить [3].

По всій ротаційній моделі навчання складається з таких компонентів:

1. Індивідуальна онлайн-інструкція.
2. Вчитель вів інструктаж малих груп.
3. Самостійна та спільна практика.

1.4. Модель "Flex"

Коли ви заходите у школу, ви зауважуєте, що близько 100 учнів знаходяться у великій просторій кімнаті. Усі вони мають ноутбуки і сидять по різних куточках цього відкритого простору. Все виглядає дуже хаотично, але коли ви запитаете когось із учнів, що вони роблять, ви почуєте відповідь, що вони працюють над математичним завданням. Усі учні мають власний "плейлист" активностей, які вони повинні виконати за певний час. Коли щось не зрозуміло, вони можуть запитати свого однокласника чи вчителя. Водночас вчителі вивчають ситуацію і запрошують групи учнів у окремі кімнати для обговорення матеріалу, який діти не зрозуміли.

Особливості цієї моделі:

- основою цієї моделі є онлайн-навчання, кожен учень має свій ноутбук і можливість переходити в інші приміщення;
- учні можуть бути максимально мобільними і враховувати свої потреби для вивчення теми;
- кожен учень має індивідуальний графік і завдання;
- разом з учнями є два вчителі (або більше) – той, який може відповісти на будь-які запитання учнів під час роботи, та вчитель, який проводить персональні зустрічі з групами дітей, які щось не зрозуміли або ж мають труднощі з виконанням завдання.

1.5. Модель "Online Lab"

Уявіть собі школу, де учні закінчують більшу частину навчання в комп'ютерній лабораторії, розміщеній у традиційній школі. Навчання проводиться дистанційно в Інтернеті, але сам заклад контролюється неінструктивними працівниками, які стежать за поведінкою та відвідуванням. Це приклад моделі онлайн-лабораторії змішаного навчання. Ця модель дуже корисна в сільській місцевості та коли школи мають обмежені ресурси.

Модель онлайн-лабораторії змішаного навчання надає більшість, якщо не весь, вміст дистанційно. Це відрізняється від суто онлайн-навчання тим, що студенти збираються в традиційних умовах для доступу до комп'ютерів. Однак не потрібно вчителів віч-на-віч.

Онлайн-модель лабораторії дозволяє диференціювати освіту, оскільки кожен учень в класі може вчитися різним речам на різних рівнях. Крім того, варіанти курсу не обмежуються наявним викладацьким складом. Тому студенти мають доступ до майже необмежених варіантів навчання, а персонал обмежений тим, який необхідний для нагляду за навчальним закладом [4].

1.6. Модель "Self-blend"

У самостійній моделі навчання (або моделі самостійної суміші навчання) студенти проходять онлайн-курси на додаток до традиційного навчання віч-на-віч. Ця модель змішаного навчання зазвичай практикується серед американських старшокласників, і студенти дотримуються такого підходу, оскільки це дає їм свободу доповнювати те, що вони навчилися, і чого вони навчали на уроці. Модель самостійного змішування навчання, як правило, підходить студентам, які вирішили приймати онлайн-курси поза часом, який вони витрачають на традиційні курси. Він вирішує проблему курсів або предметів, які не надаються школою через різні причини, такі як недостатнє фінансування, відсутність інтересу студентів або брак часу. Студенти проходять онлайн-курси або в цегляному мінометному містечку, або в офісі. Не обов'язково мати вдома комп'ютери для учнів, які використовують самостійну модель навчання, оскільки вони мають доступ до Інтернету у власних шкільних комп'ютерних лабораторіях або можуть звернутися до місцевих бібліотек, щоб завершити свою роботу. Вони також можуть сходити до сусідніх кіберзалів.

Модель самостійного змішування часто використовується учнями старших класів, оскільки вимагає від учнів контролювати їхній навчальний процес та здійснювати ініціативи, з якими молодші школярі можуть мати труднощі в управлінні. Таким чином, це відрізняється від ротаційної моделі навчання, яка

досить часто застосовується в початкових школах, тоді як багато старшокласників відвідують онлайн додаткові курси з математики, англійської мови та соціальних досліджень на додаток до своїх курсів, пропонованих у школі.

Переваги

Існує ряд переваг моделі самостійної суміші навчання:

1. Студентам, які відмовились, дуже корисно повернути гроші, отримати кредит у коледжі або повернути кредити.

2. Це не тільки розширює навчання в класі, але й забезпечує доступ до онлайн-курсу, який інакше не пропонується в школі через різні причини, наприклад, брак коштів, часу та ресурсів.

3. Окрім доступу до онлайн-курсів, він також підвищує цифрову грамотність студентів, корисну для студентів, які входять до складу 21 століття.

4. Курси онлайн-курсів є більш доступними для студентів.

5. Він надає можливість пройти спеціалізовані курси, такі як підготовчі навички коледжу та життєві навички.

6. Його можна використовувати як для професійного розвитку, так і для навчальних цілей.

Недоліки

Основним недоліком моделі самозмішування є те, що студенти не отримуватимуть інструкцій віч-на-віч для цього конкретного курсу. Це також передбачає відсутність формальної структури для недисциплінованих та немотивованих студентів [5].

1.7. Лазерні стрілкові тренажери

Лазерні вкладиші для ПМ і ПЯ

Лазерні вкладиші призначені для відпрацювання техніки стрільби з навчального або бойової зброї.

Вкладиш є мініатюрним лазерний випромінювач, який вставляється в патронник навчального або бойової зброї (пістолет Макарова, пістолет Яригіна), не спотворюючи його зовнішній вигляд і вага.

Всі елементи вкладиша – лазерний діод, цифровий мікропроцесор, елементи живлення – розташовані всередині нього.

Лазерні вкладиші серії ЛТ-905 випускаються на пістолет Макарова і пістолет Яригіна.

Перед тренуванням лазерний вкладиш встановлюється в патронник навчального або бойової зброї згідно паспорта.

У момент пострілу від удару курка спрацьовує сигнал мікропроцесора вкладиша і в тому місці, куди був зроблений постріл, на мить з'являється яскрава червона точка. При перекручуванні затвора зброї тренажер не спрацьовує, що є одним з основних його переваг перед іншими моделями.

Лазерні вкладиші використовуються спільно з паперовими або світловідбиваючими мішенями, а також у складі лазерних тирів "Рубін":

- при стрільбі по звичайних паперових мішенях стрілок візуально контролює точність попадання лазерного променя;
- під час стрілянини в лазерному тирі лазерна точка фіксується лазерної камерою, підключеної до комп'ютера.

Лазерні вставки для зброї

Лазерні вставки призначені для навчання навичкам ведення прицільної, швидкісної і інтуїтивної стрільби з навчального або бойового пістолета. Головною перевагою вставки є можливість тренуватися зі своїм особистим зброєю і виконувати всі елементи стрілецького вправи: виїмка зброї з кобури, робота з магазином, стрілянина в русі.

Вставка являє собою мініатюрний лазерний випромінювач, який вставляється в стовбур пістолета, не спотворюючи його зовнішній вигляд і вага. Всі елементи випромінювача – лазерний модуль, звуковий датчик, електронна плата, елементи живлення – розташовані всередині вставки.

Лазерні вставки ЛТ-510 випускаються на пістолет Макарова, пістолет Яригіна, пістолет ІЖ-71.

Перед тренуванням лазерна вставка встановлюється на навчальний або бойову зброю згідно паспорта.

У момент пострілу від удару курка спрацьовує звуковий датчик тренажера і в тому місці, куди був зроблений постріл, на мить з'являється яскрава червона точка.

Лазерні вставки використовуються спільно з паперовими або світловідбиваючими мішенями, а також у складі інтерактивного лазерного тиру (ІЛТ) "Рубін":

- при стрільбі по звичайних паперових мішенях стрілок візуально контролює точність попадання лазерного променя;
- під час стрілянини в лазерному тирі лазерна точка фіксується камерою, підключеної до комп'ютера.

Для початкового навчання стрільбі використовуються лазерні вставки з видимим (червоним) випромінюванням, а для професійної підготовки – вставки з невидимим (інфрачервоним) випромінюванням.

Використання лазерних вставок з невидимим випромінюванням дозволяє стрільку виконувати наведення зброї по прицільної пристосуванню, повністю виключивши його націлювання по випромінюванню.

Лазерна зброя

Лазерна зброя призначена для навчання навичкам ведення прицільної, інтуїтивної і швидкісної стрільби.

Перевагою лазерної зброї є можливість виконувати всі основні елементи стрілецького вправи: витягти пістолет з кобури, зняти з запобіжника, зробити прицільовання і спуск з курка, стрілянина в русі і т.д.

Лазерна зброя являє собою масогабаритні макет зброї (пістолет Макарова, пістолет Яригіна, пістолет Глок або автомат Калашникова), в стовбур якого вбудований лазерний випромінювач, а в корпус – блок управління із елементами живлення [6].

Електронні тренажери "Рубін"

Електронні тренажери призначені для відпрацювання практичних прийомів і отримання міцних навичок стрільби з різного виду зброї по електронним мішеням з подальшою світловою і звуковою індикацією попадання.

Електронний тренажер "Рубін" включає в себе лазерний тренажер і електронну мішень. Як лазерного тренажера може використовуватися лазерна зброя або лазерна насадка. Використовувані електронні мішені: EM1, EM2, EM6, EM6в, EM4, ЕМБ1

Лазерна стрілянина ведеться з лазерного тренажера по електронній мішені, розташованій на відстані 1 – 25 м від вогневого рубежу. При попаданні лазерного променя в фоточутливу область електронної мішені відбувається короточасні звуковий і світловий сигнали.

Ефективність навчання стрільбі зростає при використанні декількох мішеней, рознесених по глибині і фронту.

Електронні мішені (EM1, EM2, EM6, EM6в) закріплюються на нерухомих, що з'являються, що піднімаються або гойдаються мішенях.

Для відпрацювання ситуацій вогневого контакту електронні мішені (EM4) розміщуються на одязі бійців.

Електронні мішені ЕМБ1 використовуються для тренування біатлоністів.

Електронні мішені EM1, EM2, EM6, EM6в, ЕМБ1 можуть поставлятися в комплекті з лазерними тренажерами з видимим (червоним) або невидимим (інфрачервоним) випромінюванням.

Електронна мішень EM4 поставляється тільки в комплекті з лазерним тренажером з невидимим (інфрачервоним) випромінюванням. Мішень EM4 може використовуватися зі спеціальним световідбиваючим елементом [7].

Інтерактивний стрілецький лазерний тренажер

Інтерактивний стрілецький тир – це апаратно-програмний комплекс. Призначення інтерактивного тиру:

1. Навчання стрільбі на парах захисту Вітчизни в старшій школі.
2. Відпрацювання навички прицільної стрільби для службовців спеціальних підрозділів, навчальних армійських центрів, охоронних структур і т.д.
3. Інтерактивні стрільби по різних мішенях, як в приватних електронних тирах, так і в домашніх умовах.
4. Розширені мультимедіа можливості.

Інтерактивний тир побудований на базі класичного відео проектора. Величезна картинка проекційного зображення може бути використана для демонстрації будь-якого типу контенту: від демонстрації слайд-шоу фотографій іменинника під час святкового банкету, до показу відео роликів, телевізійних передач, фільмів, трансляції футбольних матчів на великому екрані. Єдине що потрібно зробити, так це під'єднати до проектора наявний у Вас в залі програвач, медіа плеєр або супутниковий тюнер.

Стрільба здійснюється за допомогою лазерного цілевказівника (ЛЦВ), вбудованого в пістолет або автомат короткими світловими імпульсами по цілям на пристрої відображення (проекційний екран або інтерактивна дошка). До складу такого тиру обов'язково входить камера, яка фіксує позначки лазерного променя на проекційному екрані і моментально спрямовує цю інформацію в комп'ютер. Комп'ютер обчислює координати такого пострілу і відповідним чином змінює зображення на екрані (демонструє потрапляння або промах) (рис. 1.2).



Рис. 1.2. Принцип роботи лазерного стрілецького тренажеру

Устаткування для забезпечення інтерактивного тиру

Устаткування для забезпечення інтерактивного тиру показано на (рис. 1.3 – 1.5.)



Рис. 1.3. Проектор для забезпечення інтерактивного тиру



Рис. 1.4. Проекційний екран для забезпечення інтерактивного тиру



Рис. 1.5. Камера для лазерного тиру

Камера в лазерному інтерактивному тирі оснащена світлофільтром. Світлофільтр використовується для того, щоб в умовах динамічного зображення камера не реагувала і, відповідно, не подавала неправдиву інформацію в комп'ютер. Світлофільтр штучно затемнює відеоінформацію, таким чином камера реагує тільки на самі яскраві світлові імпульси, тобто на імпульси лазера при "стрільби". Світлофільтр повинен збігатися з кольором лазера (рис. 1.6).

Звичайно, якщо при відпрацюванні стрілецьких навичок використовуються тільки статичні зображення, світлофільтр на камері не потрібен. Однак такого роду вправи істотно збіднюють можливості тиру і знижують якість підготовки стрілка.



Рис. 1.6. Лазерний цілевказівник

Яскравість лазера значно вище яскравості зображення екрану, тому камера легко, оперативно, точно його зчитує і транслює інформацію в комп'ютер для прорахунку. У нашому тире ми використовуємо лазерні випромінювачі малої потужності (<5 мВт) видимого спектру. У червоному оптичному діапазоні довжина хвилі лазерного випромінювача 650 нм, діаметр точки на екрані до 6мм, таким чином точність розрахунку координати "пострілу" висока [8].

Макети зброї

Масогабаритні макети (ММГ) стрілецької зброї, зокрема, автоматів Калашникова, можуть бути пристосовані для використання в інтерактивному лазерному тирі.

Згідно з українським законодавством придбання ММГ не вимагає отримання спеціального дозволу, а самі ММГ відносять до предметів господарського-побутового призначення. Проте, такі макети є точною копією бойових аналогів і зберігають працездатність окремих функцій, тому в навчальних закладах слід звернути увагу на організацію зберігання і безпечне використання ММГ на парах.

ММГ автомата являє собою неактивний екземпляр бойової зброї, в якому заварений патронник, спиляно затвор і ударник, стовбур просвердлений і видалений викидач патрона, а також виконані інші конструктивні доробки, що

виключають можливість використання макета в якості працездатного вогнепальної зброї.

Для застосування ММГ автомата Калашникова в інтерактивному лазерному тирі на його дуло накручується насадка з вбудованим лазерним випромінювачем або під стовбур встановлюється навісний лазерний цілевказівник (рис. 1.7).



Рис. 1.7. ММГ автомата АК-47

Як і оригінал, макет АК-47 має ствол зі ствольною коробкою, затворну раму з затвором, поворотний механізм, ударно-спусковий механізм, цівка, трубку газу. Його пристрій дозволяє виробляти включення-виключення запобіжника, зведення і спуск ударно-спускового механізму, неповне розбирання і складання відповідно до інструкцією з стрілецької справі.

Ця модель автомата послужила основою для створення цілого сімейства бойового і цивільного стрілецької зброї різних калібрів, включаючи автомати АКМ і АК74 (і їх модифікації), кулемет РПК, карабіни і гладкоствольні рушниці "Сайга" та інші. АК і його модифікації є найпоширенішим стрілецькою зброєю в світі (рис. 1.8).



Рис. 1.8. ММГ пістолета

Для стрільби в лазерному тирі відмінно підходять пневматичні пістолети, оснащені навісним лазерним цілевказівником. Придбання на території України пневматичного пістолета не потрібно отримання спеціального дозволу. З метою безпечного використання все пневматичні пістолети перед продажем в обов'язковому порядку деактивуються і в подальшому не можуть застосовуватися для стрільби металевими кульками навіть при встановленому газовому балончику.

Для застосування в складі лазерного тиру кожен пневматичний пістолет додатково допрацьовується: в нього встановлюється натискувальна спусковий скобою кнопка, яка короткочасно "запалює" лазер [9].

Переваги використання пневматичних пістолетів в лазерному тирі:

- широкий вибір моделей, можна придбати і обладнати лазерним випромінювачем саме ту модель пістолета, яка найбільше подобається;
- простота установки навісного лазерного вказівника цілі, можна мати кілька моделей зброї і по черзі встановлювати на них один лазерний випромінювач;
- легкість і точність регулювання наведення лазерного променя.

Висновки до розділу

У першому розділі дипломного проекту було проаналізовано поняття моделі змішаного навчання та його види: модель "*Face-to-Face Driver*", модель "*Rotation*", модель "*Flex*", модель "*Online Lab*", модель "*Self-blend*", модель "*Online Driver*".

Як показує аналіз існуючих моделей змішаного навчання, ключовим моментом в їх проектуванні є вибір правильного співвідношення всіх методів навчання, що використовуються, та встановлення оптимального поєднання традиційних технологій з технологіями електронного навчання.

При проектуванні електронного інтерактивного навчального середовища спиралися на метод комплексного системно-діяльнісного моделювання процесу навчання, який передбачає одночасне поєднання моделювання змісту навчання, моделювання навчально-пізнавальної діяльності студентів та моделювання взаємодії учасників навчального процесу.

Моделювання змісту в цій системі є основним, особливості технічної дисципліни обумовлюють підпорядкування змістовного компонента навчання діяльнісному компоненту.

Також, було проаналізовано лазерні стрілкові тренажери та оснащення для їх успішної роботи.

Лазерний стрілецький тренажер побудований на базі класичного відео проектора.

Стрільба здійснюється за допомогою лазерного цілевказівника (ЛЦВ), вбудованого в пістолет або автомат короткими світловими імпульсами по цілям на пристрої відображення (проекційний екран або інтерактивна дошка). До складу такого тиру обов'язково входить камера, яка фіксує позначки лазерного променя на проекційному екрані і миттєво спрямовує цю інформацію в комп'ютер. Комп'ютер обчислює координати такого пострілу і відповідним чином змінює зображення на екрані (демонструє потрапляння або промах).

РОЗДІЛ 2

ПОРІВНЯННЯ ІГРОВИХ РУШІВ

2.1. Логіка вибору ігрового рушія

Щороку ігрова індустрія розвивається з неймовірною швидкістю. Зовсім недавно люди дивувалися 8-бітній іграшці, а зараз вже одягають *VR* системи і радіють повного занурення в гру. У багатьох початківців розробників виникає питання, який ігровий рушій краще. На даний момент існує величезна кількість ігрових рушіїв, від *GameMaker*, *Marmelade* до *Unity* і *Unreal Engine*, а пов'язано це з тим, що компаніям легше було написати свій рушій, який буде задовольняти всім вимогам, ніж використовувати чужий ігровий рушій, вивчати його інтерфейс, його мову програмування. Але зараз ігрові рушії еволюціонували, і багато розробників обирають той чи інший рушій, щоб зберегти купу часу і грошей.

Отже, щоб вивчати будь-який ігрової рушій, потрібно зрозуміти, чим буде займатися програма. Розважати друзів, розробляти інді-ігри для мобільних пристроїв або створити величезні AAA-проекти. Якщо ви вибрали перший варіант, то тут ви спокійно можете брати абсолютно будь-який ігровий рушій, наприклад: *Unity*, *Marmelade*, *GameMaker*. А якщо потрібен другий або третій варіант, тоді потрібно добре розібратися в кожному ігровому рушію.

Відразу варто відзначити, що всі документації по ігровим рушіям англійською мовою. Навіть той же *Unity* і *Unreal Engine* перевели тільки основи до своїх рушіїв, все інше англійською. І так, офіційної документації буде не вистачати, коли з'явиться перша проблема. Потрібно буде звертатися за допомогою на форуми, де спільнота зовсім російськомовна, або читати книжки, але тільки англійською [10].

2.2. *Unity3D*

Unity – ігровий рушій для розробки двомірних і тривимірних додатків та ігор під різні платформи, такі як: *Windows*, *Windows Phone*, *Android*, *Apple iOS*,

Linux, Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One. Також можна спробувати зробити гру під *Web Player*. *Unity* дуже тісно співпрацюють з *Oculus*, *Apple* і рядом інших компаній, тому бути *Unity Developer* дуже престижно.

До інтерфейсу *Unity* звикаєш за тиждень. Повністю освоюєш рушій приблизно за 0,5 – 1 год. Єдине, що заважає в *Unity*, так це дуже дивне наближення до об'єкта. Скролінг мишею дозволяє то сильно швидко віддалятися від об'єкта, то занадто повільно наближатися, що не дуже добре. Це, звичайно, не проблема, якщо клацати 2 рази по потрібному об'єкту в *Hierarchy*, але якщо потрібна точність фокусування, то, в *Unity* поки з цим проблеми. Взагалі *Unity* розбитий на кілька віконць: *Hierarchy*, де ви можете спостерігати назви об'єктів на сцені, групувати їх, *Scene, Inspector* (налаштування виділеного об'єкта), *Project*, де видно всі матеріали проекту, *ToolBar*, тобто меню з інструментами.

Розрахунки фізики виробляє *NVIDIA PhysX*.

Unity можна скачати абсолютно безкоштовно. Ця версія не буде демо-версією або на певний час, з її допомогою ви зможете створити свій власний проект. Тільки виставити платним наше творіння в маркет не можна, доведеться ставити рекламу в додаток або гру. Якщо доходи будуть більше 100.000 \$, то необхідно буде купити *PRO* версію за 3000 \$ або 75 \$ на місяць. З одного боку це грамотний крок, щоб розробники отримували нормальний дохід, а з іншого боку – це дивно. *Epic Games*, на відміну від *Unity*, заявили, що якщо вам щось подобається, *set it free*, тобто одним з найпотужніших рушіїв можна тепер користуватися безкоштовно. Єдине – ділитися 5 % від продажу копії гри. Це розумніше, ніж стільки проблем з *Unity*, так *Epic Games* не зупиняє дивувати, якщо ви потребуєте в засобах, то *Epic Games* надасть вам 5000 \$ для випуску вашого проекту в маси.

Unity 5 використовує *C#* і *JavaScript*. З цими мовами особливих проблем в *Unity* не зустрічає. Раніше *Unity 4* використовував *Boo*, але, ніхто ним не користувався, тому команда розробників прибрала цю мову.

На даний момент *Unity* має найбільше ком'юніті. У російському сегменті *Unity* є найпопулярнішим. За даними з інтернету, *Unity* використовують понад

50 % розробників. І все більше розробників залишають старі ігрові движки і переходять розробляти гри в *Unity*. Давайте будемо чесними, в *Unity* більше 60 % – це початківці програмісти, які не випустили ще жодної гри. 20 % людей повелися на рекламу в спільнотах, 10 % людей вивчили інтерфейс *Unity* і пішли робити черговий "урок" на *YouTube*, а ось решта 10 % – це ті самі *Unity Developers*, які по-справжньому верстають гри.

Unity має дуже потужний магазин *Unity Asset Store* (тільки через це магазина можна спокійно вибирати *Unity*) Тут є абсолютно все: від текстур і 3D моделей до готових проектів. Такий величезний асортимент ні у одного конкурента годі й шукати. Магазин *Asset Store* висуває *Unity* в лідери. Тепер про збірки:

Unity дуже швидко збирає сцени. Якщо вам потрібно буде швидко протести той чи інший момент в грі, а у вікні *Scene* це не завжди зручно, то сміливо збирайте, і через 1 – 2 хвилини у нас готова папка з *exe*-шніком всередині. Що ж стосується оптимізації, то в *Unity* не найкраща оптимізація під *PC* і *Mac*. Проекти, на кшталт *Mortal Kombat X*, *Tom Clancy: The Division* краще не пробувати створювати на *Unity*. Все-таки цей рушій створений для інді-проектів, а не для жорстоких стрілялок. Той же самий *Rust*, *The Forest*, якби вони були зроблені на *Unreal Engine*, то вони були б більш деталізовану з тією ж продуктивністю. Створювати фільми або кат-сцени краще в іншому редакторі, *Unity* жадає дуже потужний ПК.

У *Unity3d* є ряд проблем. По-перше, в *Unity 5* дивний мультиплеер. Наприклад, в *The Forest* досі не дороблений мультиплеер, а пов'язано це із тим, що налаштовується він дуже не просто. Є й хороший приклад – це *HeartStone*. По-друге, безкоштовна версія не дозволяє викласти проект під всі платформи, потрібно буде брати *Full Version* за 75 \$ на місяць або 3000 \$. По-третє, обмежений набір інструментів. Вам доведеться створювати деякі доповнення до рушію або докуповувати в *Asset Store* [11].

2.3. CryEngine 3

CryEngine 3 – ігровий рушій, розроблений компанією *Crytek*. *CryEngine*, на відміну від інших ігрових движків, націлений на створення ігор для *PC* і консолей. *Crytek* зробили дуже потужний рушій, який видає кращу графіку з усіх рушіїв. Тут є і динамічний світ, і затінення і затуманення в реальному часі, і управління рівнем деталізації ландшафту. Якщо вам потрібна зовнішня складова вашої гри, то *CryEngine* зробить це краще за всіх. Особи і персонажі в *CryEngine* виходять по-справжньому захоплюючими. А ось кілька прикладів ігор, зроблених на *CryEngine*: *Crysis*, *Far Cry*, *Warface*, *Star Citizen*, *Ryse: Son of Rome*, *Homefront: The Revolution*, *Evolve*, Проект армату, *Kingdom Come: Deliverance*.

Плюси:

- можливість прикрасити гру відмінною графікою;
- створення потужного звукового супроводу;
- найпростіший процес створення штучного інтелекту в сегменті;
- початківцю розробнику буде легко зробити користувальницький інтерфейс;
- розробки ігор для *PC*, *Xbox 360*, *PlayStation 3* з підтримкою *DirectX9* і *10*.

Мінуси:

- недбала технічна підтримка безкоштовної версії;
- підтримувана ОС тільки *Microsoft Windows*;
- відносно високий поріг входження.

2.4. Unreal Engine 4

Unreal Engine 4 – ігровий рушій, створений *Epic Games*. *Unreal Engine 4* – найпопулярніший ігровий рушій для створення фільмів і AAA-проектів. Даний рушій володіє високими графічними можливостями. З *Unreal Engine 4* можна розробляти гри під *PC*, *Mac*, консолі, *IOS*, *Android*, *Windows Phone*. На відміну від *Unity*, *UE4* має потужний інструмент для дизайну ігрових рівнів прямо в рушії, неймовірно зручну систему *Blueprint*, яка не має аналогів, красивий дизайн самого

движка, інтуїтивність у використанні і багато інших ласощів. З усіх ігрових движків, *Unreal Engine 4* є найбільш інноваційним. Він поєднує в собі високу продуктивність при білді, покращену графіку, проста мова програмування (не C #, але все-таки) і зручність у використанні. Є дуже потужне ком'юніті, що допомагає вирішити всі проблеми. *Epic Games* роздає *UE4* абсолютно безкоштовно, весь вихідний код відкрито, а все, чим ми повинні віддячити *Epic Games*, так це 5 % з продажу однієї копії. Якщо ви зібралися робити фільм, то платити і зовсім не потрібно. Команда *Epic* тісно співпрацює з розробниками *VR* технологій, а значить, що вже зараз можна створювати ігри під *PlayStation VR*, *Oculus* і т.д.

На сайті *Unreal Engine* є тонни документації. Також є розділ для тих, хто перейшов з *Unity3d*. У російському сегменті є безліч форумів, в яких вам з радістю готові допомогти. На *YouTube* є багато різних відео, де молоді геймдевери розкажуть вам про інтерфейс, ази скриптинга, правильному білді і т.д. Магазин в *Unreal Engine*, чесно кажучи, відстає від конкурентів. Якщо вам потрібні *3D* моделі або готові сцени, то краще пошукати їх в інтернеті, ніж купувати *Asset* в їх маркеті. По-перше, тут дуже мало контенту, по-друге, ціни тут нечувані, кусючі. *Unity Asset Store* поки лідирує, і *Epic Games* є до чого прагнути.

На *Unreal Engine 4* було зроблено величезну кількість AAA-проектів, таких як: *Mortal Kombat X*, *Dead Island 2*, *Smite*, *Paragon*, *Infinity Blade*.

2.5. HeroEngine

Даний рушій створений для *ММО* ігор, включає розробку клієнта і сервера. Для розробки використовується власну мову програмування (*HeroScript*), але можливо використовувати *C++*. Вартість ліцензії на рік близько 100 \$. Мінус або плюс, але даний рушій підтримує тільки операційну систему *Windows*.

Плюси:

- в наявності кілька карт для створення відкритого світу, є можливість їх "безшовного" з'єднання;
- зручний набір інструментів для моделювання карт;

- підходить для створення комплексних місій, крафтінга і збирання ресурсів;
- технічна підтримка здійснюється за допомогою сервісу *HeroCloud*, що вельми зручно.

Мінуси:

- скриптово рушій потужний, але незручний в управлінні;
- *HeroEngine* разом з сервісом підтримки клієнтів *HeroCloud* занадто дорого коштує і навряд чи буде доступний початківцям розробникам;
- високий поріг входження;
- можна створювати ігри будь-яких жанрів і типів, але *HeroEngine* робить тільки онлайн ігри.

2.6. *Rage Engine*

Rage Engine. *Rage* – ігровий рушій, розроблений *Rockstar*. *Rage* – *Rockstar Advanced Game Engine*. *Rage Engine* надає широкий спектр можливостей, включаючи в себе графічний, фізичний, звуковий і анімаційний движки, роботу з мережею і т.д. Багато років *Rockstar* домагався, щоб машина в грі поводитися так само, як і наяву. Зараз система враховує вагу машини, силу зчеплення шин з поверхнею і інші характеристики. Також вони приділили увагу безшовному відкритого світу, рендер місцевості відбувається дуже швидко: деякі об'єкти видаляються зі сцени, деякі додаються, а відбувається це все за частки секунди. Фізика людини і ШІ в *Rage Engine* вважаються дуже просунутими. *Rockstar* купила *Euphoria* – програмний компонент, який створює анімації "на льоту". У тій же самій *Grand Theft Auto 5* використовувався цей елемент і саме через нього Майкл, Тревор, Франклін вийшли такими живими [12].

2.7. Phaser

Жанрова спрямованість – браузерні 2D гри будь-яких жанрів.

Будь-яка платформа, де є веб-браузер з підтримкою *HTML5* (*Canvas*, *WebGL*). В них ліцензія: повністю безкоштовний для використання в комерційних проєктах; мова програмування: *JavaScript* і *TypeScript*; відкритий вихідний код: доступний на *github*; мультіплеер: можливий.

Переваги: не вимагає розкриття початкових кодів (як в *GPL*), малу вагу, висока швидкість роботи (багато в чому завдяки рендерингу *Pixi.js* і націленості на мобільні браузери), візуальний редактор *MightyEditor*, низький поріг входження, велика спільнота, велика кількість прикладів, що поставляються з фреймворком (понад 500), велика кількість створених ігор.

Недоліки: немає на даний момент навчальних матеріалів російською мовою.

Розробники рушія: *Richard Davey* і команда *Photon Storm*.

Phaser – це ігровий фреймворк для створення браузерних 2D ігор. *Phaser* ефективно і швидко обробляє 2D графіку, так як використовує бібліотеку *Pixi.js* для рендеринга, яка визнана однією з найкращих для відтворення 2D графіки, як для комп'ютерів, так і мобільних браузерів, які підтримують *HTML5* (*Canvas*, *WebGL*). Основна спрямованість *Phaser* – мобільні браузери, тому він дуже добре оптимізований для них і, як наслідок, на комп'ютерних браузерах *Phaser* працює дуже швидко.

Редактори. *Phaser* має візуальний редактор *MightyEditor*, який спрощує розробку. А так же має інструмент *Phaser SandBox*, який дозволяє експериментувати з будь-якими можливостями фреймворка або власним кодом і зберігати посилання для демонстрації будь-кому вашої роботи або для демонстрації проблеми.

Завантаження *Asset*. Фреймворк дозволяє одним рядком коду завантажувати *Asset*: малюнки, звуки, *spritesheet*-и, *tilemap*-и, вміст *JSON* і *XML* файлів.

Фізика. *Phaser* використовує 3 вбудованих фізичних движка: *Arcade Physics* (дуже легкий і відмінно підходить для дуже слабких пристроїв), *Ninja Physics*

(просунутий рушій з зав'язкою на тайли) і рушій *p2.js* з підтримкою *full-body* спрайтів.

Спрайт. Легко працювати з спрайтами: переміщення, поворот, масштабування, анімацію, зіткнення і т.д. Спрайт мають повну підтримку роботи з ними за допомогою мишки і тачпада: *drag & drop*, затиску (*snap*) і визначення *pixel perfect* кліка.

Групи. Об'єднання спрайтів в групи (паки) для повторного циклу використання (*recycling*). Це дає оптимізацію та зручність роботи. Наприклад, можна створити групу "*Bullets*" і однією командою перевірити зіткнення об'єкта цієї групи з об'єктом з групи "*Aliens*", якщо було зіткнення, то спрацює *callback* функція.

Анімації. *Phaser* підтримує класичні анімації на *spritesheet'ax*, а також анімацій з файлів: *Texture Packer*, *Flash CS6 / CC JSON* (як *Hash*, так і *Array* формати) і *Starling XML*.

Системи частинок. У *Phaser* вбудована система частинок під назвою *Arcade*, яка дозволяє легко створювати ефекти з частками, такі як: вибухи, дощ, дим, вогонь і т.д. Наприклад, ви можете закріпити еммітер до об'єкта, який рухається.

Камера. *Phaser* має можливості легко управляти камерою, переміщати її по ігровому Миру або вона може слідувати за спрайтом.

Введення. Для *Phaser'a* неважливо чи отримує гра введення з багатокнопковий мишки, сенсорного екрану або геймпада – робота з пристроями введення здійснюється одноманітно.

Звук. *Phaser* підтримує і *WebAudio* і застарілий *HTML Audio*. Він автоматично обробляє ситуацію, коли мобільний пристрій йде в режим блокування. *Phaser* бере на себе всю складну роботу зі звуком на мобільних пристроях.

Tilemap-и. *Phaser* завантажує тайли, малює їх і обробляє зіткнення з ними за допомогою всього декількох рядків коду. Він підтримує файли *CSV* і файли з *Tiled Map Editor*, обробляючи тайли з шарів цього редактора. Він має потужні

функції для роботи з тайловими картами в реальному часі: поміняти (*swap*), замінити (*replace*), видалити, додати, оновити і т.д.

Масштабування гри під різні пристрої. *Phaser* має вбудований інструмент (під назвою *Scale Manager*), який дозволяють адаптувати вашу гру під будь-який дозвіл. Він контролює відношення сторін екрану, максимальна і мінімальна масштабування і повноекранний режим.

Система плагінів. Розробники *Phaser*-а намагаються, щоб *Phaser* мав всі необхідні вбудовані можливості, але при цьому був легко розширюваним за рахунок плагінів, які розробники ігор створюють для своїх проєктів. Як правило, розробники ігор роблять доступними свої плагіни для спільноти *Phaser*.

Мобільні браузері. *Phaser* спочатку розроблявся для мобільних браузерів, тому він має високу ступінь оптимізації для мобільних браузерів. Очевидно, що для комп'ютерних браузерів він теж швидкий. Але на відміну від багатьох інших фреймворків *Phaser* продовжує бути націленим на мобільні браузері і якщо якась із функцій не працює на мобільному браузері досить швидко, то вона не додається в ядро *Phaser*.

Протестований в бою. Так як на *Phaser*-и створено сотні комерційних ігор, в які грають мільйони людей, то за час його існування багато баги були вичищені. Фреймворк продовжує активно розвиватися і якщо знаходяться помилки або недоробки, то вони оперативно виправляються [13].

2.8. Construct 2

Construct 2 – це заснований на *HTML5* конструктор *2D* ігор, розроблений компанією *Scirra*. Конструктор спрямовано в першу чергу на людей, які не розуміються в програмуванні, дозволяючи швидко створювати ігри способом *Drag-and-drop* з використанням візуального редактора та логічної системи, заснованої на принципі поведінки та реакції. *Construct 2* є прямим нащадком попередньої версії програми, *Construct Classic*.

Як і в *Construct Classic*, основним методом програмування ігор та додатків у *Construct 2* є використання "листів подій", що схожі на файли рушія, які

використовуються у мовах програмування. Коли виконується умова, задана користувачем в листі подій, слід за нею виконується дія чи функція.

Особливістю *Construct 2* в порівнянні з іншими конструкторами є так названі «поведінки». Поведінка – це заздалегідь заготовлений набір (шаблон) властивостей об'єкту. Поведінки потрібні для прискорення процесу розробки гри, коли користувач не задає всі властивості сам в листі подій, а просто користується необхідним шаблоном. Прикладом поведінки є *8 direction*, яка дозволяє переміщувати об'єкт у восьми напрямках за допомогою клавіш.

Розробники *Construct 2* забезпечують та навіть заохочують створення плагінів від сторонніх розробників. Так, на офіційному сайті можна знайти поради та пари з написання та налаштування плагінів для коректної роботи. Всі плагіни *Construct 2* написано на *Javascript*.

Підтримувані платформи: *HTML5*

Основні експортні платформи побудовано на *HTML5*. Це означає підтримку через *Google Chrome*, *Firefox*, *Internet Explorer 9+*, *Safari 6+* і *Opera 15+* серед настільних браузерів, а також підтримкою *Safari* в *iOS 6+*, *Chrome* та *Firefox* для *Android*, *Windows Phone 8+*, *BlackBerry 10+* і *Tizen*.

Крім того, *Construct 2* здатен експортувати в деякі інтернет-майданчики та магазини, серед яких *Facebook*, інтернет-магазин *Chrome*, *Amazon Appstore*, власний майданчик *Scirra Arcade* та *Kongregate*.

Construct 2 забезпечує прямий експорт до *Windows*, *OS X*, 32-бітні та 64-бітні версії *Linux*, за допомогою *Node-Webkit*. Експорт робиться задля того, щоб користуватися деякими особливостями, які не підтримуються *HTML5*. 23 жовтня 2012 року *Scirra* також оголосила про повну підтримку нового інтерфейсу *Windows 8* під назвою *Metro*, в тому числі покупки в додатку, сенсорний ввід, керування плитками, акселерометр тощо.

Також *Construct 2* підтримує експорт на рідні мобільні платформи *iOS* та *Android*, але за рахунок використання сторонніх технологій, таких як *CocoonJS*, *PhoneGap* та *Crosswalk*. На жаль, через відсутність власного експортера погано

оптимізовані додатки можуть суттєво сповільнювати швидкодію на мобільних платформах.

В *iOS* ви можете запускати створені гри з *Construct 2* в веб-додатку, без підключення до Інтернету. Даний спосіб дозволяє експортувати гру в *html*. Після цього ви завантажуєте гру на сайт, заходьте з мобільного пристрою *iOS*, і зберігаєте гру на робочий стіл.

20 січня 2014 року *Scirra* оголосила, що в *Construct 2* буде забезпечено підтримку ігрової консолі *Wii U*. У тому ж році було випущено плагін, який дозволяє робити додатки сумісними з *Nintendo Web Framework*. Жодної інформації стосовно підтримки інших консолей не оголошувалося.

Існує декілька видів ліцензії *Construct 2*:

Personal – для окремих інди-розробників, з можливістю обмеженого комерційного використання (заробивши на грі більше 5.000 \$, користувач зобов'язаний зробити апгрейд ліцензії до *Business*). Вартість ліцензії складає 129,99 \$.

Business – для професійних ігрових студій з необмеженою можливістю комерційного використання. Коштує 429.99 \$

Free – безплатна демонстраційна версія рушія з суттєвими обмеженнями. Комерційне використання заборонене.

Site – спеціальна версія для навчальних закладів вартістю 359.99 \$ на рік [14].

2.9. Corona SDK

Corona – це вільна міжплатформна платформа, яка ідеально підходить для створення ігор та додатків для мобільних пристроїв та настільних систем. Використовуючи потужну, але просту для засвоєння мову скриптів *Lua*, понад 1 000 вбудованих *API*, широкий вибір плагінів та розширення *Corona Native (C / C ++ / Obj-C / Java)*, ви зможете реалізувати свої мрії про додаток. Ви навіть можете миттєво відстежувати проекти на кількох пристроях за допомогою *Live Builds*.

Використовуючи *Corona*, ви можете публікувати на всіх основних платформах з однієї бази коду, включаючи *Apple iPhone* та *iPad*, телефони та планшети *Android*, *Firefox Amazon*, *Mac Desktop*, *Windows Desktop* і навіть підключені телевізори, такі як *Apple TV*, *Fire TV* та *Android* телевізор. Ніяких окремих проектів чи переписувань не потрібно!

Протягом 8 років Корона перетворилася в зрілу, випробувану на бій рамку. Понад 1 000 *API* надають доступ до спрайтових анімацій, аудіо та музики, фізики *Box2D*, вирівнювання об'єктів, вдосконалених графічних фільтрів, випромінювачів частинок, мереж, системного управління, управління текстурою, рідних елементів, даних та багато іншого.

Corona використовує *Lua*, потужну і просту в навчанні мову сценаріїв, підкріплену обширною документацією. *Lua* розроблений, щоб бути легким, швидким, але при цьому потужним. Він використовується в основних назвах, таких як *Civilization*, *Warcraft* та незліченних іграх з Інді. Дизайнери також люблять *Lua*, тому що вона дуже доступна і вона пропонує простий покращуючи синтаксис – якщо ви ніколи раніше не кодували, *Lua* і *Corona* – ідеальне місце для початку.

У поєднанні з *IDE* або текстовим редактором, який вам уже комфортний, інтерактивний симулятор *Corona* негайно відповість на зміни вашого коду, надаючи попередній перегляд у реальному часі того, як буде виглядати ваша програма – і як вона буде вести себе – на фактичних пристроях. Все, від графіки до анімації, імітації фізики до прекрасних ефектів частинок, *Corona Simulator* упорядковує ваш робочий процес і дозволяє миттєво бачити зміни.

Коли ви будете готові перейти за межі *Corona Simulator* і протестувати свою програму на фактичних пристроях, просто встановіть прапорець на екрані збірки і запуститься наша система *Live Build*. Тоді, як і миттєвий оновлення *Corona Simulator*, будь-який пристрій на вашому пристрої локальна мережа, на якій працює *Live Build* програми, миттєво оновиться, заощаджуючи час частих збірок та повторних встановлень.

Від швидкоплинної графіки *OpenGL* до легкої сценарної потужності *Lua*, *Corona* оптимізована для продуктивності на кожному рівні ядра. Завдяки *Corona* ваші програми автоматично збираються під час створення, упорядковуються для продуктивності та стабільності.

Ядро *Corona* вже багате функціями, але якщо вам потрібно більше, на ринку Корони є багато плагінів, які ще більше розширюють роботу *Corona*. Все, що стосується реклами в додатках, аналітики, засобів масової інформації, апаратних можливостей та іншого, знаходиться у вас в межах досяжності через вибір майже 200 плагінів, і це число продовжує зростати [15].

2.10. GameSalad

GameSalad Creator

Жанрова направленість: просто *Web 2D*-ігри.

Мова програмування: перетягування, без програмування.

Платформи: *Mac OS X*, *Web*, *iOS (iPod Touch, iPhone, iPad)*, *Android*, *Kindle*, *Nook*, *Windows*.

Достоїнства: простір в освоєнні, можливість публікації ігор на сайті конструктора, *Web*.

Недоліки: прості ігри.

GameSalad – це веб-конструктор, який надає вам можливість створити ваші ігри на *Mac OS X* для мереж та *iOS*, і опублікував їх на сторінці конструкторів порталу. *GameSalad* розробляє для тих, хто любить відкривати ігри візуально.

У вас є цікава ідея за іграми, але ви не використовуєте рутину програмування та є бажання сконцентруватись виключно на геймплеї ігри. З *GameSalad* немає ніякої необхідності програмувати, немає навіть скриптових команд. Кожний елемент гри створює виключає візуально та інтуїтивно зрозуміло. З функцією перетягування і перетягування вам потрібно буде переставляти потрібні функції, підключати їх і компонувати. І в геймплейних іграх, що постійно існують від вас, ви можете зробити свою іграшку особливою благодійною допомогою своїх компонентів: графіки та звуків. З конструктором

ігор *GameSalad* розробляється і перетворюється в дуже простому і цікавому процесі.

Редагування гри відбувається в режимі реального часу. *GameSalad* можна назвати онлайн інструментом для створення прототипів ігор. Вам не потрібно ждати, вибираєте іграшку для тестування. *GameSalad* дозволяє вам повідомити про невідомості, редагувати ігри прямо в той момент, коли вони запущені. Ви можете економити час, багатократно клонувати елементи гри. Ви можете видаляти, копіювати та вставляти об'єкти. У грі є можливість використовувати реалістичну фізику.

GameSalad можливо вам від одного кліку публікувати ваші ігри – у *Facebook*, *Twitter* та *MySpace*. Завдяки чому ваші ігри зможуть пошукати батьків, друзів та кого угодно. Ви можете вимкнути свої ігри на своєму блозі або на домашній сторінці.

GameSalad є потужною системою розробки ігор для безлічі людей (це 99 % з усієї маси бажаючих зробити свою іграшку), які не вміють програмувати, та мріють створити свою іграшку. Цей конструктор може називатись легким гейммейкером для *Mac OS X*.

На цьому конструкторі ігор можна створити прості ігри, наприклад, де ви граєте в ролі метелика, які збирають мед, уникаючи бджіл. Він не вимагає знань в програмуванні, дуже простий і зрозумілий. Підключення всіх елементів гри (ефекти, персонаж, противники та ін.) Відбувається за системою перетягування.

У липні 2012 року вийшла нова відкрита бета-версія конструктора під ОС *Windows*.

Про-акаунт коштує 499 \$ за рік [16].

2.11. *Lumberyard*

Компанія *Amazon* розробила новий ігровий рух, який отримав назву *Lumberyard*. Перше, що потрібно знати, – все бажаючі можуть використовувати його досконало безкоштовно.

По словам розробника, розробка дозволила створити ігрові проекти ААА-класу для настільних ПК, ігрових консолей, мобільних пристроїв і навіть платформи віртуальної реальності. Для сканування і використання рушію Дроворуб не потрібно платити ні копійки, ні додаткові сервіси для розробки, будуть платними. Саме так компанія *Amazon* збирається заробляти на нових продуктах.

Як підтверджується, ігровий рух *Amazon Lumberyard* «тісно інтегрується» не лише з обласної платформи *Amazon Web Services*, але і не обслуговує трансляцію відеоігри *Twitch*, який компанія купила за 970 мільйонів доларів у 2014 році.

Захід *Lumberyard* – типовий для компанії крок: максимальна диверсифікація бізнесу та «підживлення» сторонніх розробників на власних сервісах. Розвиваючі можуть абсолютно безкоштовно використовувати рушій *Lumberyard* для створення ігор, але як тільки вони захоплять додати підтримку мультиплеєрів або інших інтернет-служб, підключивши необхідність придбати "серверний час" в *Amazon*. Тому вони захоплюють представити максимально простою та зручною організацією організації прямих потокових трансляцій та дивиться у сторону *Twitch*.

І в той час *Amazon* спеціально підкреслює необхідність необхідного оформлення підписки на власних обласних послугах (дійсно, для одночасного використання ігрових або компактних, у яких є власні сервери, необхідні для їх наявності), компанія не вирішує використовувати конкурентоспроможні служби інших служб компанії *Google* або *Microsoft*. Це повний розумний і логічний хід, який забезпечує Амазонку величезним впливом. У випадку створення очевидних успішних ігор на рухомій *Lumberyard* вручає *League of Legends* з 67 мільйонів щомісячних гравців, її успіх у багатьох стане загрожує від *Amazon*.

Ігровий рушій *Lumberyard* дозволяє створити ігри для *Windows*, *Xbox One* та *PS4*, підтримуючи *Mac*, *Linux*, *iOS*, *Android* та *Oculus Rift*, вони додають «скоро». Її технологічна візуалізація основана на наборі популярних та потужних інструментів, розроблених, ліцензованих у *CryEngine*, а також пропонують всі

традиційні навороти, доступні з підтримкою ефективні елементи частин та технологій моделювання потоків, які живуть в реальному часі, і закінчуючи інструментарієм для створення рослин з безпрецедентною технологією.

2.12. *LibGDX*

Рушій написаний на *Java*. Ну і писати ігри майже не пропонується на *Java*. Хоча підходить будь-яка мова, яку можна запустити на *Java*-машині. Вмільці пишуть на *Groovy*, *Scala* та *Python*. У мережах можна знайти розміри, як це зробити.

На офіційному сайті є галерея, запропонована, розроблена на цьому русі. Якщо ви зробите свою іграшку або додаток, ви можете додати її в цей каталог. Вдасться безкоштовна реклама. На даний момент у грі доступні сотні пропозиції. Це говорить про популярність руху і про його конкретній зрілості.

LibGDX – не для програмування дрібниць. Якщо ви хочете швидко зробити іграшку на вечір, і до цього не працювали з тими рухами – не отримав. Нічого у вас немає досвідченої програмізації, особливо на *Java*, щоб *LibGDX* дав вам необхідну свободу для творчості. Якщо ви хочете дізнатися більше про це – читайте далі, розкажіть про цікаві особливості руху.

Рушій кросплатформенний. Працює на *Windows*, *Linux*, *Mac OS X*, *Android*. З деякими обмеженнями вводиться на *iOS* та *HTML5*.

Вшита бібліотека, що має математику. Швидко почитайте синус-косинус за таблицями не проблема для цього руху.

Для *2D* – підтримка графа сцен із трансформації (повороти, масштабування тощо). Дуже зручна штука для всіх ефектів.

Підтримка великої кількості сторонніх інструментів. Рушій працює з редактором карти Кахельна плитка, підтримує фізичний рушій *box2d*, вміє малювати *.ttf* шрифти і не тільки.

LibGDX – кросплатформенний рушій. У всіх випадках є конкретні відмінності, коли ви пишете під Андроїдом, а коли ви пишете під *Linux*. Якийсь код буде загальним, а який-небудь специфічно для *Android*. Наприклад, робота з

сенсором, акселерометром і т.д. У вас є один головний проект (*core*). У ній найголовніший код гри. А для кожної платформи створюється окремий проект. Наприклад, ви робите іграшку, яка називається *coolgame*. Ваша гра буде під ПК і для андроїда. Тоді у вас будуть три проекту – *coolgame-core*, *coolgame-desktop*, *coolgame-android*. Якщо ваша гра не використовує чотири-особливих на конкретних платформах (акселерометр, камера та ін.), то ви не будете дописувати код в андроїд і на робочому столі приватних. Якщо вам потрібна робота з конкретною апаратною методикою, ви напишете цей код у проекті для потрібних платформ.

У офіційній вікі, наприклад, така система підключається реклама *AdMob*, а за фактом ви можете таким чином отримати доступ до будь-яких апаратно-залежних частин потрібної платформи.

Граф сцен – не нове поняття для ігрових рухів. Ігровий екран показує як певний сценарій (Сцена), де розміщуються та взаємодіють актори (Актор). Акторами можна керувати, змінювати їх положення, масштабувати і т.д. Причому деякі актори можуть містити себе в інших акторів – так називають акторів-груп або батьківських акторів. І коли ми міняємо позиції груп-батьків, я змінюю і положення всіх акторів-дітей.

Це дуже зручна штука. Наприклад, ви робите діалогове вікно. Там є заголовок, кілька кнопок, текст, деякі малюнки і т.д. Ви робите одну родительську групу, і ви вже вміщуєте всі ці заголовки, кнопки та елементи виступу. Причому, поміщаєте відносно тієї родительської групи. Потом, коли потрібно вказати діалогове віконце, ви вказуєте потрібну позицію для цього віконця. Усі кнопки, надписи та пробірки в цьому навколишньому будуть розміщені правильно відносно групи-батька.

LibGDX має повноцінну підтримку такої справи. Простий актор називається Актором. Група називається *Group*. Ви можете комбінувати акторів як завгодно, вкладати їх у другу. Якщо надійно рухатися, добре оптимізований, він буде перераховувати координати і робити прості ресурси операцій, коли потрібно, коли не потрібно. Це також приємно.

UI – це велика для ігор. Часто в рухах або немає бібліотеки інтерфейсу користувача, або ж вона жахлива. Згадаємо хоча б від *Unity3D*. Первісні версії, до 4.6, користувальницький інтерфейс створили, який ніхто не використовував для серйозних проектів. Використовували сторонні бібліотеки, такі як *NGui*.

У *LibGDX* на цій планеті все добре. Є досить багатий набір базових UI-компонентів. Це кнопки, текстові мітки, просто картинки, діалогові вікна, панелі прокрутки, дерева, чекбокси, списки та прочі. Усі елементи можна настроїти як з коду (наприклад, вибрати шрифт, фонові зображення та ін.), так і стилізувати за допомогою *json*-файлів. Для тих, хто знає з веб-розробником, це нагадає ситуацію з *CSS*.

Так, у *LibGDX* ви створили *json*-стилі для елементів, де описуєте зовнішній вигляд кожного чоловіка. Потім, в коді, ви використовуєте цей *json*-стиль до елементу. Знову ж таки, це зручно для великих проектів. Відкриття даних від коду.

Який би рушій не був хорошим, завжди є моменти, коли не хватає базової функціональності. Потрібні розширення. *LibGDX* пропонує кілька розширених "з коробки". Це робота з фізикою (*Box2D*), робота з *.ttf* шрифтами, підтримка компонентної системи (*ECS*). Сторонні же розробники зробили розширення для зручної роботи з соц-мережами (*Facebook*, *Twitter*), ненапруженою роботою із систем внутрішніх грошей покупок. Є редактор UI – *Overlap2D*. Якщо ви шукаєте по *GitHub*, можна знайти багато цікавого.

Найперше – вам потрібна встановлена *JDK* (*Java*-машина та *Java*-компілятор), не нижче версії 1.7. Ви можете завантажити все, що можна знайти, і завантажити безкоштовно на сайті *Oracle*.

Далі з офіційного сайту вам потрібно скачати генератор проектів. Це *java*-додавання в *.jar*-форматі. Запустіть його. Ви задаєте кілька питань. Куди генерувати проект, які розширення використовувати. Запишіть ці поля, натисніть кнопку "Генерувати проект", і ви отримаєте у вказаній вами папці набір *Gradle*-проекти. Далі ви можете відкрити любимі ваші срібло розробки (*Eclipse*,

Android Studio або щось, що інше, що підтримує *Gradle*), імпортуйте проект і працюйте далі з ним, як звичайно на *Java*-проектах [17].

Висновки до розділу

У другому розділі дипломного проекту розглянуто декілька найбільш популярних ігрових рушіїв.

Було обрано ігровий рушій *Unity*, бо тут використовується компонентно-орієнтований підхід, в рамках якого розробник створює об'єкти і до них додає різні компоненти. Завдяки зручному *Drag & Drop* інтерфейсу і функціональним графічному редактору рушія дозволяє розставляти об'єкти в реальному часі і відразу ж тестувати, який вийшов результат.

Сильна сторона *Unity* – підтримка величезної кількості платформ, технологій, *API*. Створені на движку гри можна легко перенести між ОС *Windows*, *Linux*, *Android*, *iOS*, на консолі сімейств *PlayStation*, *Xbox*, *Nintendo*, на *VR*- і *AR*-пристрої. *Unity* підтримує *DirectX* і *OpenGL*,

На основі аналізу ігрових рушіїв, було вирішено для розробки навчального середовища застосовувати ігровий рушія *Unity*.

РОЗДІЛ 3

РОЗРОБКА ІНТЕРАКТИВНОГО НАВЧАЛЬНОГО СЕРЕДОВИЩА ЛАЗЕРНОГО СТРІЛЕЦЬКОГО ТРЕНАЖЕРУ НА ОСНОВІ ІГРОВОГО РУШІЯ

3.1. Розробка інтерактивного навчального середовища лазерного стрілецького тренажеру на основі ігрового рушія

Для створення інтерактивного навчального середовища лазерного стрілецького тренажеру на основі ігрового рушія було обрано програмне забезпечення *Unity Hub*. *Unity Hub* – безкоштовний сервіс компанії *Unity Technologies*, який дозволяє створювати власні ігри.

На (рис. 3.1) продемонстрований інтерфейс цієї програми, який дозволяє відразу заходити в створені проекти.

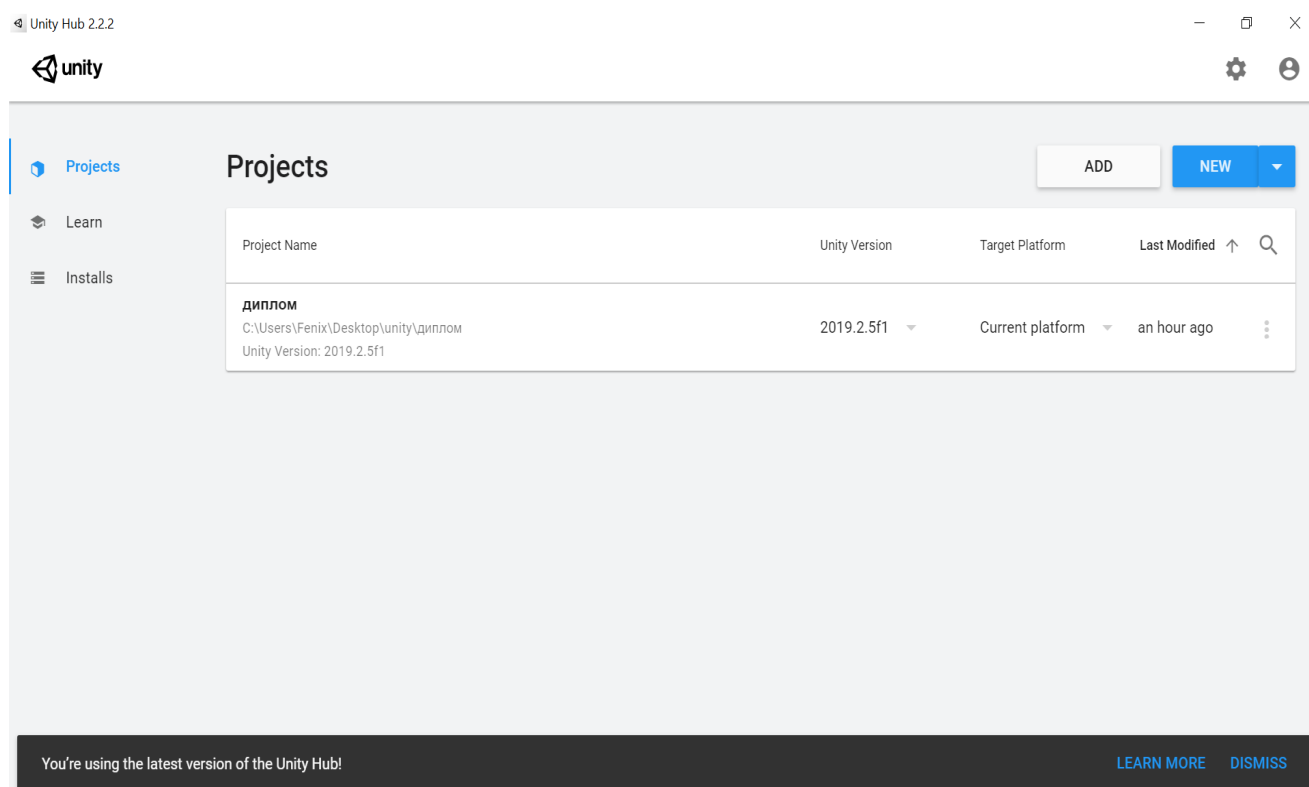


Рис. 3.1. Загальний вигляд програмного забезпечення *Unity Hub*

Для наповнення програми, було обрано 10 питань з декількома варіантами відповідей.

Далі йде перелік питань з відповідями, де правильні варіанти підкреслені.

Що таке прямий постріл?

1. Це постріл при якому висота траєкторії не піднімається над лінією прицілювання вище цілі на всій своїй довжині (рис. 3.2).
2. Це постріл при якому куля піднімається вище цілі.
3. Це постріл при якому куля знижується нижче цілі.

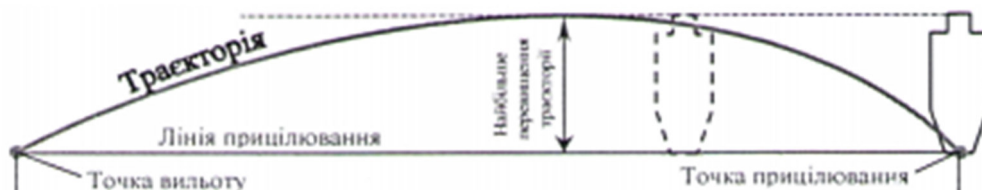


Рис. 3.2. Схема прямого пострілу

Траєкторія польоту кулі це:

1. Балістична крива лінія, яку описує центр тяжіння кулі у польоті (рис. 3.3).
2. Зміщення кулі під час польоту.
3. Зміщення кулі під час польоту.
4. Пряма лінія, котра являється продовженням осі каналу ствола у момент вильоту кулі.

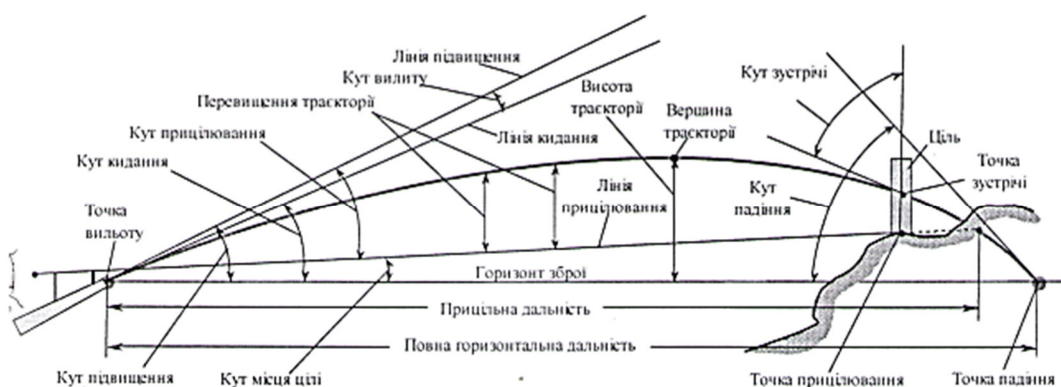


Рис. 3.3. Питання 2

Калібр автомата АК-74 (рис. 3.4)

1. 5,45 мм.
2. 9 мм.
3. 5,62 мм.
4. 7,62 мм.



Рис. 3.4. Питання 3

Оберіть мішень №4

1 (рис. 3.5).

2 (рис. 3.6).

3 (рис. 3.7).

4 (рис. 3.8).

Оберіть мішень №8

1 (рис. 3.5).

2 (рис. 3.6).

3 (рис. 3.7).

4 (рис. 3.8).



Рис. 3.5. Питання 4-5



Рис. 3.6. Питання 4-5



Рис. 3.7. Питання 4-5



Рис. 3.8. Питання 4-5

Кількість нарізів в каналі ствола пістолета ПМ(рис. 3.9)

- а) 5.
- б) 3.
- в) 6.
- г) 4.



Рис. 3.9. Питання 6

Вивірочна мішень гранатомета

- 1 (рис. 3.10).
- 2 (рис. 3.11).
- 3 (рис. 3.12).
- 4 (рис. 3.13).

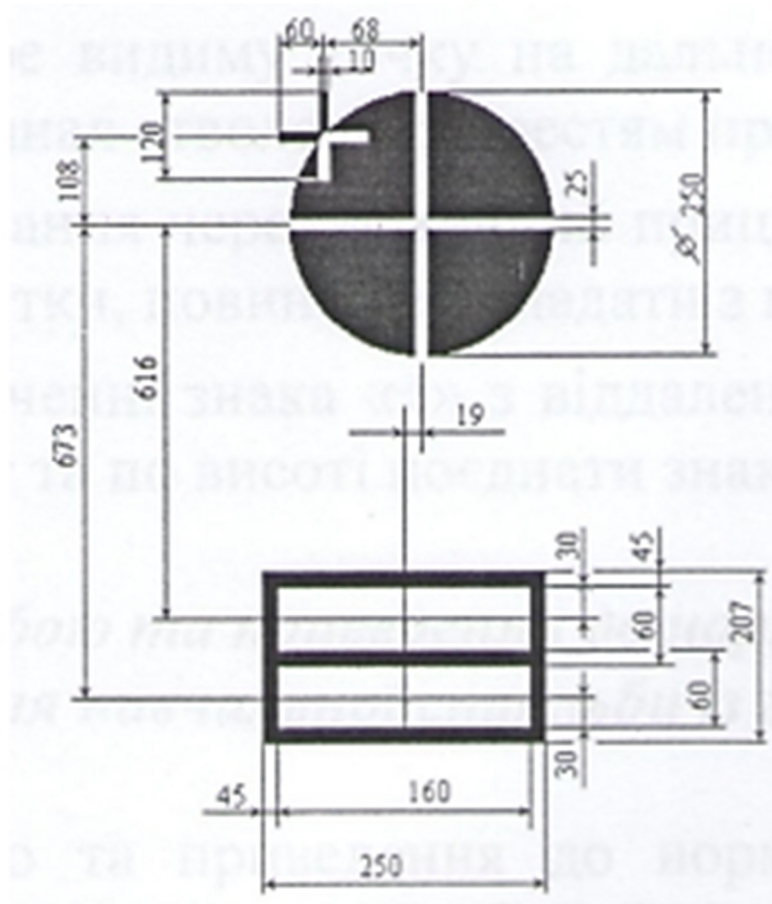


Рис. 3.10. Питання 7

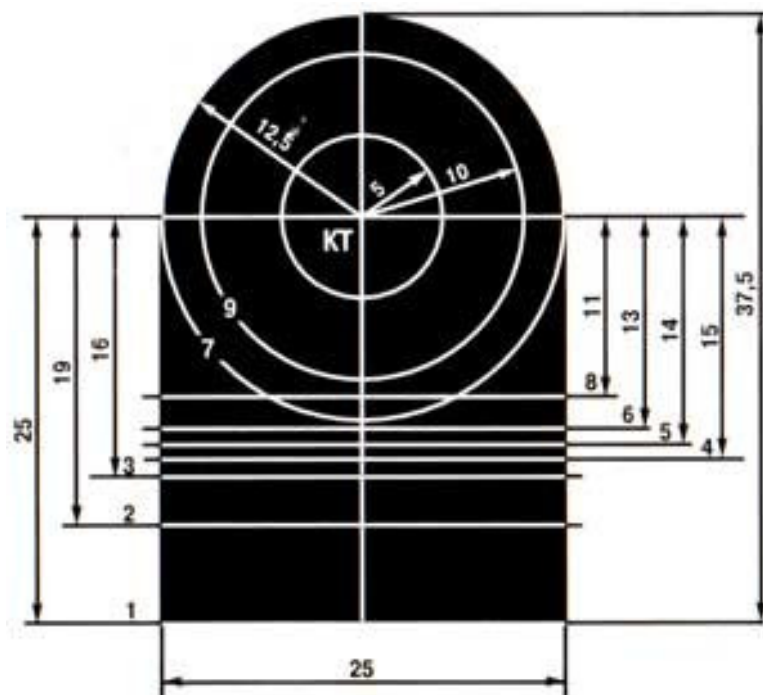


Рис. 3.11. Питання 7

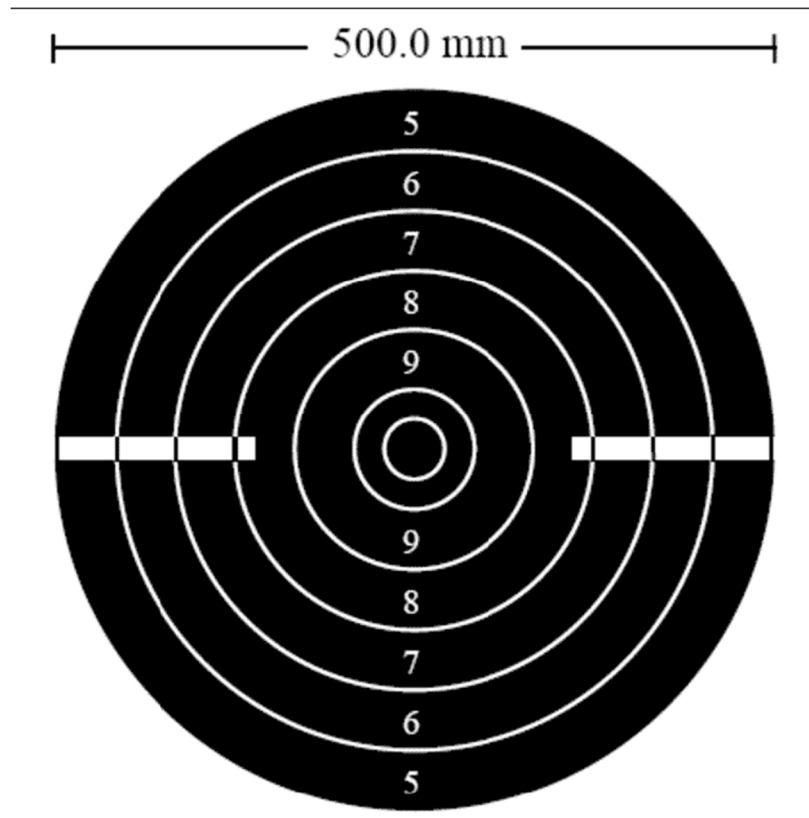


Рис. 3.12. Питання 7

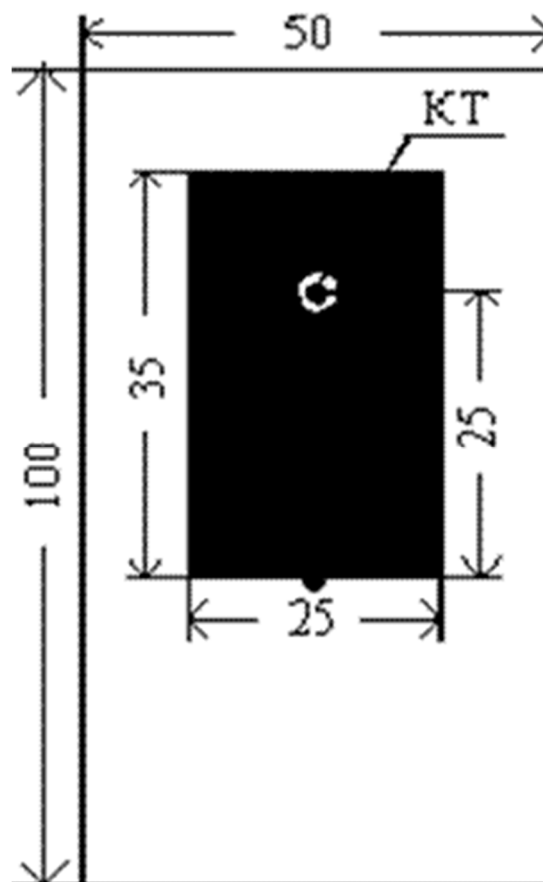


Рис. 3.13. Питання 7

Вкажіть пороховий заряд на РПГ-7 (рис. 3.14)

- а) 1.
- б) 2.
- в) 3.
- г) 4.



Рис. 3.14. Питання 8

Вкажіть гранату ПГ-7ВМ

- 1 (рис. 3.15).
- 2 (рис. 3.16).
- 3 (рис. 3.17).
- 4 (рис. 3.18).



Рис. 3.15. Питання 9



Рис. 3.16. Питання 9



Рис. 3.17. Питання 9



Рис. 3.18. Питання 9

Вкажіть сітку оптичного прицілу ПГО-7В (рис. 3.19 – 3.22)

1 (рис. 3.19).

2 (рис. 3.20).

3 (рис. 3.21).

4 (рис. 3.22).

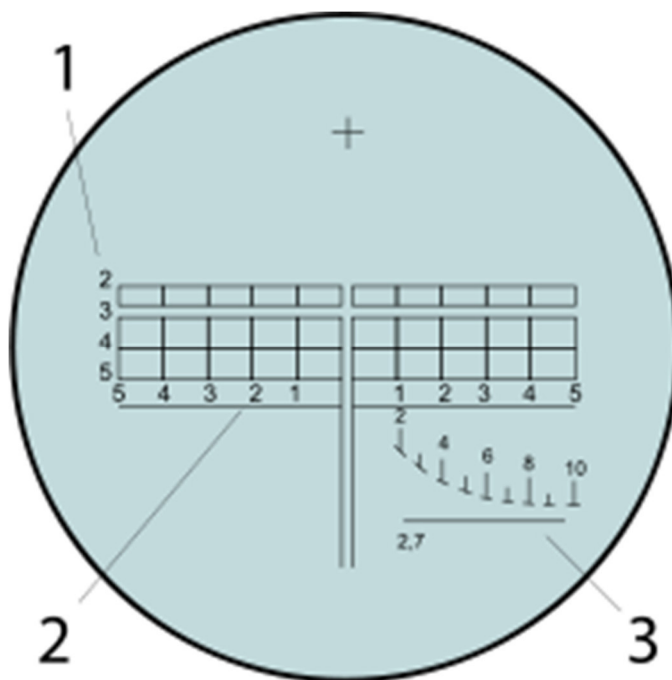


Рис. 3.19. Питання 10

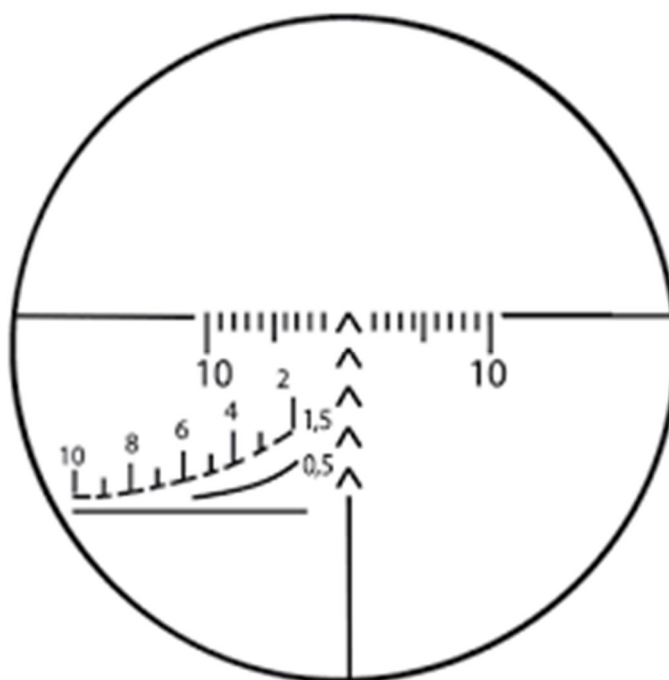


Рис. 3.20. Питання 10

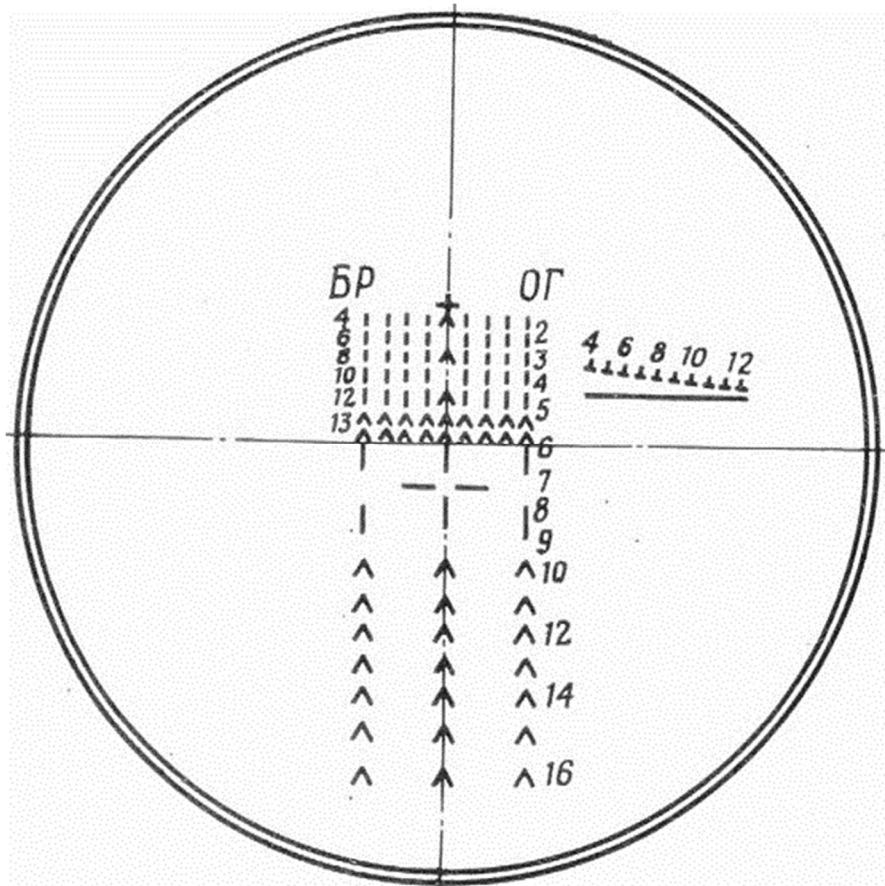


Рис. 3.21. Питання 10

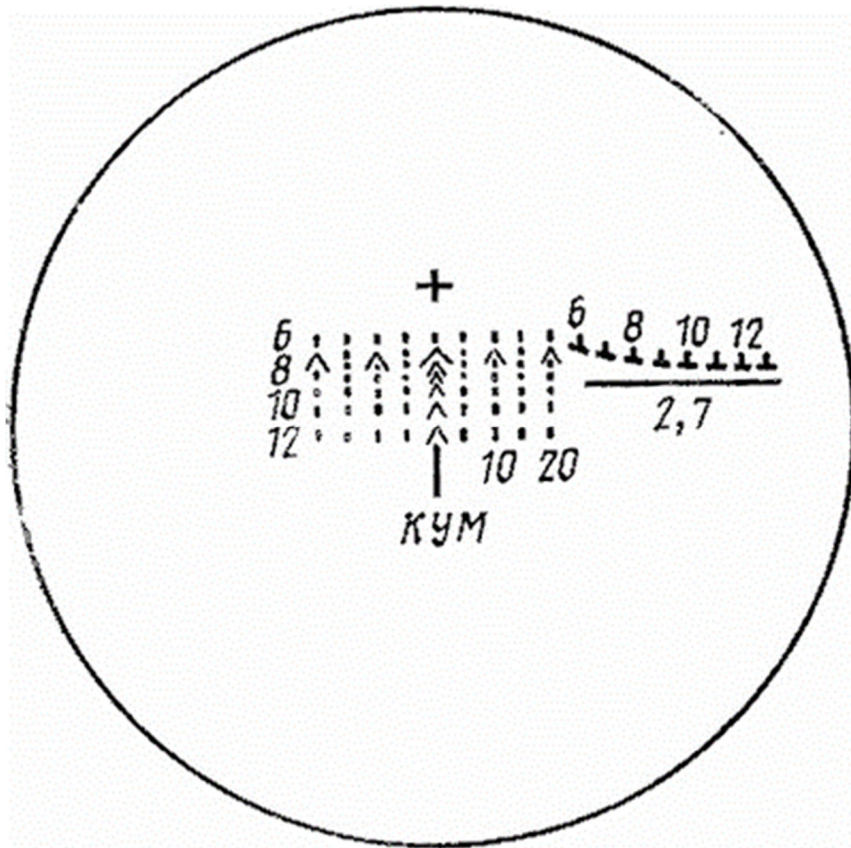


Рис. 3.22. Питання 10

Після створення нового проекту, створюємо сцену на якій буде все відображатись, та додаємо файл з кодом на мові C#, який буде описувати поведінку програми.(рис. 3.23)

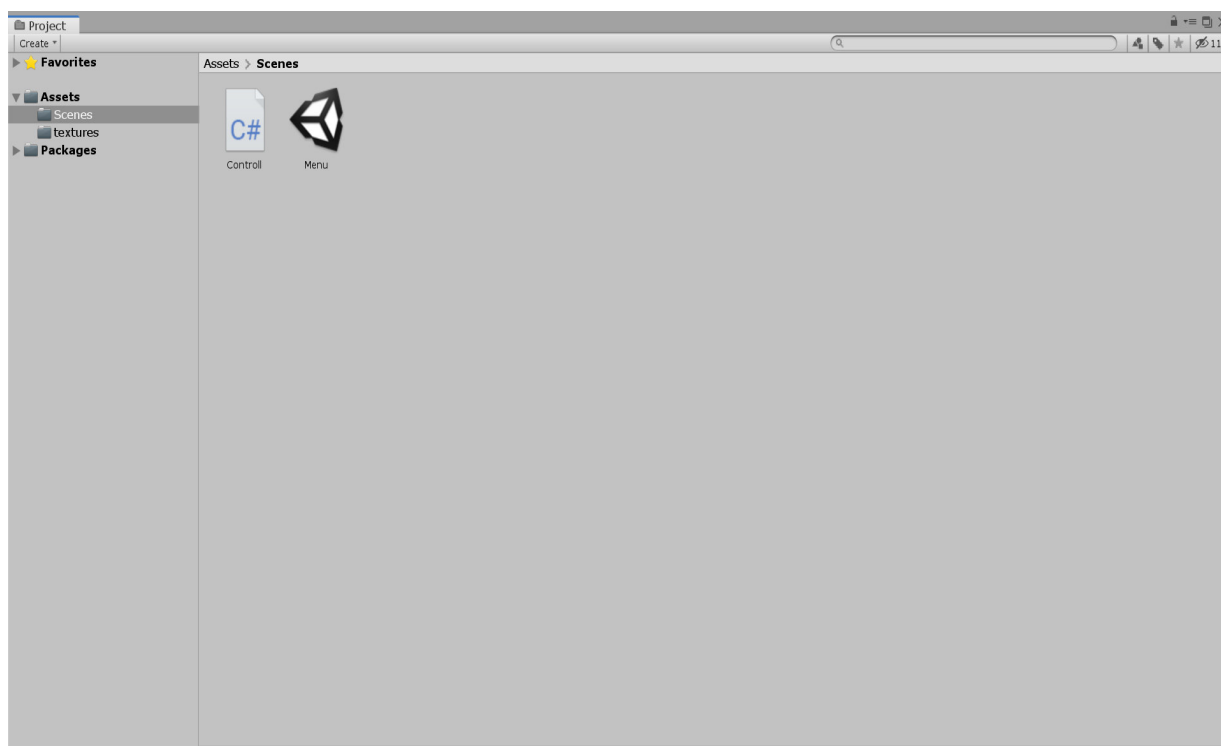


Рис. 3.23. Сцена та файл з кодом

Оскільки в нас повністю пуста сцена (рис. 3.24), то ми додаємо панель, на якій буде відображатись інформація.

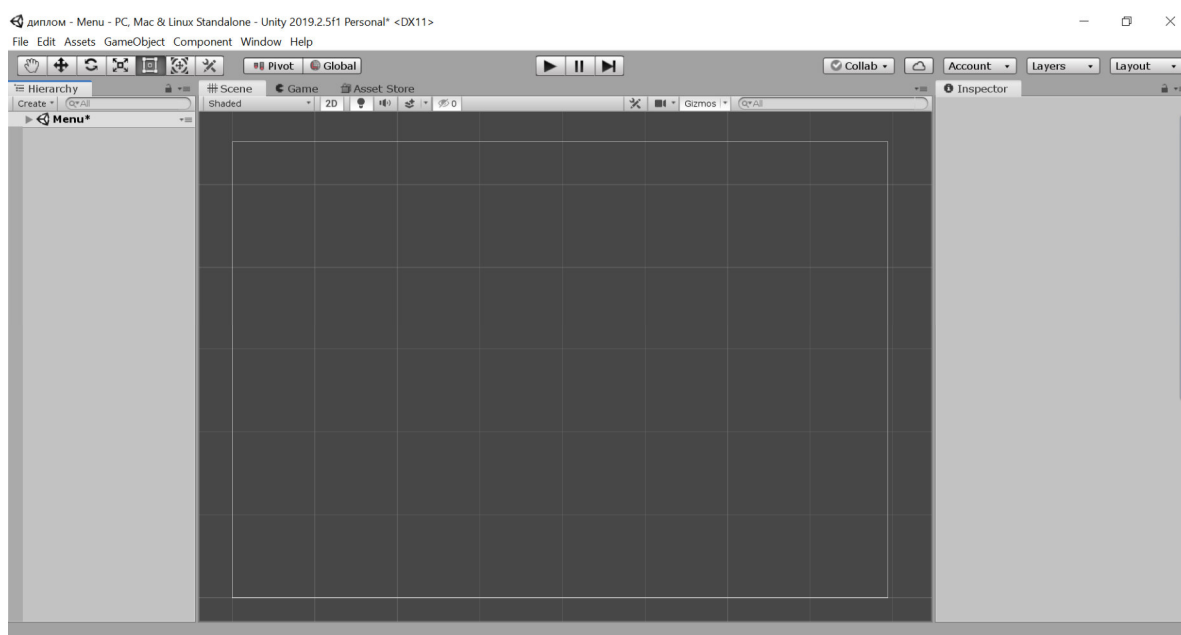


Рис. 3.24. Вигляд сцени перед внесенням до неї панелі

Для того, щоб потім була можливість використовувати рисунки, вони повинні бути додані до проекту.(рис. 3.25)



Рис. 3.25. Список рисунків у проекті

Оскільки за тематикою, проект є тренажером, то для фону панелі обрано малюнок з зеленню. На (рис. 3.26-3.27) показано, як встановлюється фон панелі.

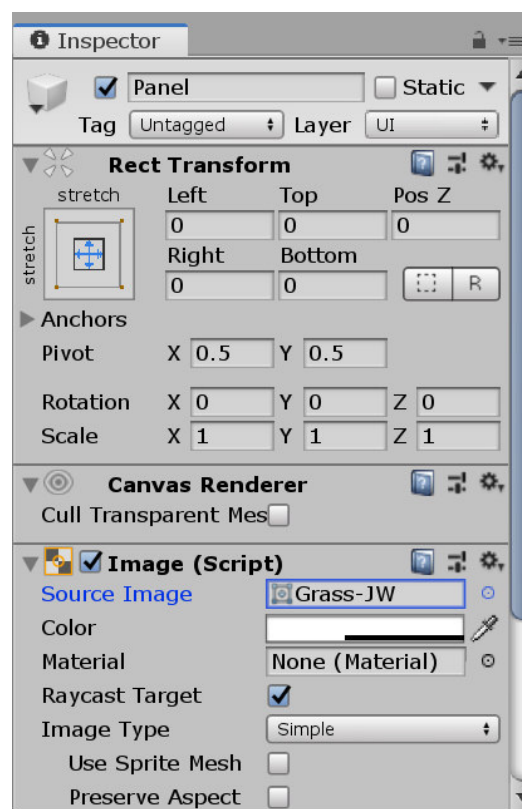


Рис. 3.26. Параметр, який відповідає за фон панелі

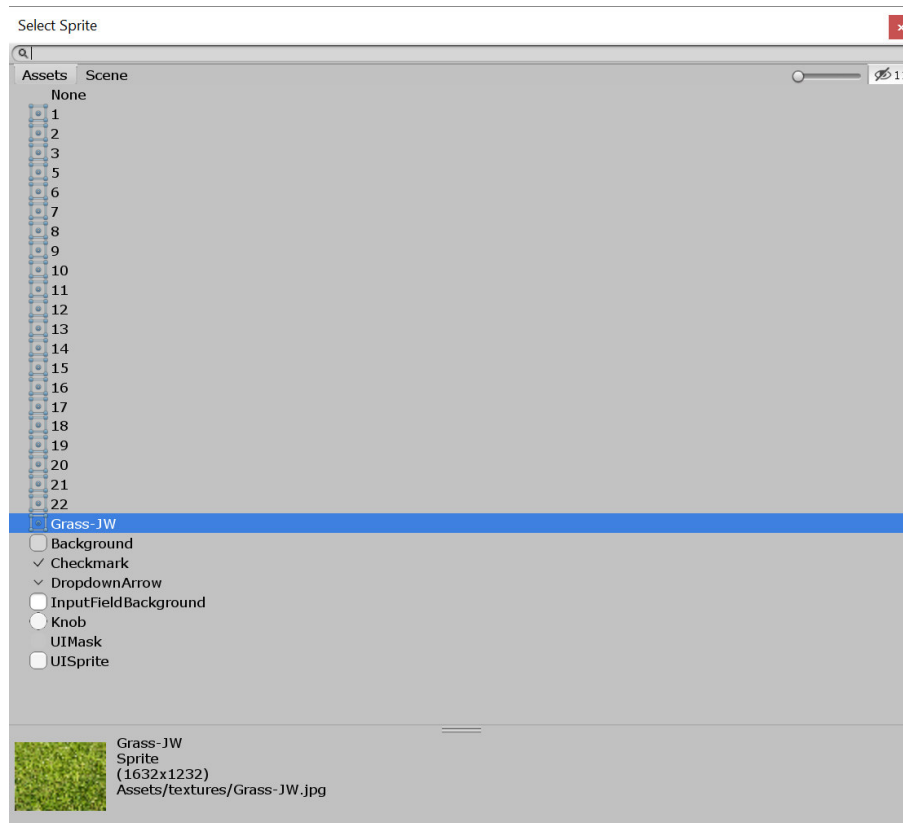


Рис. 3.27. Рисунок було обрано серед списку

Далі, на панель було встановлено дві кнопки, які відповідають за початок тесту та за вихід з тесту. Для зручності роботи програми, їх було додано в один пустий об'єкт, який називається *Menu* (рис. 3.28)

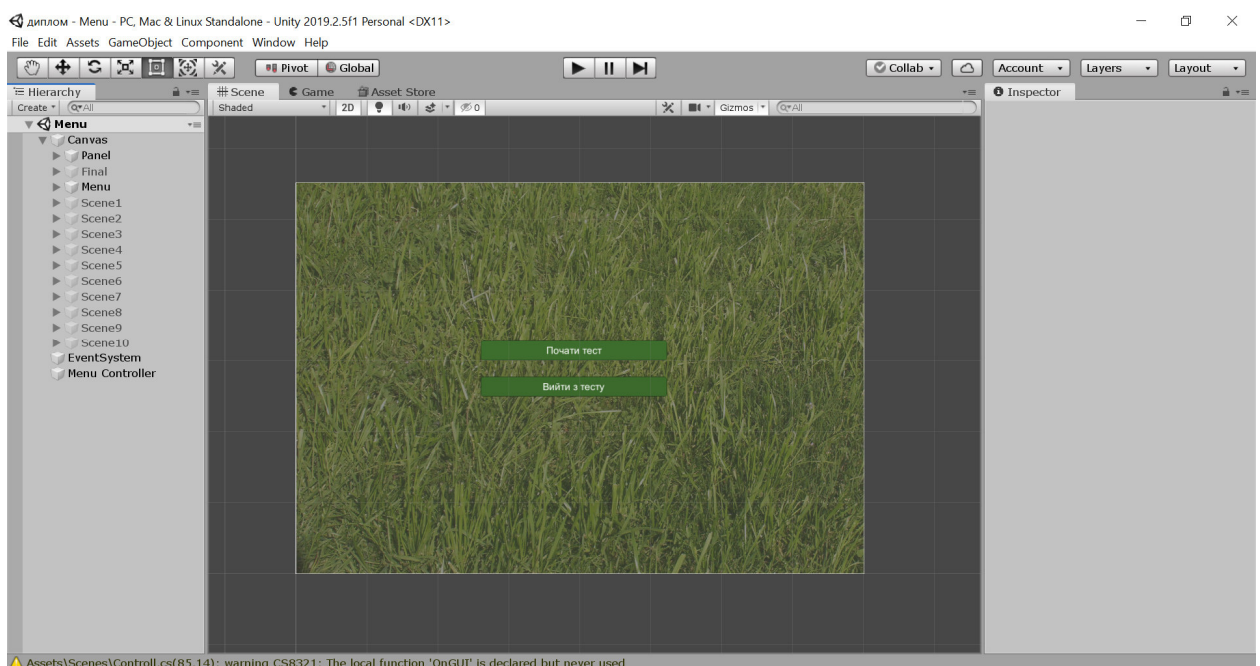


Рис. 3.28. Додання кнопок

За схожим принципом створюємо й інші сцени для тесту та сцену з результатом (рис. 3.29-3.39)

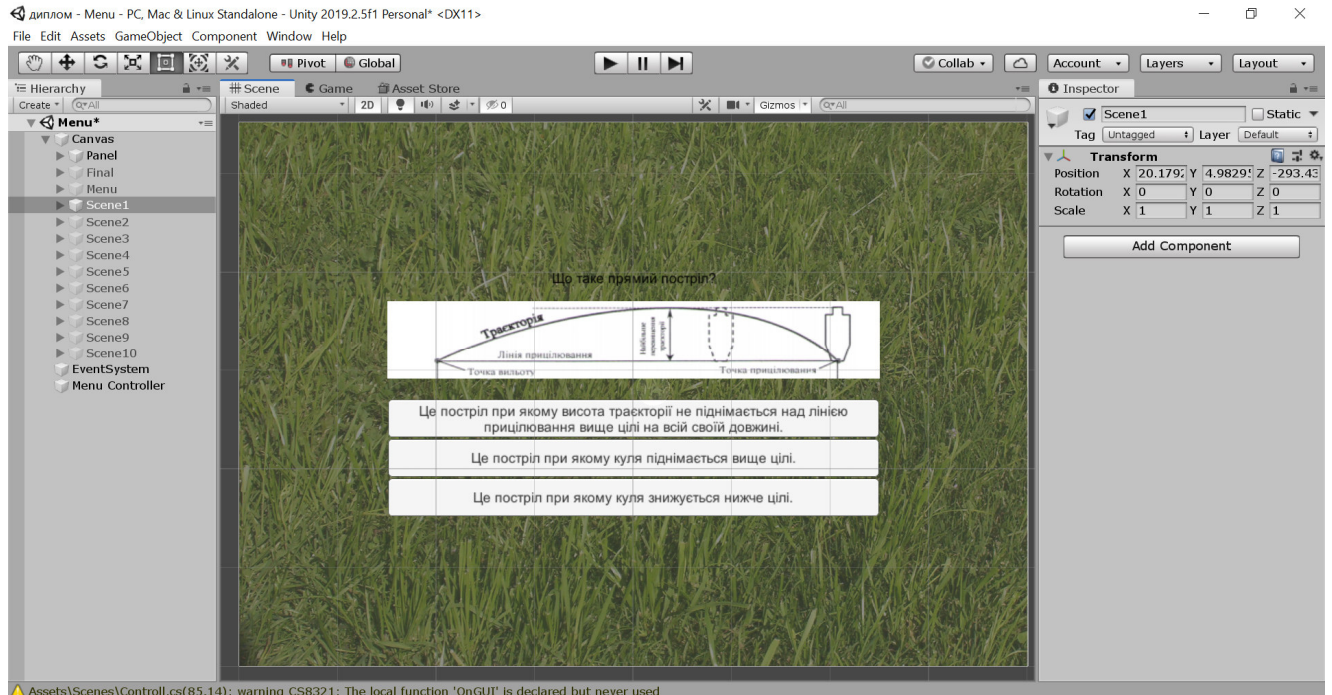


Рис. 3.29. Вигляд першого питання в програмі

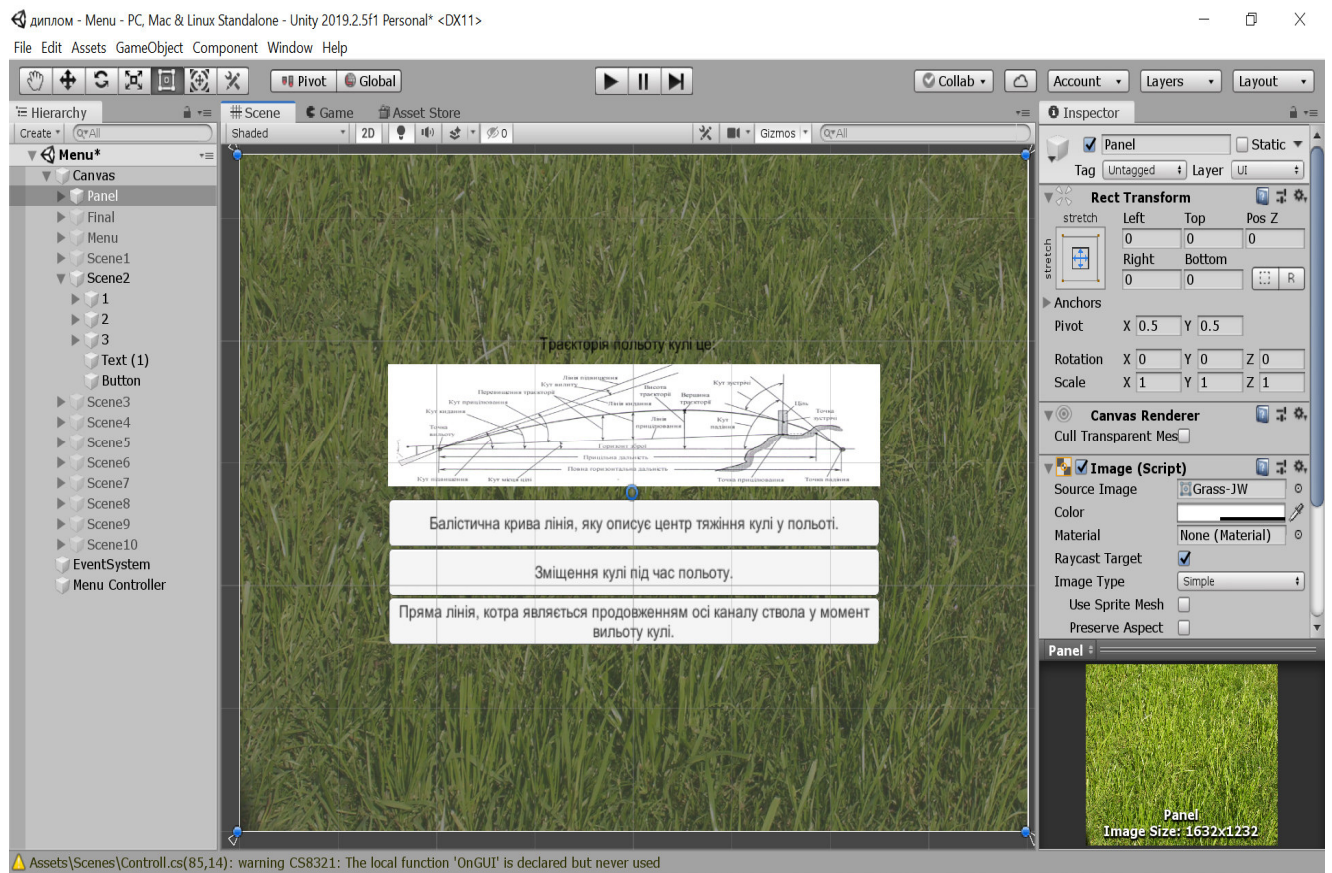


Рис. 3.30. Вигляд другого питання в програмі

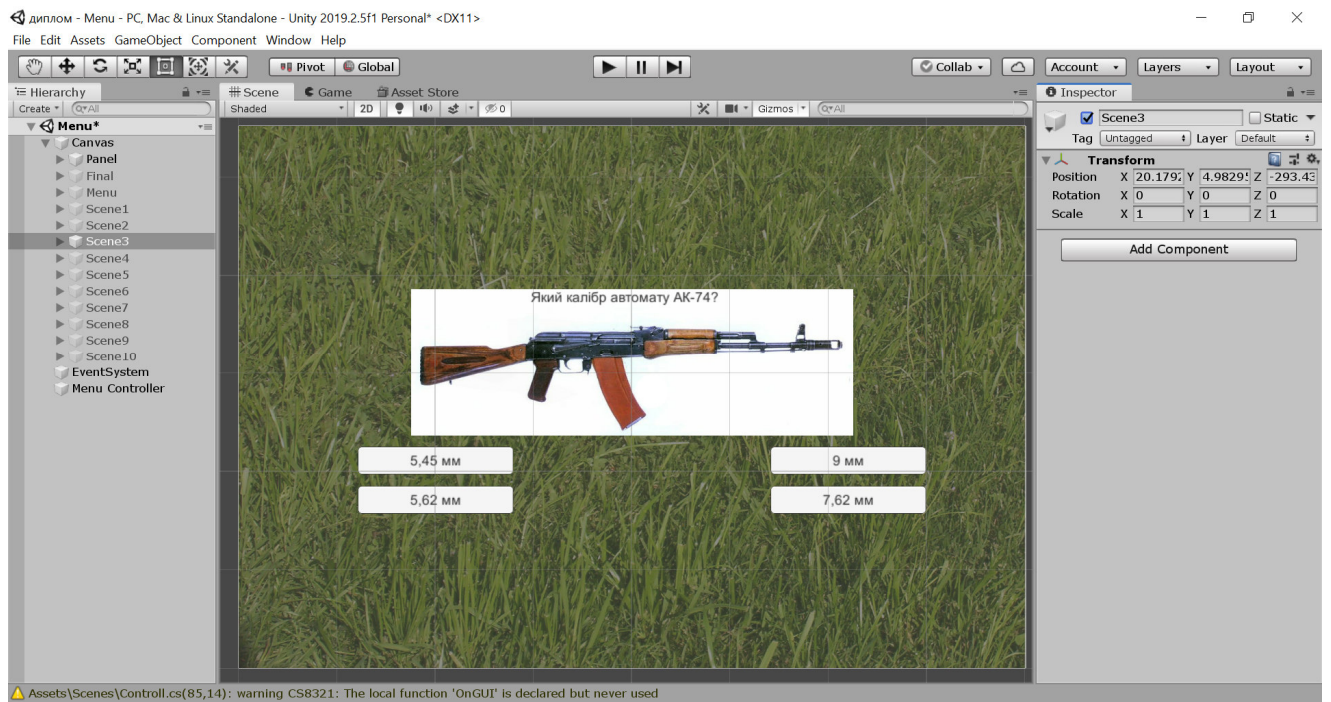


Рис. 3.31. Вигляд третього питання в програмі

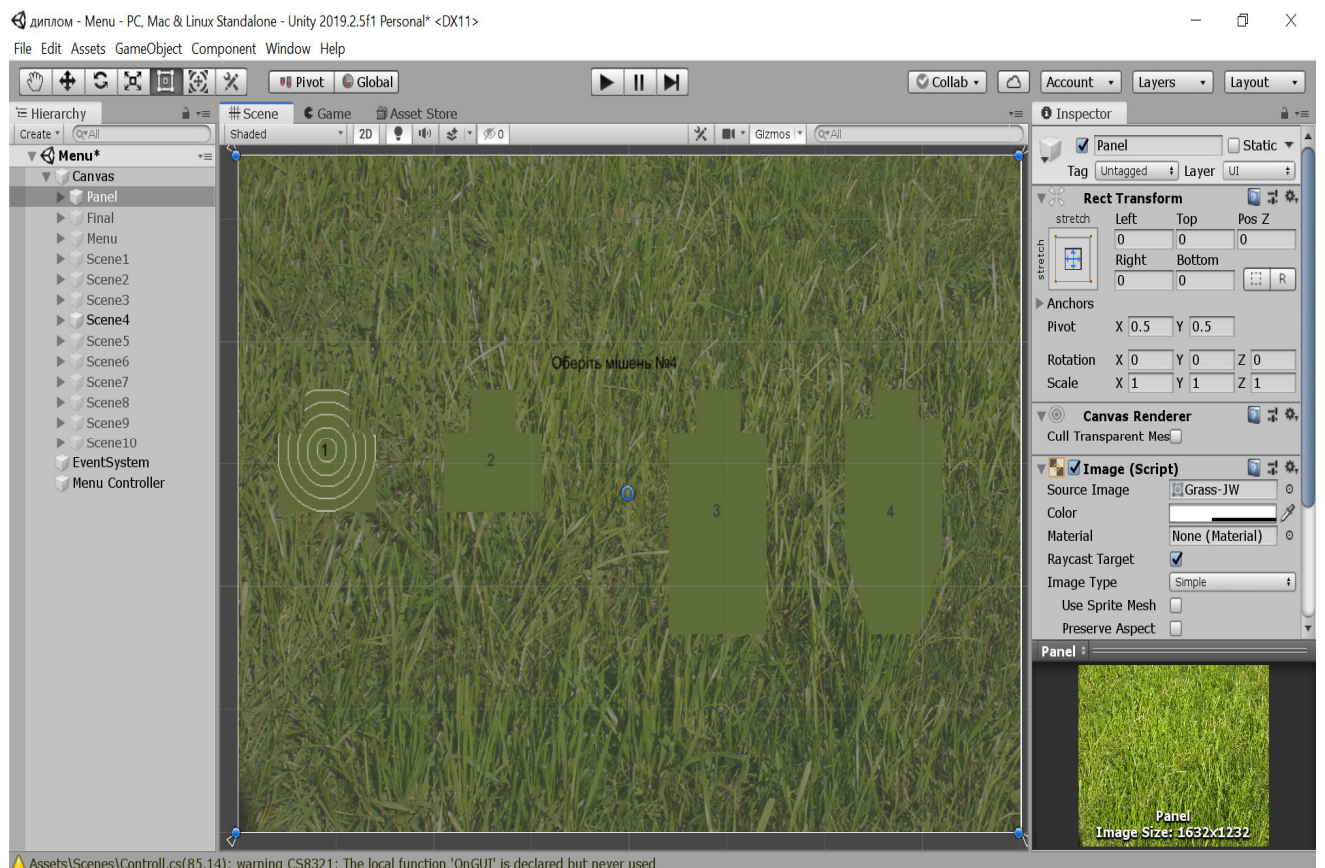


Рис. 3.32. Вигляд четвертого питання в програмі

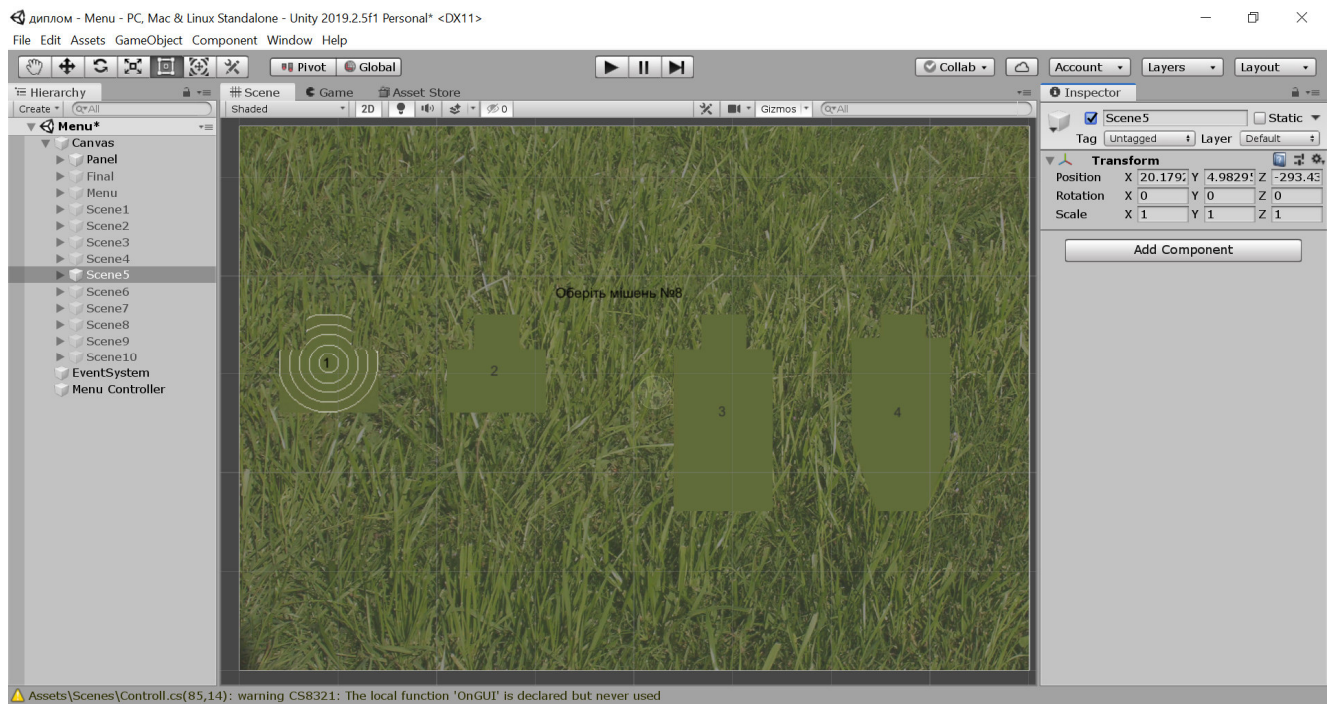


Рис. 3.33. Вигляд п'ятого питання в програмі

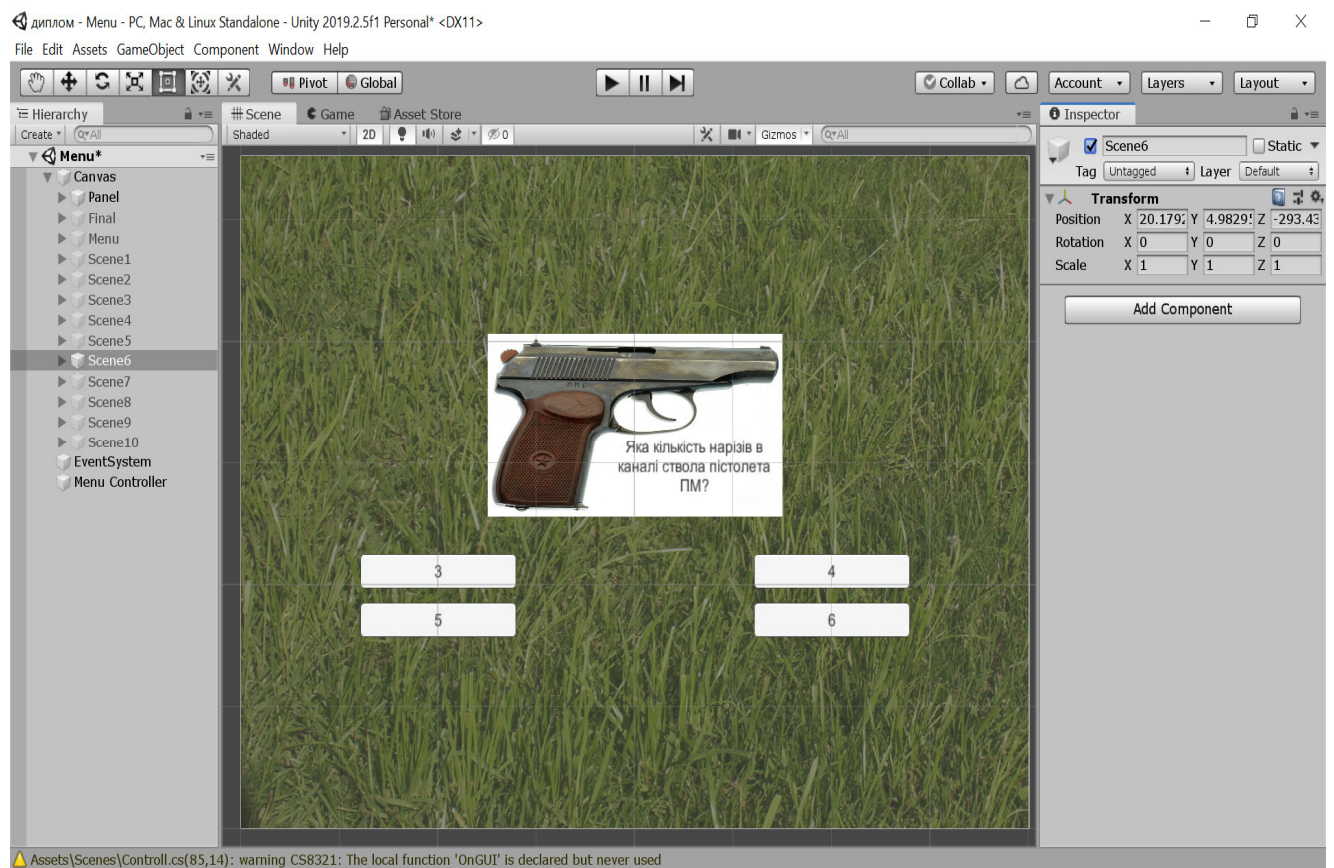


Рис. 3.34. Вигляд шостого питання в програмі

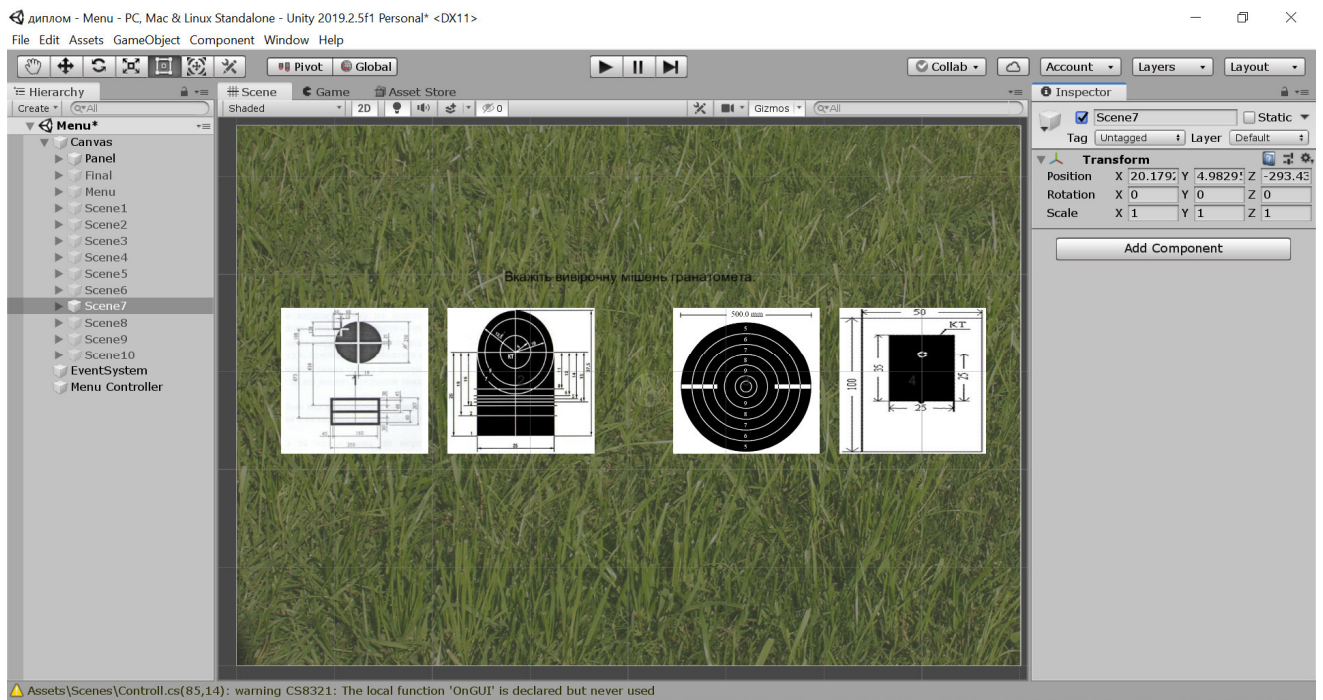


Рис. 3.35. Вигляд сьомого питання в програмі

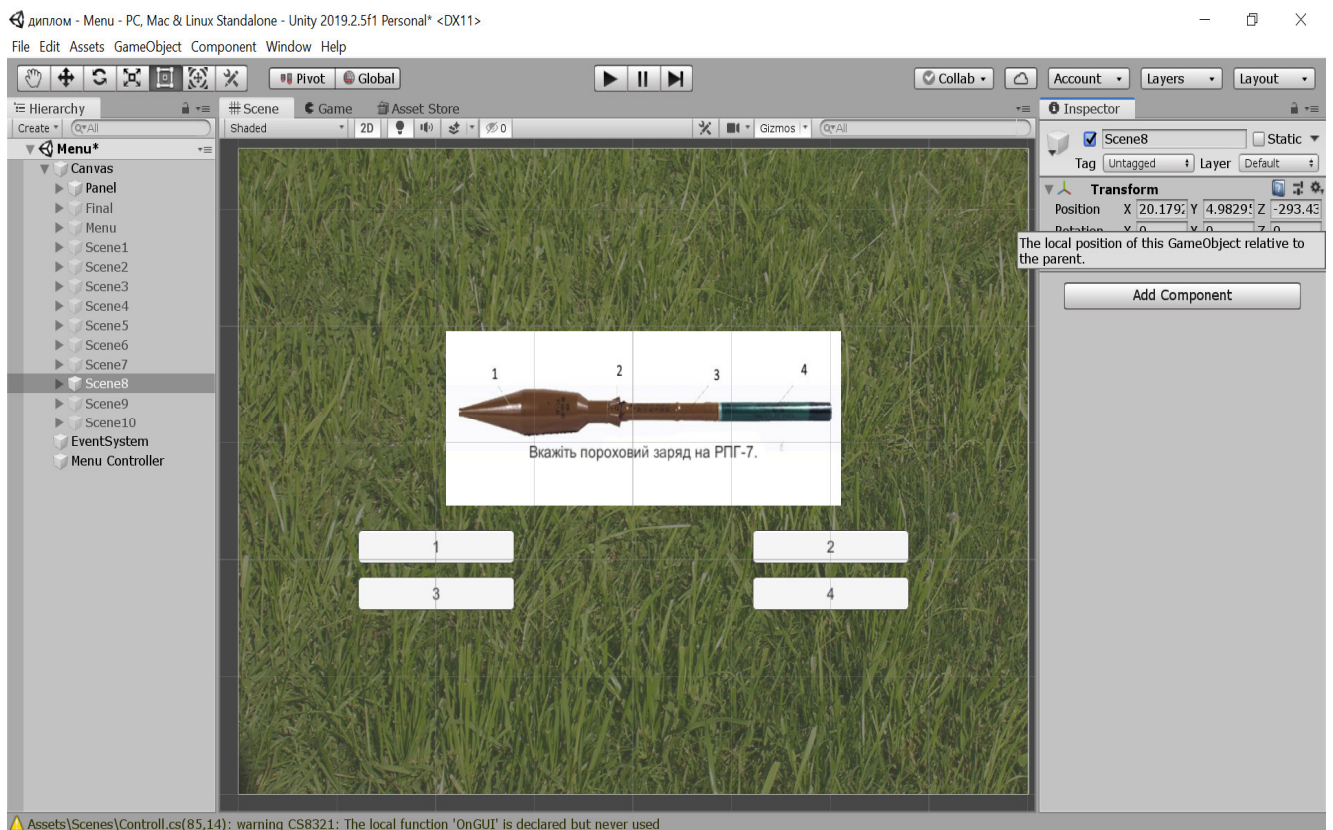


Рис. 3.36. Вигляд восьмого питання в програмі

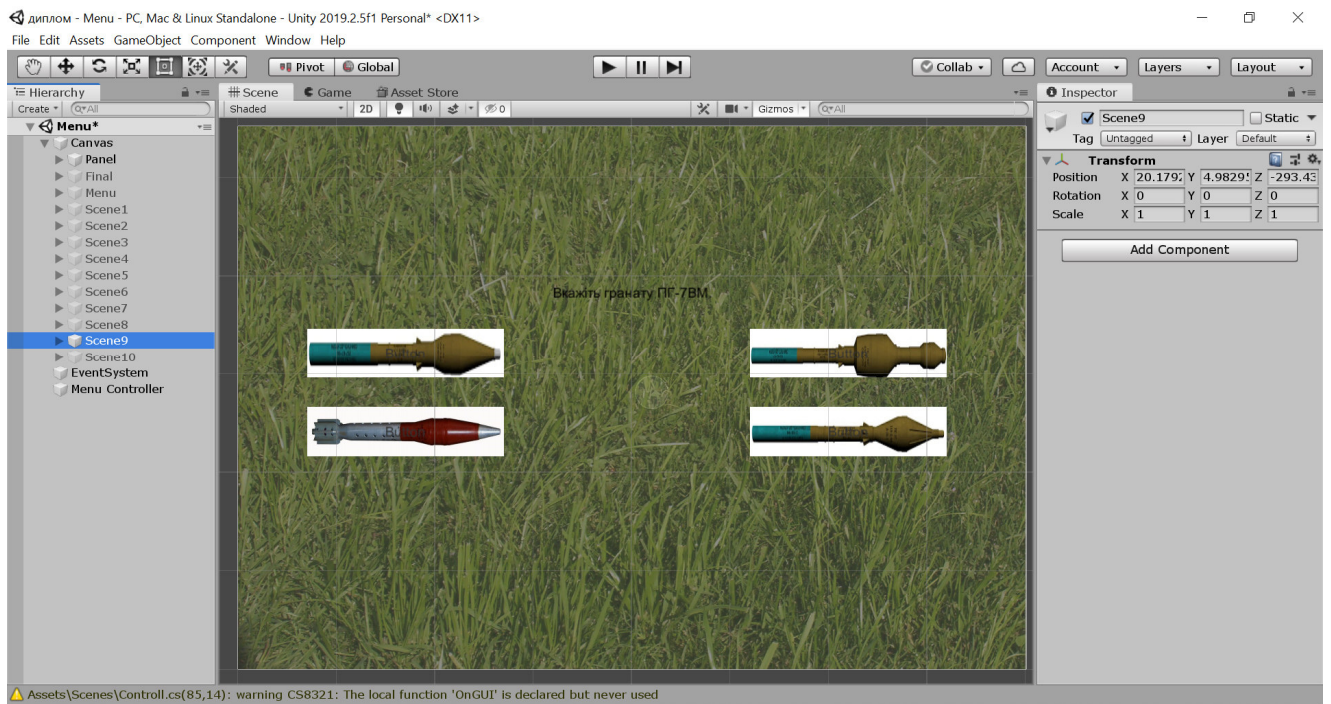


Рис. 3.37. Вигляд дев'ятого питання в програмі

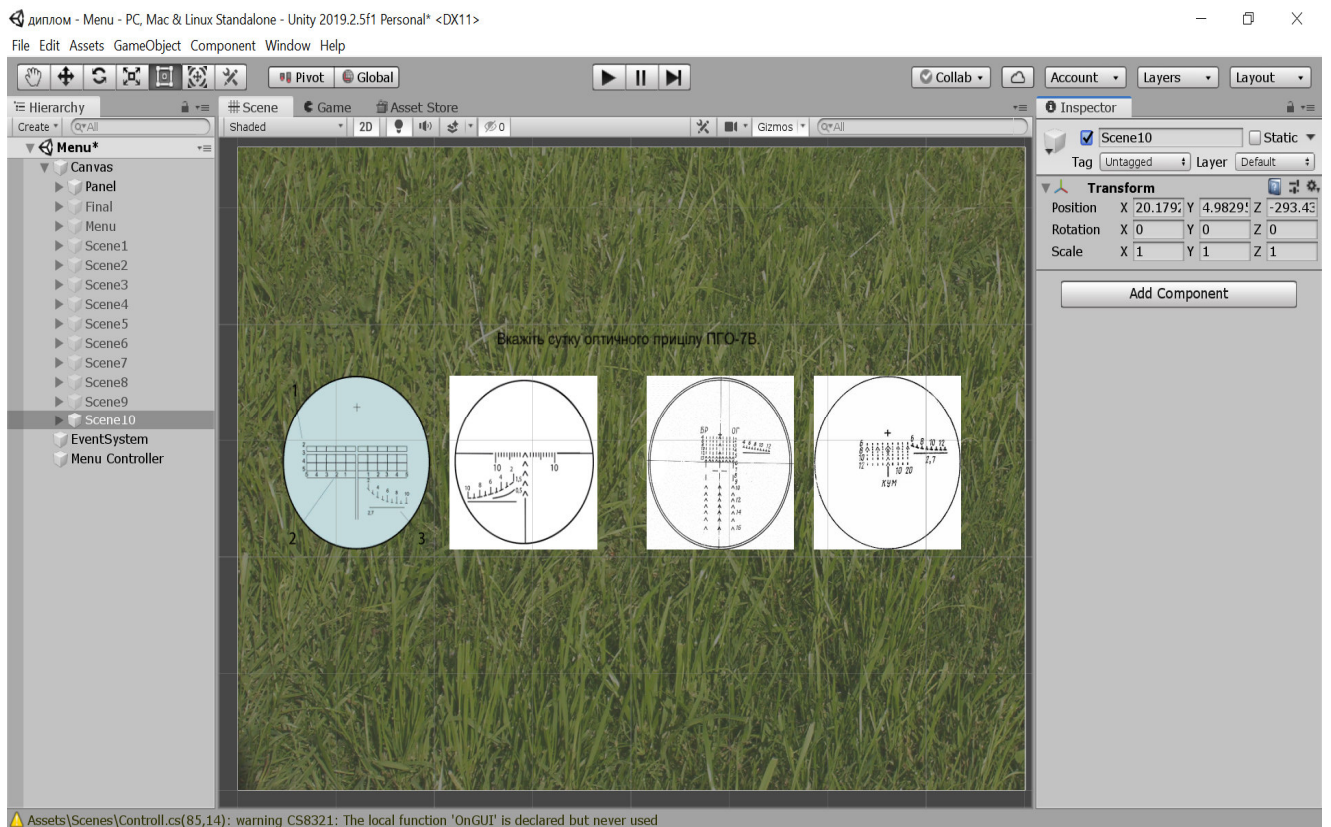


Рис. 3.38. Вигляд десятого питання в програмі

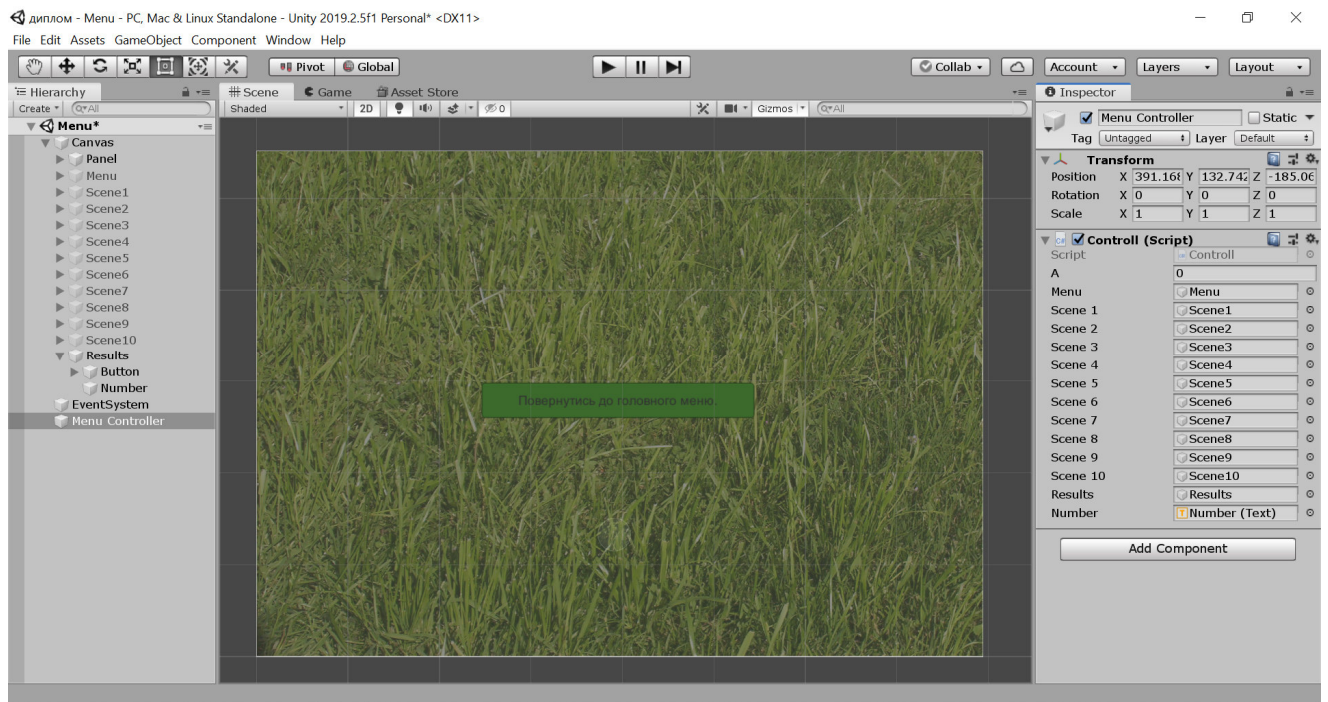


Рис. 3.39. Вигляд сцени де виводяться результати

Далі, було написано код для того, щоб працювала програма. Для написання коду використовувалась програма *Visual Studio*.

Код програми:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;

public class Controll : MonoBehaviour
{
    public int a = 0;
    public GameObject Menu, Scene1, Scene2, Scene3, Scene4,
Scene5, Scene6, Scene7, Scene8, Scene9, Scene10, Results;
    public Text Number;

    public void Start()
```

```
{
    Menu.SetActive(true);
    Scene1.SetActive(false);
    Scene2.SetActive(false);
    Scene3.SetActive(false);
    Scene4.SetActive(false);
    Scene5.SetActive(false);
    Scene6.SetActive(false);
    Scene7.SetActive(false);
    Scene8.SetActive(false);
    Scene9.SetActive(false);
    Scene10.SetActive(false);
    Results.SetActive(false);
}

public void ExitGame()
{
    Application.Quit();
}

public void NewGame()
{
    Menu.SetActive(false);
    Scene1.SetActive(true);
    Scene2.SetActive(false);
    Scene3.SetActive(false);
    Scene4.SetActive(false);
    Scene5.SetActive(false);
    Scene6.SetActive(false);
    Scene7.SetActive(false);
    Scene8.SetActive(false);
    Scene9.SetActive(false);
}
```

```
        Scene10.SetActive(false);
        Results.SetActive(false);
    }
    public void RightAnswer()
    {
        a++;
    }
    public void ChangeOnScene2()
    {
        Scene1.SetActive(false);
        Scene2.SetActive(true);
    }
    public void ChangeOnScene3()
    {
        Scene2.SetActive(false);
        Scene3.SetActive(true);
    }
    public void ChangeOnScene4()
    {
        Scene3.SetActive(false);
        Scene4.SetActive(true);
    }
    public void ChangeOnScene5()
    {
        Scene4.SetActive(false);
        Scene5.SetActive(true);
    }
    public void ChangeOnScene6()
    {
        Scene5.SetActive(false);
        Scene6.SetActive(true);
    }
}
```

```

public void ChangeOnScene7()
{
    Scene6.SetActive(false);
    Scene7.SetActive(true);
}
public void ChangeOnScene8()
{
    Scene7.SetActive(false);
    Scene8.SetActive(true);
}
public void ChangeOnScene9()
{
    Scene8.SetActive(false);
    Scene9.SetActive(true);
}
public void ChangeOnScene10()
{
    Scene9.SetActive(false);
    Scene10.SetActive(true);
}
public void ChangeOnResults()
{
    int b = 0;
    if ((a == 10) | (a == 9)) b=5;
    if ((a == 8) | (a == 7)) b = 4;
    if ((a == 6) | (a == 5)) b = 3;
    if ((a == 4) | (a == 3)) b = 2;
    if ((a == 2) | (a == 1)) b = 1;
    if (a == 0) b = 0;
    string text = "Вітаю, ви пройшли тест! З 10 питань ви
правильно відповіли на: " + a.ToString() + ". Рекомендована
оцінка: " + b + ".";
}

```

```

Results.SetActive(true);
Scene10.SetActive(false);
Number.text = text;
}
public void BackToMenu()
{
Results.SetActive(false);
Menu.SetActive(true);
a = 0;
}
}

```

Оскільки програма не може працювати напряму з кодом, то було додано пустий об'єкт, куди додано код. І кожен елемент програми працював вже з цим елементом. Для цього, кожній кнопці на панелі був присвоєний параметр *OnClick*, який посилався на об'єкт з кодом. Також, для кожної сцени був створений окремий метод, який дозволяв перейти на наступну сцену (рис. 3.40 – 3.41).

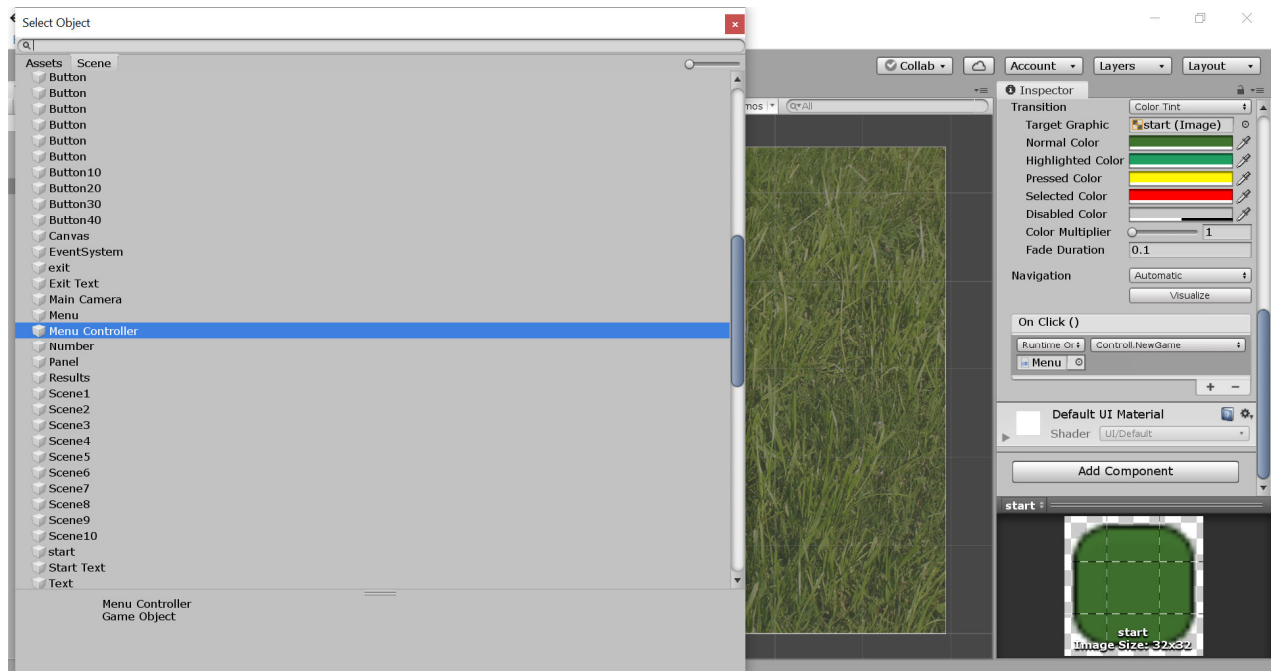


Рис. 3.40. Вибір об'єкту з кодом

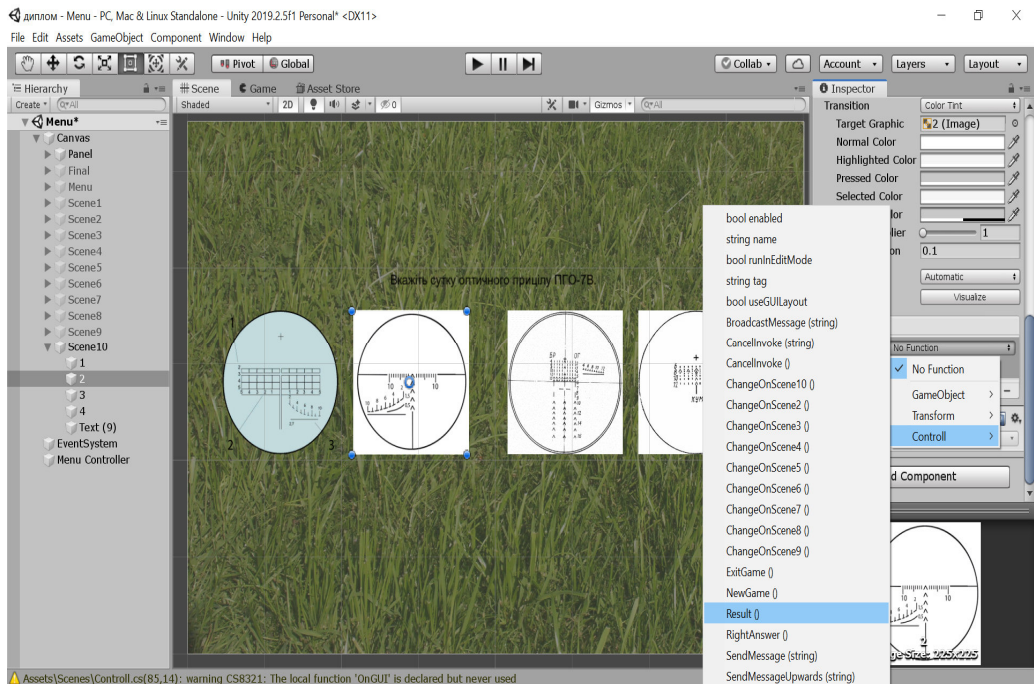


Рис. 3.41. Вибір методу для відповідної кнопки

Також, програмі потрібно прив'язати код до об'єктів на сцені. Для цього, в об'єкті в якому знаходиться код, вказуємо які об'єкти коду відповідають об'єктам сцени (рис. 3.42).



Рис. 3.42. Прив'язка об'єктів

Далі, для того щоб програма працювала без встановлення *Unity*, потрібно її скомпілювати, та обрати місце куди зберегти. Оскільки програма буде працювати на комп'ютерах, то обираємо саме ці платформи (рис. 3.43 – 3.44).

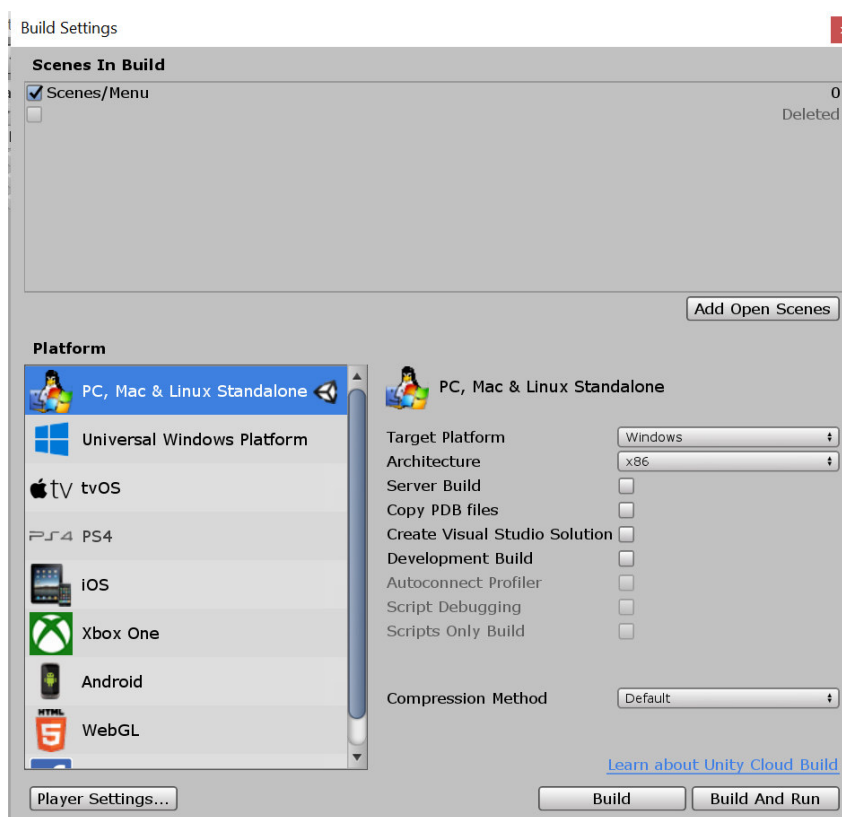


Рис. 3.43. Компілювання програми

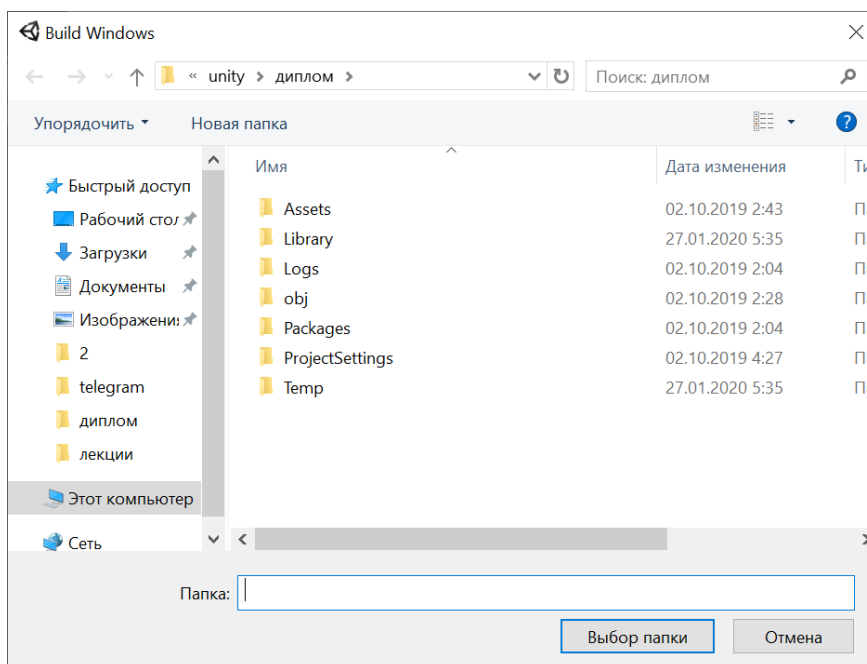


Рис. 3.44. Обираємо куди скомпілювати програму

Після того, як програма була скомпільована, ми маємо декілька папок та застосунок для запуску програми (рис. 3.45).

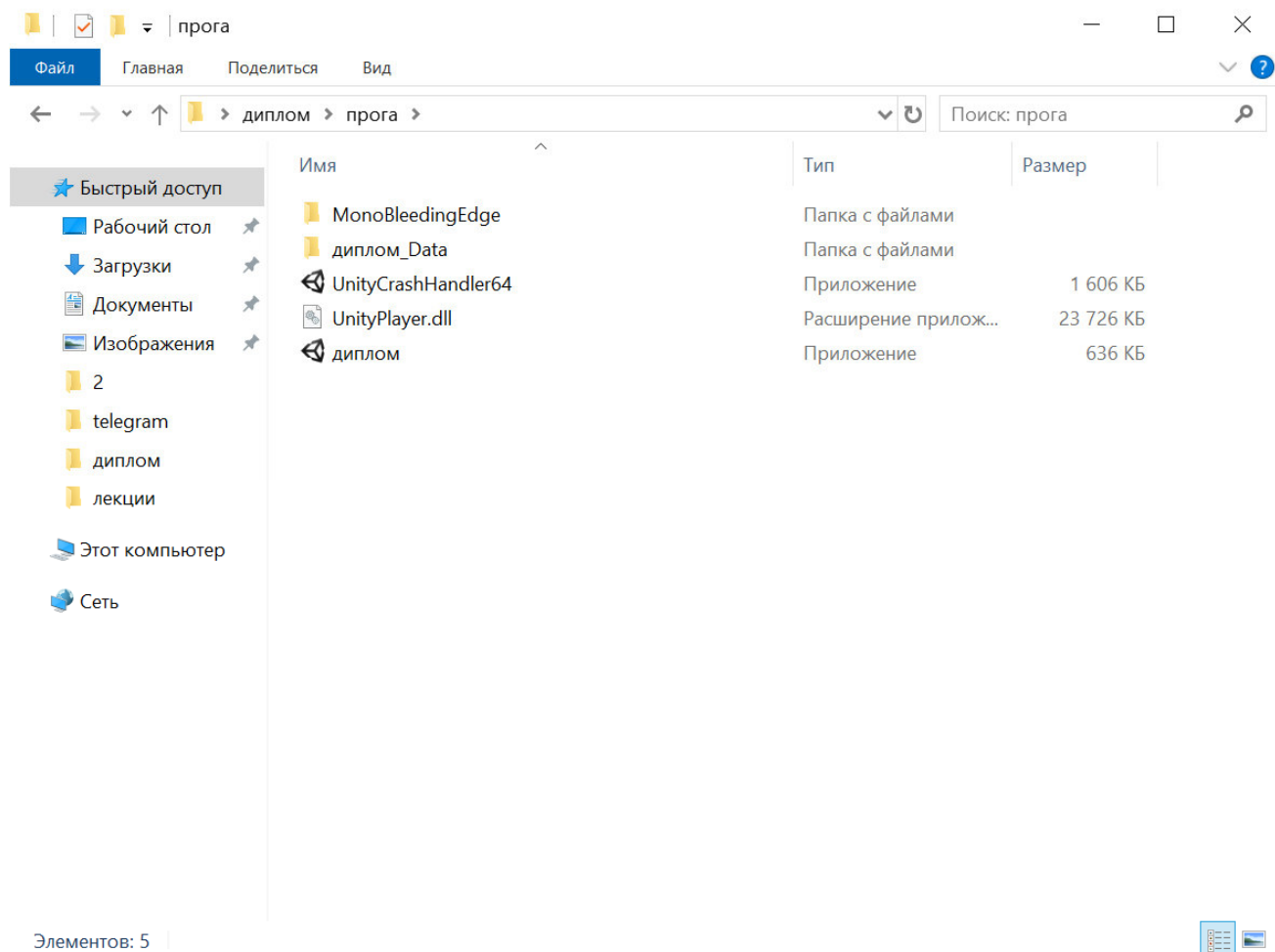


Рис. 3.45. Готові файли програми

Оскільки для розповсюдження програми незручно переносити багато файлів та папок, було вирішено скомпілювати інсталятор для дипломної роботи. Для цього, було встановлено та використано зовнішню програму *Smart Install Maker* (рис. 3.46).

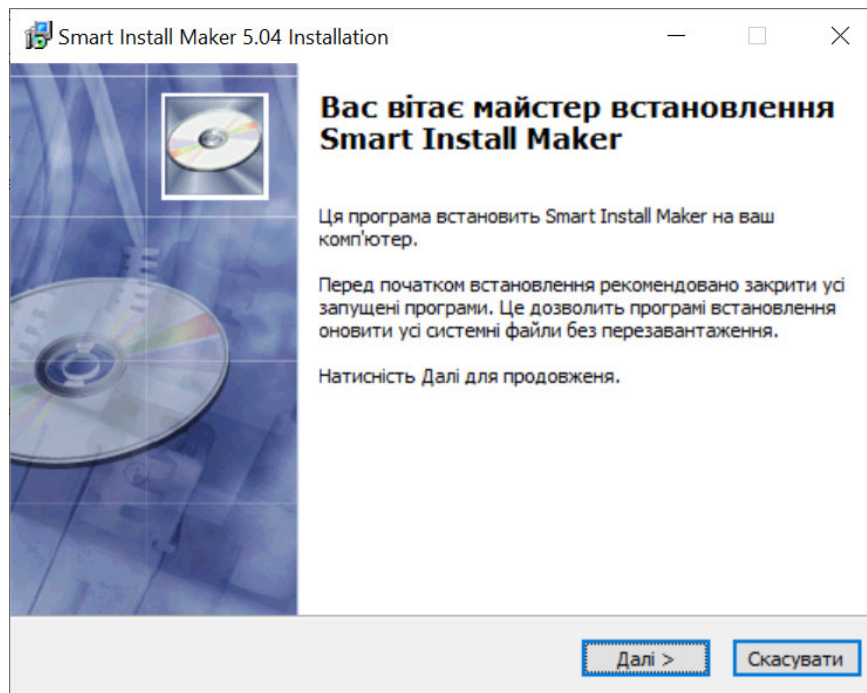


Рис. 3.46. Встановлення *Smart Install Maker*

Для того, щоб зробити інсталятор, було обрано заголовок інсталятору та версію програми. Також, додано назву компанії та поштову адресу для запитань по експлуатації програми. Також, встановлено місце, куди розмістити створений інсталятор.(рис. 3.47)

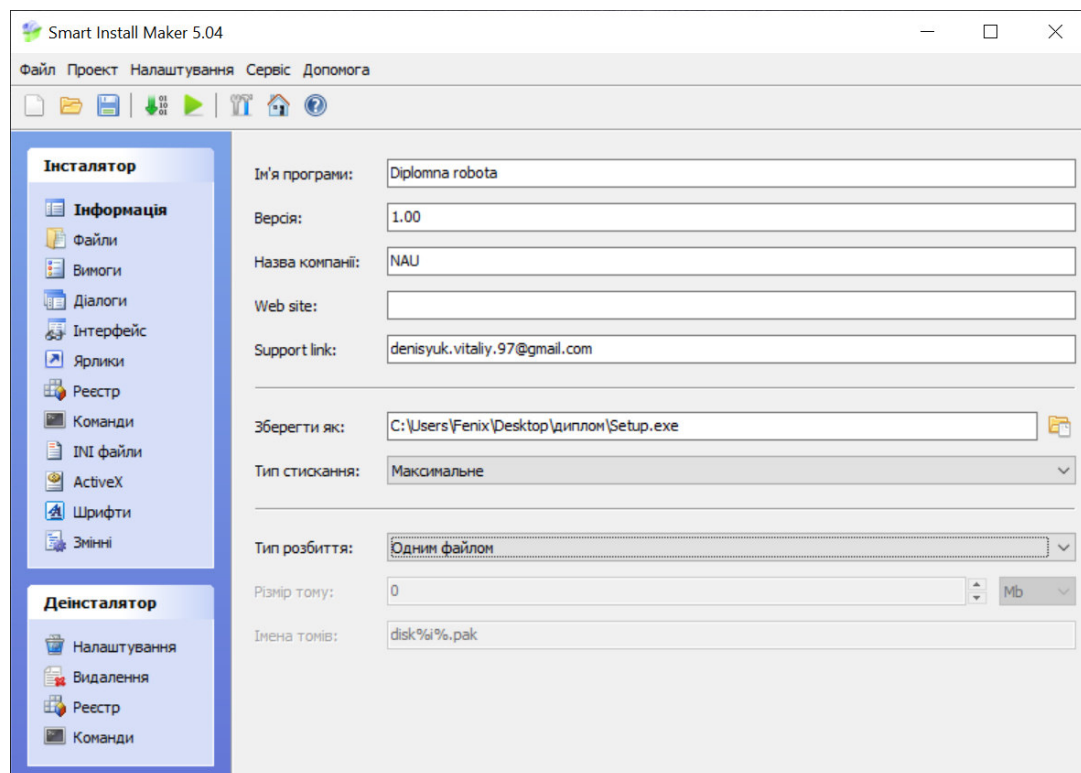


Рис. 3.47. Параметри задані для інсталятору

Далі, було обрано файли які потрібно буде інсталювати на комп'ютери користувачів (рис. 3.48).

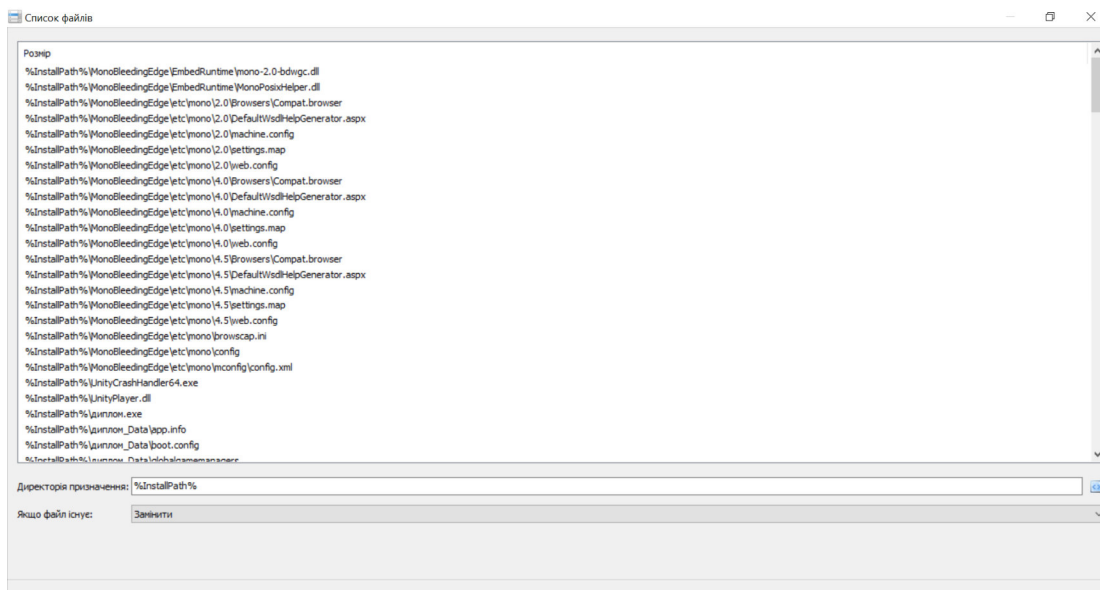


Рис. 3.48. Вибір файлів які потрібно інсталювати

Оскільки більшість користувачів запускають свої програми з робочого столу, було вирішено додати це вікно до вікна інсталювача. Також, було додано інші параметри інсталяції. Оскільки програма не має корпоративної групи, то це вікно було пропущено (рис. 3.49).

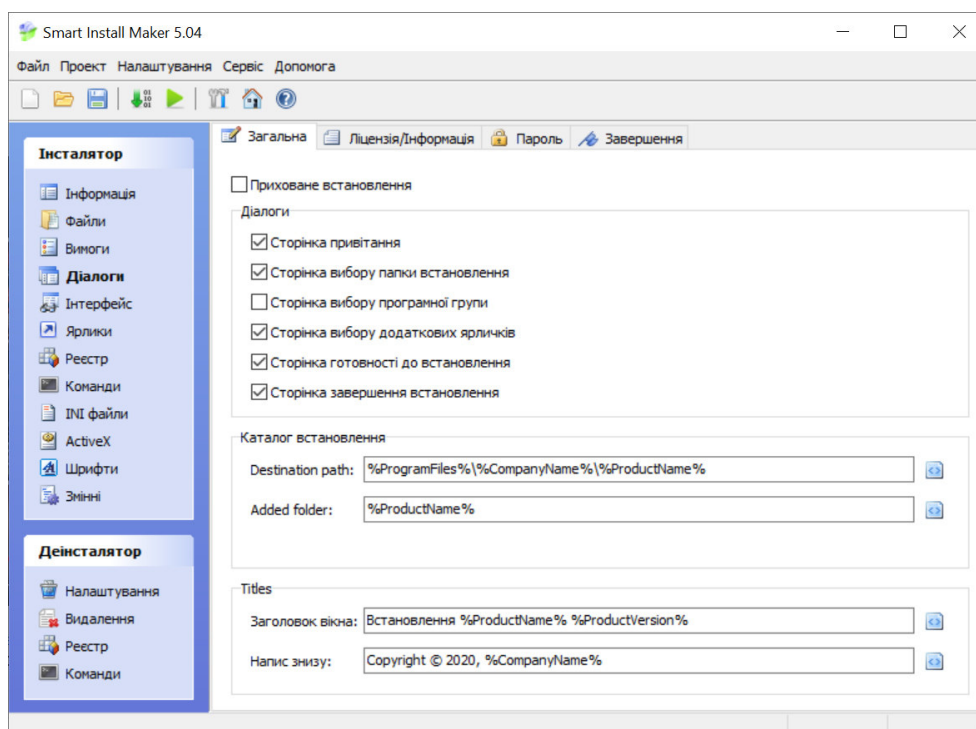


Рис. 3.49. Вибір діалогових вікон інсталювача

Після компіляції інсталювачу було отримано один файл (рис. 3.50).

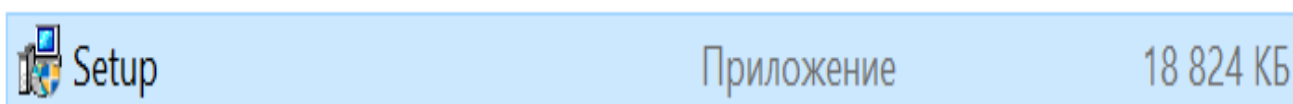


Рис. 3.50. Отриманий файл для інсталяції

Враховуючи, що початкова папка з програмою важила 61,7 МБ, то створення інсталювачу можна вважати доцільним.

Висновки до розділу

У третьому розділі було розроблено інтерактивне навчальне середовище лазерного стрілецького тренажеру на основі ігрового рушія *Unity*. Було досліджено процес розробки програми на даному ігровому рушію.

Було прийнято рішення мінімізувати вигляд програми для того, щоб не ставити додаткові програми для показу контенту навчального середовища.

Були розв'язані проблеми мінімізування розміру програми та графічного навантаження. Також, була введена 5-ти бальна система оцінювання на основі кількості правильних відповідей.

ВИСНОВКИ

У першому розділі дипломного проекту було проаналізовано поняття моделі змішаного навчання та його види: модель "*Face-to-Face Driver*", модель "*Rotation*", модель "*Flex*", модель "*Online Lab*", модель "*Self-blend*", модель "*Online Driver*".

Як показує аналіз існуючих моделей змішаного навчання, ключовим моментом в їх проектуванні є вибір правильного співвідношення всіх методів навчання, що використовуються, та встановлення оптимального поєднання традиційних технологій з технологіями електронного навчання.

При проектуванні електронного інтерактивного навчального середовища спиралися на метод комплексного системно-діяльнісного моделювання процесу навчання, який передбачає одночасне поєднання моделювання змісту навчання, моделювання навчально-пізнавальної діяльності студентів та моделювання взаємодії учасників навчального процесу.

Моделювання змісту в цій системі є основним, особливості технічної дисципліни обумовлюють підпорядкування змістовного компонента навчання діяльнісному компоненту.

Також, було проаналізовано лазерні стрілкові тренажери та оснащення для їх успішної роботи.

Лазерний стрілецький тренажер побудований на базі класичного відео проектора.

Стрільба здійснюється за допомогою лазерного цілевказівника (ЛЦВ), вбудованого в пістолет або автомат короткими світловими імпульсами по цілям на пристрої відображення (проекційний екран або інтерактивна дошка). До складу такого тиру обов'язково входить камера, яка фіксує позначки лазерного променя на проекційному екрані і миттєво спрямовує цю інформацію в комп'ютер. Комп'ютер обчислює координати такого пострілу і відповідним чином змінює зображення на екрані (демонструє потрапляння або промах).

У другому розділі дипломного проекту розглянуто декілька найбільш популярних ігрових рушіїв.

Було обрано ігровий рушії *Unity*, бо тут використовується компонентно-орієнтований підхід, в рамках якого розробник створює об'єкти і до них додає різні компоненти. Завдяки зручному *Drag & Drop* інтерфейсу і функціональним графічному редактору рушії дозволяє розставляти об'єкти в реальному часі і відразу ж тестувати, який вийшов результат.

Сильна сторона *Unity* – підтримка величезної кількості платформ, технологій, API. Створені на движку гри можна легко перенести між ОС *Windows*, *Linux*, *Android*, *iOS*, на консолі сімейств *PlayStation*, *Xbox*, *Nintendo*, на VR- і AR-пристрої. *Unity* підтримує *DirectX* і *OpenGL*,

На основі аналізу ігрових рушіїв, було вирішено для розробки навчального середовища застосовувати ігровий рушії *Unity*.

У третьому розділі було розроблено інтерактивне навчальне середовище лазерного стрілецького тренажеру на основі ігрового рушія *Unity*. Було досліджено процес розробки програми на даному ігровому рушію.

Було прийнято рішення мінімізувати вигляд програми для того, щоб не ставити додаткові програми для показу контенту навчального середовища.

Були розв'язані проблеми мінімізування розміру програми та графічного навантаження. Також, була введена 5-ти бальна система оцінювання на основі кількості правильних відповідей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Комплект интерактивного стрелкового лазерного тира "захисник вітчизни" [Електронний ресурс]. Режим доступу: <https://sound-design.kiev.ua/p291325125-komplekt-interaktivnogo-strelkovogo.html>.
2. Лазерные тренажеры [Електронний ресурс]. Режим доступу: <http://www.lasertools.ru/products/trainer.html>.
3. *Online Lab Model in Blended Learning: Definition, Application & Examples* [Електронний ресурс]. Режим доступу: <https://study.com/academy/lesson/online-lab-model-in-blended-learning-definition-application-examples.html>.
4. *Online Driver Model: Definition, Application & Examples* [Електронний ресурс]. Режим доступу: <https://study.com/academy/lesson/online-driver-model-definition-application-examples.html>.
5. *Self-blended model of learning* [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/Self-blended_model_of_learning.
6. "Змішане навчання": персоналізоване навчання кожного учня [Електронний ресурс]. Режим доступу: <http://prosvitcenter.org/zmishane-navchannya>.
7. *Face-to-Face Driver Model: Application & Examples* [Електронний ресурс]. Режим доступу: <https://study.com/academy/lesson/face-to-face-driver-model-application-examples.html>.
8. *Rotation model of learning* [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/Rotation_model_of_learning.
9. Сравнение игровых движков [Електронний ресурс]. Режим доступу: <http://devgam.com/sravnenie-igrovyyh-dvizhkov-kakoj-vybrat>.
10. Обзор игровых движков для создания 3D игр [Електронний ресурс]. Режим доступу: <https://sibac.info/studconf/science/xxxi/84837>.
11. Лучший игровой движок по версии пользователей хабра [Електронний ресурс]. Режим доступу: <https://habr.com/ru/post/307952/>.

12. *Phaser – Game Framework* [Электронный ресурс]. Режим доступа: <https://gcup.ru/load/engines/phaser/3-1-0-2504>.

13. *Construct 2* [Электронный ресурс]. Режим доступа: https://uk.wikipedia.org/wiki/Construct_2.

14. *The 2D Game Engine* [Электронный ресурс]. Режим доступа: <https://coronalabs.com/product/>.

15. Обзор игрового движка *LibGDX* [Электронный ресурс]. Режим доступа: <https://gamedevmania.ru/engines/libgdx>.

16. *Amazon Lumberyard* – новый полностью бесплатный движок для игр AAA-класса [Электронный ресурс]. Режим доступа: <https://itc.ua/news/amazon-lumberyard-novyiy-polnostyu-besplatnyiy-dvizhok-dlya-igr-aaa-klasa/>.

17. *GameSalad Creator* [Электронный ресурс]. Режим доступа: <https://gcup.ru/load/constructors/gamesalad/2-1-0-516>.