

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ**  
Фізико-математичний факультет  
Кафедра інформатики та прикладної математики

«Допущено до захисту»

Завідувач кафедри

\_\_\_\_\_ Моїсеєнко Н.В.

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Реєстраційний № \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ФАКУЛЬТАТИВ З РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР ДЛЯ УЧНІВ ЛЦЕІВ**

Кваліфікаційна робота

ступінь вищої освіти «магістр»

спеціальності

014 Середня освіта (Інформатика)

**Пругло Олександра Олександровича**

Керівник:

кандидат фізико-математичних наук,

доцент

**Моїсеєнко Наталя Володимирівна**

Оцінка:

Національна шкала \_\_\_\_\_

Шкала ECTS \_\_\_\_\_ кількість балів \_\_\_\_\_

Члени комісії

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Кривий Ріг – 2024

## ЗАПЕВНЕННЯ

Я, Пругло Олександр Олександрович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело. Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.



## ЗМІСТ

ВСТУП .....	4
РОЗДІЛ 1 ВИВЧЕННЯ РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР У ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ .....	7
1.1. Аналіз предметної області.....	7
1.1.1. Основні компоненти шкільного курсу інформатики .....	7
1.1.2. Недоліки нинішньої системи навчання інформатики в школах України .	8
1.1.3. Актуальність включення розробки ігор у шкільний курс інформатики.	11
1.2. Вивчення розробки комп'ютерних ігор у системах освіти різних країн світу .....	14
1.2.1. Розвиток курсу розробки ігор у США.....	15
1.2.2. Інтеграція розробки ігор у європейську освіту .....	16
1.2.3. Азійський підхід до навчання розробці ігор.....	19
1.2.4. Ігрова освіта в країнах, що розвиваються. ....	21
1.3. Аналіз навчальних курсів з розробки комп'ютерних ігор ІТ-шкіл України	23
Висновки до першого розділу.....	26
РОЗДІЛ 2 МЕТОДИЧНА РОЗРОБКА ФАКУЛЬТАТИВНОГО КУРСУ З РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР ДЛЯ ПРОФІЛЬНОГО НАВЧАННЯ ІНФОРМАТИКИ УЧНІВ ЛІЦЕЇВ .....	27
2.1. Аналіз можливостей вивчення розробки комп'ютерних ігор у навчальних програмах старшої профільної школи .....	27
2.2. Опис методичного комплексу для підтримки факультативного курсу з розробки комп'ютерних ігор для профільного навчання .....	30
2.3. Рекомендації щодо використання навчально методичного комплексу в навчальному процесі.....	38
Висновки до другого розділу .....	49
ВИСНОВКИ .....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53

## ВСТУП

**Актуальність теми дослідження.** У сучасній Україні, яка активно інтегрується у світове цифрове середовище, розробка комп'ютерних ігор у шкільному курсі інформатики набуває особливого значення. Ця тема є актуальною через швидкий розвиток ІТ-індустрії в країні, яка сьогодні займає провідні позиції у світовому аутсорсингу програмного забезпечення. Геймдев стає важливою частиною цього сектору, пропонуючи молоді перспективи професійного розвитку, конкурентні зарплати та можливість створювати власні продукти, які можуть досягати міжнародного успіху.

Індустрія програмного забезпечення - це динамічна та орієнтована на ринок галузь, яка вимагає постійних інновацій та адаптації до мінливих потреб клієнтів та технологічних тенденцій [1]. Одним з найбільш відомих і прибуткових сегментів цієї індустрії є індустрія відеоігор, яка виробляє інтерактивні розважальні продукти, які звертаються до широкого кола аудиторій і платформ [2, 3].

Крім того, в умовах викликів, які стоять перед Україною, включення теми розробки ігор у шкільний курс дає можливість молоді залишатися конкурентоспроможною на світовому ринку праці. Участь у сучасному цифровому світі забезпечує країні потенціал для інноваційного розвитку, а також стимулює творчу активність серед підростаючого покоління.

В умовах, коли українська освіта реформується для підготовки учнів до викликів сучасного світу, інтеграція розробки комп'ютерних ігор у шкільний курс інформатики сприяє розвитку критично важливих навичок. Це, зокрема, логічне мислення, креативність, розуміння алгоритмів, вміння вирішувати комплексні задачі та працювати в команді. Учні не лише вивчають основи програмування, а й навчаються застосовувати отримані знання для створення реальних продуктів, що робить навчання більш цікавим і практично орієнтованим.

Враховуючи важливість та потенціал індустрії відеоігор, багато навчальних закладів, які готують інженерів-програмістів, включили розробку ігор як частину своєї навчальної програми. Основними мотиваціями для навчання розробки ігор є підвищення привабливості та ефективності навчання [4-8], підготовка випускників до конкурентної та вимогливої ігрової індустрії [9, 10], сприяння командній роботі та

навичкам співпраці [11, 12], а також розвиток можливостей управління проектами та вирішення проблем [4, 5]. Особливо важливою ця тема стає в контексті цифровізації української освіти та прагнення зробити її більш інтерактивною і такою, що відповідає світовим стандартам. Розробка комп'ютерних ігор дозволяє інтегрувати знання з інших дисциплін, таких як фізика, математика чи література, підтримуючи міждисциплінарний підхід до навчання. Це сприяє не лише технічному, а й гуманітарному розвитку учнів, формуючи всебічно освічену та мотивовану особистість.

Розробка комп'ютерних ігор у школах — це не лише про навчання програмуванню, а й про формування нового покоління українців, готових вирішувати складні задачі майбутнього.

Актуальність обраної проблеми спонукала нас до вибору теми роботи: «Факультатив з розробки комп'ютерних ігор для учнів ліцеїв».

**Метою дослідження** є теоретично обґрунтувати і розробити зміст факультативного курсу з розробки комп'ютерних ігор для учнів ліцеїв.

У відповідності з метою дослідження було сформовано **завдання**:

1. розглянути зміст шкільної програми з інформатики і з'ясувати можливе місце в ній розробки комп'ютерних ігор;
2. здійснити аналіз основних підходів та засобів навчання розробки комп'ютерних ігор в навчальних закладах світу та України;
3. розробити зміст факультативного курсу та надати методичні рекомендації щодо організації навчання розробки комп'ютерних ігор для учнів ліцеїв в профільному навчанні інформатики.

**Об'єкт дослідження** є навчання інформатики в профільних класах.

**Предмет дослідження** є факультативний курс з розробки комп'ютерних ігор для учнів ліцеїв.

Досягненню мети і розв'язанню поставлених завдань сприяло використання комплексу **методів дослідження**: теоретичні – вивчення та аналіз джерел інформації, наукових робіт провідних вітчизняних та зарубіжних дослідників, синтез, порівняння; емпіричні - (розробка навчально-методичного комплексу з розробки комп'ютерних ігор).

**Практична значущість**: розроблений факультативний курс з розробки

комп'ютерних ігор можна використовувати для навчання в профільних класах ліцеїв.

**Структура та обсяг роботи** – робота складається зі вступу, двох розділів, висновків, списку використаної літератури, що містить 38 найменувань. Основний зміст роботи викладено на 52 сторінках. Повний обсяг роботи складає 56 сторінок.

# РОЗДІЛ 1

## ВИВЧЕННЯ РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР У ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ

### 1.1. Аналіз предметної області

Шкільна програма з інформатики в Україні має на меті забезпечити учнів базовими знаннями й навичками роботи з комп'ютером, інформаційними технологіями та основами програмування. Ця програма створена відповідно до державних стандартів і покликана підготувати учнів до роботи в сучасному цифровому світі. Вона орієнтована на формування ключових компетенцій, таких як інформаційна грамотність, критичне мислення, цифрова безпека, а також базові технічні навички.

В старшій профільній школі учні можуть вивчати спеціалізовані теми, такі як бази даних, веброзробка, основи кібербезпеки та інформаційних систем. Програма передбачає реалізацію складніших проєктів, таких як вебсайти чи інтерактивні програми, що готують учнів до вступу у вищі навчальні заклади або роботи в IT-індустрії [13-15].

#### 1.1.1. Основні компоненти шкільного курсу інформатики

##### *Алгоритмізація та програмування*

Цей блок є основою для розвитку логічного мислення й розуміння принципів роботи комп'ютерних програм. У молодших класах акцент робиться на простих візуальних середовищах, таких як Scratch, які дозволяють створювати програми у формі ігрових сценаріїв. Учні складають алгоритми у вигляді блоків, які легко взаємодіють між собою. У старших класах програмування переходить на більш складний рівень із використанням мов, таких як Python або Pascal, що дає змогу розробляти більш серйозні алгоритми та знайомитися з основами структурного й об'єктно-орієнтованого програмування.

##### *Робота з текстовими, графічними та табличними редакторами.*

Учні вивчають інструменти Microsoft Office (Word, Excel, PowerPoint) або їхні аналоги (Google Docs, LibreOffice), які є базовими для роботи з текстами, таблицями й створенням презентацій. Ці навички є важливими для підготовки реальних

документів і проєктів у навчанні та майбутній професійній діяльності.

#### *Інтернет та цифрова безпека.*

Окрему увагу приділено вмінню використовувати інтернет для пошуку інформації, роботи з електронною поштою й спілкування у цифровому середовищі. Учням викладаються основи цифрової безпеки, такі як захист паролів, уникнення шкідливого програмного забезпечення, правила етичного використання інформації та безпечне поведіння в соціальних мережах.

#### *Основи апаратного забезпечення*

Учні отримують базові знання про будову комп'ютера, основні компоненти (процесор, пам'ять, накопичувачі, периферійні пристрої) і їхнє функціонування. Це допомагає зрозуміти, як апаратна частина взаємодіє з програмним забезпеченням.

#### *Використання спеціалізованого програмного забезпечення*

Окрім стандартних офісних програм, учнів знайомлять із програмами для обробки зображень (Photoshop, GIMP), відеомонтажу (Movavi, Adobe Premiere) і створення анімації. Це забезпечує початкові навички роботи з мультимедійним контентом.

За думкою Н. Морзе, Т. Нанаєвої та О. Пасічник [16] незважаючи на переваги навчальної програми з інформатики в старшій школі, як-от: модульність програми, можливість побудови гнучкої траєкторії навчання, практична орієнтованість модулів, вона має досить велику кількість недоліків. Зокрема, на думку опитуваних представників ІТ-компаній, таким недоліком є слабкі варіативні курси з програмування, а на думку вчителів – необхідно більше вибіркових модулів (різної тематики та складності).

#### 1.1.2. Недоліки нинішньої системи навчання інформатики в школах України.

Сучасна система навчання інформатики в Україні має значний потенціал, але водночас вона стикається з низкою проблем, які обмежують її ефективність та актуальність для сучасного покоління. Ці недоліки пов'язані як з методологією викладання, так і з матеріально-технічною базою та організаційними аспектами.

#### *Недостатня практична складова*

Однією з основних проблем є надмірна теоретизація навчання. Значна частина уроків інформатики присвячується вивченню теоретичних аспектів, таких як будова комп'ютера, принципи роботи операційних систем чи історія розвитку



інформаційних технологій. Хоча ці теми важливі, вони подаються без достатнього практичного підкріплення. Учні часто не мають можливості застосувати ці знання в реальних умовах, що знижує їх інтерес до дисципліни.

Наприклад, програмування часто викладається як набір правил і алгоритмів, але без створення реальних проєктів, таких як ігри, вебсайти чи програми, учні не розуміють, як ці навички можна використовувати у житті чи майбутній професії.

#### *Застаріла матеріально-технічна база*

Багато шкіл, особливо у сільській місцевості чи невеликих містах, мають застаріле обладнання, яке не відповідає сучасним вимогам. Старі комп'ютери, відсутність сучасного програмного забезпечення та нестабільний доступ до інтернету значно обмежують можливості для якісного навчання.

Наприклад, у багатьох школах комп'ютери можуть бути занадто слабкими для запуску сучасних середовищ розробки, таких як Unity або навіть простіших програм для створення графіки та анімації. Це також унеможлиблює навчання роботі з сучасними хмарними сервісами, базами даних чи інструментами для командної роботи.

#### *Відсутність інтеграції сучасних методик*

Програма інформатики в більшості випадків побудована за застарілими методологіями, які не враховують сучасні підходи до навчання, такі як проєктно-орієнтоване навчання, гейміфікація або інтеграція міжпредметних зв'язків. Наприклад, розробка комп'ютерних ігор або створення вебсайтів може об'єднувати знання з математики, інформатики, графіки та навіть літератури, але такі підходи майже не використовуються.

Відсутність акценту на інноваційні та креативні завдання, які мають безпосередній зв'язок із реальним життям, знижує зацікавленість учнів у предметі. Замість цього навчання часто обмежується виконанням шаблонних вправ із навчальних посібників.

#### *Невідповідність ринку праці*

Шкільна програма не завжди орієнтована на реальні вимоги сучасного ринку праці. Навчання часто зосереджене на базових навичках роботи з офісними програмами (Word, Excel, PowerPoint), які, хоч і є важливими, вже не відповідають вимогам більшості сучасних професій у сфері ІТ. Знання про алгоритми чи

програмування викладаються у відриві від практичних застосувань, таких як розробка додатків, ігор чи роботи з базами даних.

Це призводить до того, що навіть ті учні, які зацікавлені в інформатиці, часто не отримують необхідного старту для побудови кар'єри в ІТ, і їм доводиться опановувати потрібні навички самостійно або на спеціалізованих курсах.

#### *Обмежена підготовка вчителів*

Ще однією серйозною проблемою є рівень підготовки вчителів інформатики. Багато педагогів отримали освіту за старими стандартами і часто не мають практичного досвіду роботи з сучасними технологіями. У результаті вони можуть викладати предмет за застарілими методами, використовуючи матеріали, які давно втратили актуальність.

Наприклад, викладачі можуть не мати знань чи навичок у роботі з сучасними мовами програмування (Python, JavaScript) або інструментами для створення ігор і анімації (Unity, Game Maker). Це створює розрив між очікуваннями учнів, які прагнуть сучасного підходу до навчання, і можливостями викладачів.

#### *Обмежений доступ до інтернету*

У багатьох школах проблема із доступом до якісного інтернету значно обмежує можливості для інтеграції сучасних освітніх ресурсів. Інтернет є важливим інструментом для навчання інформатики, оскільки дозволяє використовувати онлайн-курси, відеоуроки, інтерактивні платформи та інші інструменти. Відсутність стабільного підключення позбавляє учнів можливості працювати з актуальними матеріалами й технологіями.

#### *Відсутність адаптації під індивідуальні потреби*

Сучасна система навчання інформатики майже не враховує різний рівень підготовки та зацікавленості учнів. Деякі школярі вже мають базові навички програмування або графічного дизайну до початку навчання, тоді як інші лише знайомляться з основами роботи за комп'ютером. Нинішня програма пропонує однаковий підхід для всіх, що часто стає причиною втрати мотивації серед учнів із різним рівнем знань.

#### *Брак інтерактивності та креативності*

Викладання інформатики часто зводиться до сухого подання матеріалу, що не залучає учнів до активної діяльності. Відсутність інтерактивних методів навчання,

таких як командні проекти, конкурси програмування чи створення власних проєктів, робить уроки нецікавими для школярів.

Недоліки нинішньої системи навчання інформатики в українських школах є багатогранними та стосуються як методології, так і організаційних аспектів. Для підвищення ефективності викладання важливо адаптувати програму до сучасних вимог, оновити технічну базу, підвищити рівень підготовки вчителів і зробити акцент на практичних, інтерактивних і творчих підходах до навчання. Інтеграція таких напрямів, як розробка комп'ютерних ігор, може стати вирішенням багатьох із цих проблем, зробивши предмет більш цікавим і корисним для учнів.

### 1.1.3. Актуальність включення розробки ігор у шкільний курс інформатики.

Інтеграція розробки комп'ютерних ігор у шкільний курс інформатики є надзвичайно актуальною з кількох причин, що обумовлені як глобальними тенденціями розвитку інформаційних технологій, так і специфічними потребами сучасної системи освіти. Ця тема об'єднує одразу кілька важливих напрямків: розвиток технічних навичок, стимулювання креативності, інтеграцію міждисциплінарного підходу та підготовку учнів до реалій сучасного ринку праці.

#### *Відповідність сучасним технологічним тенденціям*

У світі, де цифрові технології стають невід'ємною частиною повсякденного життя, інтеграція таких складових, як розробка комп'ютерних ігор, дозволяє учням опанувати актуальні технічні навички. Розробка ігор вимагає використання сучасних мов програмування (Python, C#, JavaScript), знань у сфері графіки, анімації та навіть штучного інтелекту. Таким чином, учні отримують не лише базові знання інформатики, але й практичні навички, які можуть стати основою для подальшого розвитку в ІТ-сфері.

Глобальний ринок відеоігор демонструє стрімке зростання: згідно з даними аналітичних компаній, індустрія ігор перевищує кіно та музику за доходами. Участь у цьому ринку потребує знань і навичок, які можуть бути закладені ще у шкільні роки через створення базових ігрових проєктів.

#### *Підвищення зацікавленості учнів*

Розробка ігор відповідає інтересам сучасного покоління учнів, які з раннього віку знайомі з комп'ютерними та мобільними іграми. Викладання інформатики через

створення ігор створює потужний мотиваційний ефект, адже учні бачать практичний результат своєї роботи в реальному часі. Вони можуть створювати персонажів, будувати рівні, додавати анімації та взаємодії, що стимулює їх до навчання.

Така активність не лише робить навчання захоплюючим, а й сприяє розвитку впевненості у своїх силах. Учні відчувають задоволення від створення власного продукту, який можна продемонструвати друзям, вчителям або навіть на конкурсах.

### *Розвиток міждисциплінарного мислення*

Розробка ігор є міждисциплінарною діяльністю, яка об'єднує знання з різних галузей:

- математика: для обчислення траєкторій, симуляції фізичних процесів, роботи з алгоритмами;
- фізика: для створення реалістичних взаємодій у грі, таких як гравітація чи зіткнення об'єктів;
- мистецтво: для створення графічного дизайну, персонажів та ігрових локацій;
- література та сценарна майстерність: для написання сюжету гри та діалогів персонажів;
- інформатика: для програмування, розробки алгоритмів та роботи з ігровими двигунами.

Така інтеграція дозволяє учням застосовувати знання з інших предметів у новому контексті, що сприяє глибшому розумінню матеріалу та підвищує їхню зацікавленість у навчанні загалом.

### *Формування корисних навичок*

Сучасна освіта повинна відповідати потребам ринку праці, на якому зараз цінуються різноманітні навички, зокрема:

- креативність: розробка ігор стимулює творчість, адже учні самостійно створюють ігровий світ, персонажів, механіки гри;
- критичне мислення: для розв'язання проблем, які виникають під час розробки (наприклад, оптимізація коду чи усунення багів);
- навички командної роботи: розробка ігор часто вимагає спільної роботи в групах, що допомагає розвивати комунікаційні навички;
- технічна грамотність: учні вчаться працювати з сучасними

інструментами, такими як Unity, Unreal Engine, Game Maker.

Усе це сприяє підготовці школярів до сучасного ринку праці, де такі навички стають обов'язковими.

### *Підготовка до кар'єри в IT та творчих індустріях*

Інтеграція розробки ігор у шкільну програму може стати першим кроком до вибору кар'єри в IT-сфері чи креативних індустріях. Навички, отримані під час створення ігор, є універсальними й можуть бути застосовані у багатьох професіях: програмування, дизайн, анімація, продюсування ігор, тестування.

Багато випускників шкіл прагнуть розвиватися у сфері IT, але не завжди знають, з чого почати. Шкільний курс із розробки ігор може стати стартовим майданчиком для знайомства з галуззю.

### *Вирішення проблем нинішньої системи освіти*

Розробка ігор може стати відповіддю на основні недоліки сучасного викладання інформатики. Замість надмірної теоретизації, учні отримують можливість працювати над реальними проєктами, які вимагають застосування їхніх знань на практиці. Такий підхід допомагає розвинути навички критичного мислення, самостійності та впевненості у своїх силах.

До того ж, створення ігор допомагає адаптувати курс інформатики до інтересів учнів, що робить навчання більш інклюзивним та доступним для школярів із різними рівнями підготовки.

### *Можливості для позакласної діяльності*

Крім інтеграції в основний курс інформатики, розробка ігор може стати основою для позакласних гуртків та змагань. Учні можуть брати участь у конкурсах розробки ігор, таких як Game Jam, що дає їм можливість отримати визнання, знайти однодумців та розвивати свої проєкти.

Інтеграція розробки комп'ютерних ігор у шкільну програму інформатики є надзвичайно актуальною та перспективною. Вона дозволяє вирішувати одразу кілька завдань: підвищення зацікавленості учнів, розвиток технічних та креативних навичок, інтеграцію міждисциплінарного підходу, підготовку до майбутньої кар'єри. Такий підхід відповідає потребам сучасної освіти та дозволяє створити фундамент для розвитку нового покоління фахівців, які здатні креативно та ефективно використовувати технології у різних сферах життя.

Розробка комп'ютерних ігор може стати відмінним доповненням до існуючої програми. Вона дозволяє інтегрувати знання з алгоритмів, графіки, дизайну та програмування, пропонуючи учням захопливу й практично орієнтовану діяльність. Цей підхід сприятиме розвитку креативності, критичного мислення та командної роботи, а також зробить навчання інформатики більш актуальним і цікавим для сучасного покоління.

Однак навчання розробці ігор не обходиться без викликів і труднощів. Деякі з поширених бар'єрів, які перешкоджають інтеграції курсів розробки ігор у навчальні програми, це відсутність міждисциплінарних навичок, обмеження часу, недостатній інтерес та досвід серед викладачів, а також сприйняття того, що розробка ігор не є серйозною академічною темою [16, 17].

Поточна ситуація дозволяє реалізувати курс з розробки комп'ютерних ігор лише у вигляді факультативу.

Факультатив (лат. *facultatis* — необов'язковий) — навчальний курс, не обов'язковий для відвідування.

Факультативні заняття впроваджуються в шкільну практику як форма диференційованого навчання. Для факультативів розробляються авторські варіанти програм, навчальні посібники.

Практичні факультативні заняття проводять для формування пошукових навичок і вмінь у процесі розв'язування технічних задач. Структура практичних факультативів передбачає постановку завдань, обґрунтування їх актуальності, практичного значення; розв'язування задач та конкретизацію результатів роботи; обговорення результатів, підбиття підсумків заняття.

## **1.2. Вивчення розробки комп'ютерних ігор у системах освіти різних країн світу**

Розвиток інформаційних технологій і стрімке поширення комп'ютерних ігор сприяли зростанню інтересу до включення розробки ігор у навчальні програми в різних країнах світу. Ігрова індустрія стала не лише сферою розваг, але й джерелом інновацій, досліджень та розвитку креативних навичок. Сучасна освіта враховує ці тренди, адаптуючи свої методи та програми для інтеграції розробки ігор як засобу підготовки учнів до майбутніх викликів.

### 1.2.1. Розвиток курсу розробки ігор у США.

У Сполучених Штатах Америки розробка комп'ютерних ігор зайняла важливе місце в освітніх програмах на всіх рівнях — від початкової школи до університетів. Це пов'язано не лише зі зростанням популярності ігрової індустрії, але й із прагненням розвивати у студентів навички, які відповідають вимогам сучасного технологічного ринку. США вважаються піонерами в інтеграції розробки ігор в освіту, що обумовлено декількома ключовими факторами.

#### *Університетські програми*

Американські університети одними з перших почали впроваджувати спеціалізовані програми з розробки комп'ютерних ігор. Наприклад, National High School Game Academy Карнегі-Меллонського університету [18], MIT Game Lab Массачусетського технологічного інституту [19] та Університет Південної Каліфорнії пропонують курси, що охоплюють всі аспекти створення ігор — від програмування до дизайну та психології гравця. Програми націлені на інтеграцію таких дисциплін, як графіка, штучний інтелект, наративний дизайн, фізика та соціальні аспекти ігрового процесу.

Особливістю таких програм є акцент на проектному підході: студенти працюють у командах над реальними ігровими проектами, використовуючи професійні інструменти, такі як Unity або Unreal Engine. Це дозволяє їм не тільки отримати теоретичні знання, але й практичний досвід, який високо цінується в індустрії.

#### *Курси для середньої школи*

У середніх школах США все більше уваги приділяється розвитку STEM-навичок (наука, технології, інженерія та математика). Розробка комп'ютерних ігор стала ідеальним інструментом для залучення школярів до технічних дисциплін. Наприклад, через популярні платформи, такі як GameMaker або Scratch, школярі вивчають основи програмування, логіку, дизайн та творче мислення.

Існують програми, розроблені спеціально для того, щоб зробити навчання цікавим і мотивуючим. Наприклад, організація Code.org підтримує школи у впровадженні навчальних планів, які базуються на створенні простих ігор, що допомагають учням засвоювати концепції кодування. Це сприяє залученню дітей різного віку до навчання технологіям у ігровій формі.

### *Літні табори та позашкільні програми*

В США популярні літні табори та позашкільні програми, які спеціалізуються на розробці комп'ютерних ігор. Наприклад, табори iD Tech Camps пропонують курси з дизайну ігор, візуалізації 3D-графіки та програмування. Ці програми розроблені для дітей віком від 7 до 18 років і дають можливість не лише розвивати технічні навички, а й працювати над спільними проектами у творчій атмосфері.

Також існують ініціативи на кшталт "Girls Who Code", які спрямовані на залучення дівчат до розробки ігор та подолання гендерного розриву в технологічній галузі.

У США освіта з розробки ігор підтримується культурною значимістю цієї індустрії. Вона розглядається не лише як спосіб створення розваг, але й як платформа для вирішення соціальних проблем, створення освітніх інструментів та розвитку критичного мислення. Наприклад, ігри використовуються у проектах для вивчення історії, розвитку емпатії або популяризації наукових концепцій.

Розвиток ігрових програм активно підтримується державними ініціативами. Уряд США через Національний науковий фонд (NSF) та Міністерство освіти фінансує розробку навчальних матеріалів, програм та інструментів для інтеграції ігрового навчання в освітній процес.

Одним із таких проєктів є STEM Video Game Challenge, де школярі та студенти змагаються у створенні освітніх відеоігор, які потім можуть використовуватися в навчальних закладах.

Розробка комп'ютерних ігор у США — це не просто навчання програмуванню чи дизайну. Це багатогранний підхід, що поєднує технології, творчість, соціальну взаємодію та критичне мислення. Успішна інтеграція ігрової розробки в освітні програми сприяє формуванню компетенцій майбутнього, таких як здатність до інновацій, командна робота та адаптація до змін. США демонструють, як ігрова освіта може бути не лише цікавою, а й ефективною інвестицією в майбутнє покоління.

#### 1.2.2. Інтеграція розробки ігор у європейську освіту.

У Європі інтеграція розробки комп'ютерних ігор в освітні програми отримала значну підтримку як на державному рівні, так і в рамках ініціатив приватних освітніх установ. Комп'ютерні ігри розглядаються як ефективний інструмент розвитку



технічних навичок, творчого мислення, міждисциплінарного навчання та вирішення реальних проблем. Європейські країни активно впроваджують різноманітні стратегії, адаптуючи розробку ігор до специфіки своїх освітніх систем.

Університети Європи, зокрема у Великій Британії, Німеччині, Фінляндії, Швеції та Нідерландах, стали лідерами у впровадженні спеціалізованих програм з розробки ігор. Ці програми мають інноваційний підхід до навчання, комбінуючи теорію, практику і креативність.

Велика Британія: Університети, такі як University of Abertay Dundee і Staffordshire University, пропонують бакалаврські та магістерські програми з розробки ігор. Велика увага приділяється практичному навчанню, де студенти створюють ігри в командному середовищі, симулюючи реальні умови ігрової індустрії.

Фінляндія: Країна відома своїм інноваційним підходом до освіти. Університети, як-от Tampere University, пропонують курси з геймдизайну, які інтегрують елементи психології гравця, візуального мистецтва та програмування. Ці курси орієнтовані на розвиток навичок, що виходять за межі ігрової індустрії, зокрема на вирішення соціальних проблем через ігри.

Швеція: Університети Швеції, такі як Gotland University, стали центрами навчання з розробки ігор, пропонуючи програми з акцентом на 3D-анімацію, дизайн рівнів та сценарії. Навчання передбачає активну співпрацю з провідними компаніями індустрії, що забезпечує студентам можливість брати участь у реальних проектах.

У європейських школах розробка ігор активно впроваджується як частина програм STEM (наука, технології, інженерія, математика). Багато країн інтегрують навчання ігровому кодуванню у шкільну програму, використовуючи інтуїтивні платформи, такі як Scratch, Unity та GameMaker.

Велика Британія стала однією з перших країн, які включили програмування в національний навчальний план початкових і середніх шкіл. Використання розробки ігор для викладання програмування допомагає зробити цей процес цікавим і доступним для учнів.

Школи у Франції впроваджують інтерактивне навчання з акцентом на створення ігор для розвитку логічного мислення. Використання спеціалізованих платформ, таких як Blockly, дозволяє школярам освоювати основи програмування у

цікавій формі.

Фінляндія відома своєю відкритістю до експериментів в освіті. Викладання розробки ігор у середніх школах спрямоване не лише на технічну підготовку, а й на розвиток креативності та командної роботи.

У Європі значну увагу приділяють професійній підготовці у сфері розробки ігор. Технічні коледжі пропонують короткострокові курси та сертифікатні програми спрямовані на навчання основам програмування, дизайну та анімації.

В Німеччині Інститути професійної підготовки, такі як Games Academy в Берліні, зосереджуються на практичних аспектах розробки ігор, що дозволяє студентам швидко влитися в індустрію.

Данія активно інвестує у професійні програми, орієнтовані на підготовку спеціалістів з геймдизайну, підтримуючи партнерства між навчальними закладами та ігровими студіями.

Європейські літні школи, такі як European Game Developers Summer School, пропонують курси з геймдизайну для студентів і молодих фахівців. Це інтенсивні програми, які охоплюють всі етапи створення ігор — від концептуального дизайну до випуску продукту.

Позашкільні програми, такі як GameCamp у Польщі чи Coding Pirates у Данії, забезпечують дітей і підлітків можливістю освоїти розробку ігор у дружній атмосфері, сприяючи розвитку їхніх навичок через практичну діяльність.

Європейський Союз підтримує ініціативи з інтеграції розробки ігор в освітні системи через програми фінансування, такі як Erasmus+ та Horizon Europe. Ці проекти націлені на розробку навчальних матеріалів, платформ та інструментів для використання ігор у навчанні.

Наприклад, програма Games in Schools, ініційована Європейською асоціацією індустрії відеоігор (ISFE), спрямована на навчання вчителів використанню ігор у класах для розвитку творчого та критичного мислення.

Європейський досвід інтеграції розробки ігор в освіту демонструє ефективність цього підходу у підготовці молоді до викликів сучасного світу. Програми з розробки ігор сприяють не лише технічному навчанню, а й розвитку креативності, критичного мислення та соціальної відповідальності. Європа пропонує гнучкі та інноваційні підходи до навчання, які можуть слугувати прикладом для

інших регіонів світу.

### 1.2.3. Азійський підхід до навчання розробці ігор.

Азійські країни, завдяки своїм сильним позиціям у технологічній індустрії та культурному впливу через ігрові продукти, активно впроваджують навчання розробці комп'ютерних ігор у свої освітні системи. Такий підхід базується на поєднанні традиційних цінностей з інноваційними технологіями, що забезпечує потужний поштовх для розвитку як технічних, так і творчих навичок серед студентів.

У багатьох азійських країнах навчання розробці ігор є важливою складовою університетських програм, спрямованих на підготовку професіоналів для глобального ринку.

Японія є лідером у галузі відеоігор, і університети країни, такі як Токуо Polytechnic University та Kyoto Seika University, пропонують спеціалізовані програми з розробки ігор. У цих програмах особлива увага приділяється креативності, культурним аспектам та технічній майстерності. Студенти вивчають усі аспекти створення ігор, включаючи дизайн, програмування, анімацію та сценаристику.

Південна Корея відома своєю конкурентоспроможною ігровою індустрією, де такі компанії, як Nexon і NCSoft, є світовими лідерами. Корейські університети, такі як KAIST (Korea Advanced Institute of Science and Technology), впроваджують міждисциплінарні програми, що поєднують інженерію, штучний інтелект та ігровий дизайн. Крім того, корейські студенти отримують практичний досвід через стажування у великих ігрових компаніях.

Китай інвестує значні ресурси в розвиток геймдеву, а університети, як-от Tsinghua University та Communication University of China, пропонують програми, які акцентують увагу на мобільних іграх та віртуальній реальності. Китайські навчальні заклади часто співпрацюють із міжнародними студіями, такими як Tencent Games, для забезпечення практичної підготовки студентів.

Сінгапур вважається освітнім центром Азії, і його університети, такі як DigiPen Institute of Technology та Nanyang Technological University, надають світового рівня програми з розробки ігор. У Сінгапурі велика увага приділяється інноваціям та інтерактивному дизайну, що є привабливим і залучає студентів зі всього світу.

В Азії розробка комп'ютерних ігор поступово входить до програм середньої школи, спрямовуючи учнів до раннього засвоєння технічних та творчих навичок.

Південна Корея активно інтегрує навчання програмуванню та розробці ігор у школи через державні ініціативи. Уроки програмування на базі ігрових платформ, таких як Scratch та Unity, допомагають учням розвивати логічне мислення та розуміння програмування.

Японія у шкільних програмах приділяє увагу розвитку творчих здібностей через дизайн ігор. Учні створюють прості 2D-ігри, використовуючи такі інструменти, як RPG Maker, що робить навчання доступним і захоплюючим.

Індія розвиває концепцію геймдеву через позашкільну освіту. Спеціалізовані школи, такі як WhiteHat Jr., навчають дітей основ програмування та створення ігор на ранніх етапах їхнього навчання.

Азартні студії та навчальні центри в Азії пропонують програми позашкільної освіти для дітей та дорослих, орієнтовані на створення ігор. Вони часто використовують гнучкі формати навчання:

Китай має численні приватні навчальні центри, які готують молодь до роботи в геймдеві, включаючи Game Academy Beijing. Ці програми часто проводяться у вихідні, що дозволяє поєднувати навчання зі шкільною чи університетською освітою.

Малайзія та Філіппіни активно організовують літні табори та воркшопи, де учасники розробляють свої ігрові проекти, використовуючи платформи, як-от GameMaker.

У багатьох азійських країнах державні органи активно інвестують у розвиток ігрової індустрії, а також у навчальні програми, спрямовані на підготовку спеціалістів у цій галузі.

Південна Корея має національну програму «Креативна економіка», яка включає субсидії для шкіл і університетів, що пропонують курси з розробки ігор. Державні гранти стимулюють створення навчальних лабораторій та студій при навчальних закладах.

Китай активно фінансує освітні ініціативи через програми, спрямовані на розвиток VR- і AR-технологій у навчанні, а також підтримує створення молодіжних стартапів у сфері ігор.

Незважаючи на значні досягнення, азійський підхід до інтеграції розробки ігор стикається з такими викликами, як нестача кваліфікованих викладачів, високі витрати на обладнання та необхідність адаптації навчальних програм до потреб

індустрії. Проте, зважаючи на швидке зростання ігрового ринку та зацікавленість держав, ці труднощі активно вирішуються.

У перспективі Азія може стати світовим центром освітніх інновацій у сфері геймдеву, поєднуючи високий рівень технічної підготовки з унікальним культурним підходом.

#### 1.2.4. Ігрова освіта в країнах, що розвиваються.

Ігрова освіта в країнах, що розвиваються, поступово стає важливим компонентом освітніх реформ, спрямованих на підвищення якості навчання, розвиток цифрових навичок та підготовку молоді до вимог сучасного ринку праці. Цей підхід набуває популярності завдяки своїй ефективності у залученні студентів, доступності інструментів для створення ігор та перспективам розвитку індустрії цифрових розваг у глобальному масштабі.

В умовах, коли в багатьох країнах, що розвиваються, освітні системи стикаються з обмеженими ресурсами, ігрова освіта виступає як ефективний інструмент, що дозволяє досягти кількох важливих цілей:

- розвиток цифрової грамотності. Комп'ютерні ігри та їх розробка допомагають студентам освоїти базові та просунуті навички роботи з технологіями, включаючи програмування, графіку, дизайн та командну роботу;
- підвищення мотивації до навчання. Використання ігрових елементів у навчальних програмах дозволяє зробити освітній процес більш цікавим і залучаючим, що є особливо важливим у контексті країн, де навчання часто базується на традиційних методах, таких як зазубрювання;
- підготовка до роботи в індустрії цифрових розваг. Розробка ігор відкриває можливості для розвитку підприємництва та створення нових робочих місць у сферах технологій, що є ключовим напрямком економічного зростання для багатьох країн.

Індія, як одна з провідних країн, що розвиваються, активно інтегрує розробку ігор у свою освітню систему. Ініціативи, такі як програма WhiteHat Jr., навчають дітей основам програмування через створення простих ігор. Уряд Індії також підтримує технологічні стартапи, зокрема у сфері ігрових технологій, що сприяє поширенню цих знань серед молоді.

У Кенії розробка ігор використовується для вирішення соціальних проблем. Наприклад, ігри, створені студентами, часто мають просвітницький характер і присвячені темам охорони здоров'я, екології чи прав людини. Навчальні програми підтримуються організаціями, такими як Akirachix, які фокусуються на залученні молоді та жінок у сферу технологій.

Нігерія також демонструє зростання інтересу до ігрової освіти. Стартапи, такі як Kucheza Gaming, проводять освітні заходи, орієнтовані на навчання дітей створенню ігор, а також на використання ігрових платформ для розвитку когнітивних навичок.

У Південно-Східній Азії країни, такі як Філіппіни та Індонезія, активно розвивають освітні програми, що поєднують традиційне навчання з ігровими елементами. Навчальні програми включають створення ігор, які відповідають місцевим культурним та історичним контекстам, що сприяє збереженню національної ідентичності через технології.

Попри очевидні переваги, ігрова освіта стикається з низкою викликів у країнах, що розвиваються:

- недостатній доступ до технологій. У багатьох регіонах спостерігається дефіцит комп'ютерів, програмного забезпечення та інтернету, що ускладнює впровадження таких програм.
- обмежена кількість кваліфікованих викладачів. Відсутність викладачів, які мають досвід у розробці ігор, є значною перешкодою для інтеграції таких програм у шкільну чи університетську освіту.
- економічні труднощі. Навчальні заклади часто мають обмежений бюджет, що не дозволяє інвестувати в сучасні інструменти для розробки ігор.
- суспільні упередження. У деяких країнах комп'ютерні ігри досі сприймаються виключно як розвага, а не як освітній інструмент, що може зменшувати інтерес до інтеграції геймдеву у навчальні програми.

Зважаючи на швидкий розвиток технологій, ігрова освіта в країнах, що розвиваються, має значний потенціал для подальшого зростання. Підтримка з боку міжнародних організацій, таких як ЮНЕСКО чи Світовий банк, допомагає подолати технологічні та фінансові бар'єри.

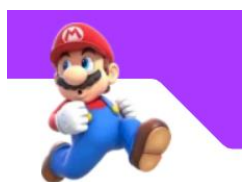
Ігрова освіта в країнах, що розвиваються, є важливим елементом сучасних

освітніх реформ, який сприяє цифровій трансформації, підвищенню мотивації учнів та підготовці до роботи у високотехнологічних галузях. Незважаючи на труднощі, ініціативи в таких країнах, як Індія, Кенія та Нігерія, демонструють, що правильний підхід до інтеграції геймдеву в освіту може суттєво покращити якість навчання та розкрити творчий потенціал молоді.

### 1.3. Аналіз навчальних курсів з розробки комп'ютерних ігор ІТ-шкіл України

Широку популярність останнім часом в Україні мають приватні ІТ-школи різного рівня, які серед іншого пропонують курси з розробки комп'ютерних ігор на основі ігрових рушіїв і інших засобів розробки.

Наприклад, школа Robocode [20] пропонує декілька курсів на базі рушія Unity:



Construct

#### *Знайомство зі світом ігор*

Учні познайомляться зі складним, але водночас захопливим світом створення ігор. Вони зрозуміють, як саме працюють ігри і навчаться самостійно створювати неймовірні світи, які існують за їх власними правилами.

#### *Розробка ігор в програмі Visual Scripting*



Unity Beginner

В ході курсу учні заглибляться у специфіку та особливості процесу розробки ігор. Спочатку ми ознайомимо їх із більш складними абстрактними поняттями, після чого вони перейдуть від блокового програмування до написання коду вручну мовою програмування C#.

#### *Розробка ігор в програмі Unity 2D*

Unity 2D

Цей курс закладе фундамент для подальшого розвитку в галузі програмування: проектування та створення своїх ігор на популярному движку Unity 3D, який активно використовується такими компаніями-гігантами як Blizzard, EA і Ubisoft. Водночас вивчатимуться основи алгоритмізації, ООП (об'єктно-орієнтоване програмування) і мова програмування C#.

### *Розробка ігор в програмі Unity 3D*

## Unity 3D

Курс дасть знання, які допоможуть розробляти повноцінні однокористувацькі 3D ігри, портувати їх під Android і публікувати в Play Market. Крім движка Unity, ми сфокусуємося на програмуванні мовою C#, поглибленні в ООП і роботі з базами даних для зберігання призначених для користувача даних.

### *Професійна розробка ігор у Unity 3D*

## Unity Pro

Ми розглянемо такі важливі питання, як інтеграція реклами в додатках, аналіз даних від користувачів (отримання повідомлень про помилки, що виникають в іграх), збір даних про функціонування додатків на пристроях клієнтів — все це реалізовуватиметься за допомогою Google Firebase.

Курс ProUnityЗначна частина курсу також буде присвячена роботі з мультиплеєром на юніті. Діти навчатимуться створювати багатокористувацькі ігри, в які можна грати у мережі разом із друзями.

Школа IT-generation [21] пропонує курс, завданнями якого є: навчити дитину основам роботи на багатоплатформеному русії №1 у світі, допомогти не боятись нових можливостей, навчити основам мови C#, дати дитині створити свою першу гру та допомогти випрацювати впевненість у своїх силах.

Школа GO-ITeens [22] пропонує трьохсеместровий курс, який складається з:

### *1 семестр*

Алгоритмізація та програмування. Основи програмування мовою C#. Типи даних. Умовні вирази та умовні конструкції if/else. Конструкція switch/case. Типові задачі з програмування. Цикли, масиви та багатовимірні масиви. Функції та перевантаження функції. Рекурсія та абстракції

### *2 семестр*

Знайомство з середовищем та архітектурою Unity. Поняття структур та значень, типи значень та посилальні типи. Основи об'єктно орієнтованого програмування. Написання скриптів, фізична та логічна взаємодія. Colliders, trigger,



rigidbody, raycasts. Система кешування PlayerPrefs. Робота з контентом та UI. Анімація, звук, камера, світло та матеріали. Системи часток

### *3 семестр*

Лабораторія ігор. Робота в команді. Концепція гри, ідея, жанр, сюжет, рівні. Прототипування гри. Організація всіх процесів, необхідних для розробки гри

Кожен семестр закінчується розробкою власного проєкту.

Навчальний центр «ІТ-Столиця» [23] пропонує курс «Розробка ігор на Unity 3D і C#» наступного змісту:

Урок 1. Знайомство з C# і написання першої програми

Урок 2. Додавання можливості вибору за допомогою умовних операторів

Урок 3. Повторюваний код з використанням циклів

Урок 4. Стиль коду і його характеристики

Урок 5. Основи програмування

Урок 6. Моделювання систем і об'єктів в середовищі Unity 3D

Урок 7. Використання матеріалів при створенні сцени

Урок 8. Створення моделей в Unity 3D

Урок 9. Ігрові об'єкти і їх взаємодії

Урок 10. Створення графічного інтерфейсу

Урок 11. Unity 3D і анімація

Урок 12. Створення першої гри із застосуванням Unity 3D і C#

Одним з найпоширеніших інструментів, на базі якого будуються навчальні курси, є ігровий рушій Unity, який дозволяє дітям втілити свої креативні ідеї в інтерактивні ігри та візуальні проєкти. Вони можуть створювати свої унікальні світи, персонажів та сюжети, розвивати уяву та дизайнерські навички [24].

Робота з Unity – це також розвиток логічного та аналітичного мислення. Діти навчаються розбиратися в алгоритмах, логічних умовах та програмуванні, що допомагає розвивати їхню здатність до вирішення проблем та складних завдань.

Вміння розробляти ігри та різні проєкти відкриває двері у світ інформаційних технологій. Вони матимуть цінні навички програмування, дизайну та роботи з технологічними інструментами, що може стати основою для вибору професії не лише в Україні, а й за її межами.

## **Висновки до першого розділу**

Розробка комп'ютерних ігор є однією з найдинамічніших галузей сучасних цифрових технологій, що відкриває широкі можливості для професійного розвитку та творчої самореалізації. Інтеграція цього напрямку в освітній процес старшої школи є актуальною через необхідність адаптації освіти до вимог сучасного світу, де цифрові компетенції стають ключовими для успішної кар'єри. Запровадження курсів з розробки ігор дозволяє не лише розвивати технічні навички учнів, а й формувати в них креативність, критичне мислення та вміння працювати в команді.

Проаналізовані технічні та методичні аспекти впровадження курсу демонструють значний потенціал для розвитку української системи освіти. Використання інструментів, таких як Unity, дає можливість учням опанувати сучасні технології, створювати власні проекти та отримувати досвід, який можна застосувати на практиці. Крім того, увага до міждисциплінарного підходу сприяє поглибленню знань із математики, фізики, літератури та інших дисциплін.

Важливо також зазначити, що такий курс допомагає підвищити мотивацію учнів завдяки інтеграції їхніх захоплень у навчальний процес. Використання ігор як основи для навчання дозволяє не лише зацікавити школярів, але й надати їм відчуття реального результату, який вони можуть продемонструвати оточенню або використати як стартовий пункт для побудови професійної кар'єри.

Успішна реалізація такого курсу вимагає модернізації технічної бази, підготовки кваліфікованих кадрів і створення сучасних методичних матеріалів. Це сприятиме розвитку якісної освіти, що відповідає потребам сучасного ринку праці та глобальним трендам.

## РОЗДІЛ 2

# МЕТОДИЧНА РОЗРОБКА ФАКУЛЬТАТИВНОГО КУРСУ З РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР ДЛЯ ПРОФІЛЬНОГО НАВЧАННЯ ІНФОРМАТИКИ УЧНІВ ЛІЦЕЇВ

### **2.1. Аналіз можливостей вивчення розробки комп'ютерних ігор у навчальних програмах старшої профільної школи**

В умовах стрімкого розвитку цифрових технологій та високої популярності ігрової індустрії у світі, включення розробки комп'ютерних ігор у навчальні програми старшої профільної школи в Україні стає все більш актуальним. Такий підхід здатний розширити горизонти традиційної освіти, сприяти розвитку інноваційного мислення та підготувати учнів до викликів майбутнього.

Першим завданням при створенні курсу з розробки ігор є вибір підходу. Визначення змісту, цілей і завдань розробки ігор є важливим кроком, особливо в світлі обмежених матеріальних і часових ресурсів. Огляд публікацій на цю тему показує, що реалізація навчальних програм з розробки ігор досить різноманітна: від окремих курсів (Джонс [25], Пербері та ін. [26], Свідик та Келер [27]) та включення відповідних розділів до традиційної програми інформатики (Коулман та ін. [28]) до набору курсів (Кларк та ін. [29], Фачада та Кодісіс [30], Пербері та ін [31], Рокко та Йодер [32], Прохоров та ін. [33]).

Зміст окремих курсів від використання рушіїв, розроблених спеціально для навчальних цілей (Gamemaker [5], RPG Maker [4], Alice [34]), розробки власних ігрових рушіїв (Labyrinth [35, 36], CAGE [37]), технічного дизайну [34], Flash [38] до повного навчального курсу з розробки ігор, що охоплює всі аспекти гри [17, 25].

Аналіз можливостей вивчення цього напрямку в українських школах має враховувати освітній потенціал, технічну базу та методичну готовність педагогів. Технічна база є ключовим фактором для успішного впровадження курсу з розробки комп'ютерних ігор у навчальні програми старшої школи. Стан технічного забезпечення шкіл в Україні значно впливає на ефективність інтеграції інноваційних підходів до навчання, включаючи використання сучасних цифрових технологій. Аналізуючи поточний стан технічної бази українських шкіл, можна виділити як позитивні зрушення, так і виклики, які потребують вирішення.

Українські школи мають різний рівень технічного забезпечення. У містах більшість шкіл обладнані комп'ютерними класами, але в сільській місцевості ситуація є більш складною. У великих містах шкільні комп'ютерні класи часто мають доступ до сучасних комп'ютерів, проте їхня кількість може бути недостатньою для всіх учнів класу. Часто використовуються бюджетні моделі комп'ютерів, що не завжди мають достатню продуктивність для роботи з ігровими рушіями, такими як Unity чи Unreal Engine. У багатьох сільських школах комп'ютерні класи або відсутні, або обладнані застарілими пристроями, які мають обмежену функціональність. Нестача комп'ютерів у сільських школах створює значні перешкоди для рівного доступу до сучасних освітніх інструментів.

Доступність ліцензійного програмного забезпечення також є проблемою для багатьох українських шкіл. Більшість сучасних ігрових рушіїв, таких як Unity чи GameMaker, пропонують безкоштовні версії для освітніх цілей, проте школи потребують інвестицій у їх впровадження:

1. Ігрові рушії:

- Unity та Unreal Engine мають великі навчальні спільноти, проте для їх ефективного використання необхідне обладнання з високими технічними характеристиками.
- Простіші платформи, такі як GameMaker чи Scratch, можуть використовуватися на менш потужних комп'ютерах, що робить їх доступними для більшої кількості шкіл.

2. Операційні системи та офісні програми:

- У багатьох школах використовуються застарілі версії операційних систем, що не завжди сумісні з сучасними освітніми платформами.
- Часто відсутні спеціалізовані програми для графічного дизайну чи звукового редагування, які є важливими для курсу розробки ігор.

Доступ до стабільного та швидкого інтернету є критично важливим для навчання, пов'язаного з розробкою комп'ютерних ігор, оскільки більшість навчальних матеріалів, програмного забезпечення та ресурсів доступні онлайн.

Сільські школи часто страждають від повільного інтернет-з'єднання або взагалі не мають доступу до мережі. Це обмежує можливість учнів працювати з хмарними сервісами, такими як Google Workspace чи GitHub, які є стандартом у

галузі розробки ігор. Міські школи мають кращу інфраструктуру, але навіть у містах якість інтернет-з'єднання не завжди відповідає потребам для завантаження великих файлів чи роботи з онлайн-курсами.

Нерідко українські школи стикаються з браком кваліфікованих кадрів, здатних ефективно реалізовувати цей курс. Більшість вчителів інформатики не мають досвіду роботи з ігровими рушіями чи розробкою ігор. Потрібні спеціальні тренінги та підвищення кваліфікації, щоб забезпечити ефективну реалізацію курсу.

Відсутність адаптованих підручників та методичних рекомендацій для викладання розробки ігор створює додаткові труднощі для вчителів.

Для забезпечення ефективної інтеграції курсу з розробки комп'ютерних ігор у старшу профільну школу в Україні необхідно:

1. Оновлення технічної бази:

- Закупівля сучасних комп'ютерів та забезпечення швидкого інтернету в усіх школах, включаючи сільські.

- Використання легкодоступного програмного забезпечення, яке не потребує високих ресурсів.

2. Розробка навчальних матеріалів:

- Створення національних програм навчання, адаптованих до реальних технічних можливостей шкіл.

3. Підвищення кваліфікації вчителів:

- Проведення тренінгів для вчителів з роботи з ігровими рушіями та сучасними методиками навчання.

4. Рівний доступ до ресурсів:

- Забезпечення рівних можливостей для навчання незалежно від регіону.

Технічна база українських шкіл має великий потенціал, проте її вдосконалення є необхідним кроком для адаптації до викликів сучасної освіти. Інвестиції у технічну інфраструктуру та людські ресурси сприятимуть створенню інноваційної та конкурентоспроможної системи освіти в Україні.

Методична готовність педагогів є ключовим чинником успішного впровадження курсу з розробки комп'ютерних ігор у навчальні програми старшої школи. Це поняття охоплює як професійну компетенцію вчителів, так і наявність методичних матеріалів, які допомагають їм організувати процес навчання. У

сучасних умовах, коли технології постійно змінюються, викладачі потребують особливої підготовки, адаптації та підтримки.

Більшість вчителів інформатики в українських школах мають базові навички роботи з комп'ютерами, однак специфіка розробки комп'ютерних ігор вимагає більш поглиблених знань. До них належать:

- програмування: викладання розробки ігор передбачає використання ігрових рушіїв (Unity, Unreal Engine, GameMaker тощо), які вимагають знання мов програмування, таких як C#, Python або JavaScript.

- робота з графічними і мультимедійними програмами: для створення ігор потрібно знати основи роботи з програмами для створення 2D- і 3D-графіки, такими як Blender, Photoshop, GIMP.

- базові навички звукового монтажу, анімації та дизайну.

- робота з інтернет-ресурсами та навчальними платформами: інтеграція сучасних цифрових інструментів у навчальний процес вимагає знання таких платформ, як GitHub (для управління проектами), Google Workspace (для спільної роботи) та інших ресурсів для самостійного навчання учнів.

Методична готовність педагогів є критичним компонентом для успішного впровадження курсу розробки комп'ютерних ігор. Для досягнення цієї мети необхідно забезпечити викладачів як технічними, так і педагогічними знаннями, створити доступ до якісних методичних матеріалів та надати можливість проходження спеціалізованих тренінгів. Інвестиції у розвиток методичної компетенції педагогів сприятимуть не лише успішній реалізації курсу, але й підвищенню рівня цифрової грамотності учнів.

## **2.2. Опис методичного комплексу для підтримки факультативного курсу з розробки комп'ютерних ігор для профільного навчання**

Навчальний план курсу з розробки комп'ютерних ігор в Unity сплановано так, щоб учні поступово освоювали всі ключові аспекти створення ігор. Він побудований за модульним принципом, де кожен модуль охоплює теоретичні знання та практичні завдання, спрямовані на формування конкретних навичок.

### *План курсу*

#### 1. Початок роботи в Unity

2. Ігрові об'єкти та компоненти
3. Навігація та освітлення
4. Написання власних компонентів
5. Концепція префабів та сцен
6. Фізика, коллайдери та тригери
7. Камера та візуалізація
8. Система введення
9. Анімації
10. Unity UI

Навчання починається з ознайомлення зі середовищем розробки Unity, включаючи встановлення Unity Hub, налаштування середовища і огляд інтерфейсу. Учні вчаться розуміти, як організовані проекти в Unity, і знайомляться з поняттям сцен (Scenes), де відбувається основна робота з об'єктами, план першого уроку зазначений на рис. 2.1.

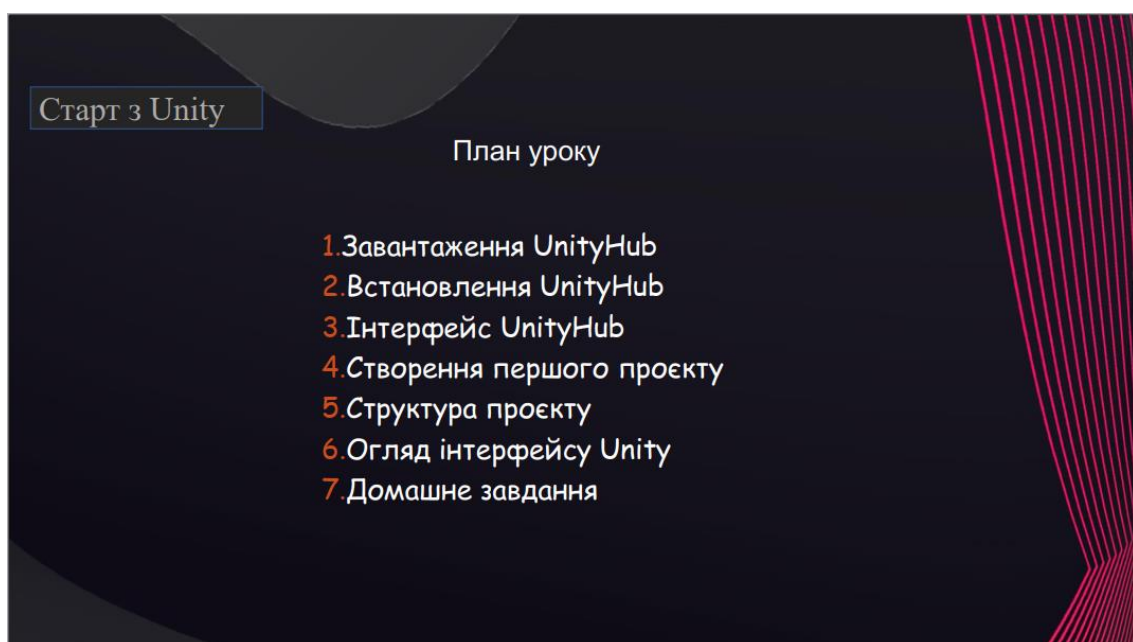


Рис. 2.1. План першого заняття

Після базового ознайомлення з Unity учні переходять до вивчення структури об'єктів у грі. На цьому етапі акцент робиться на ієрархії об'єктів, їхніх властивостях, а також взаємодії між компонентами. Учні дізнаються, як створювати прості ігрові об'єкти, налаштовувати їхню фізику, додавати освітлення, звуки та текстури. У рамках практичних завдань вони створюють невеликі сцени з інтерактивними

елементами, що дозволяє закріпити теоретичні знання через реалізацію простих ігрових механік.

Подальші модулі курсу зосереджені на розширених аспектах роботи з Unity, таких як використання фізичних компонентів (Rigidbody, Colliders) та системи освітлення. Учні вчаться створювати реалістичні сцени, працювати з тінями, текстурами, матеріалами та шейдерами. Це дає змогу їм зрозуміти, як створювати візуально привабливі ігри. Особлива увага приділяється розумінню масштабу у сцені та важливості роботи з одиницями виміру, що допомагає уникати типових помилок при проектуванні рівнів.

Навчальний план включає також модуль програмування, де учні знайомляться з основами мови C#. Вони вивчають, як писати скрипти для управління об'єктами, створювати інтерактивність, працювати з методами життєвого циклу (Awake, Start, Update). Практична частина передбачає написання скриптів для руху об'єктів, створення анімацій, взаємодії користувача з грою за допомогою нової системи вводу.

Окремий модуль присвячений роботі з Prefabs – шаблонами об'єктів, які можна використовувати багаторазово. Учні вчаться створювати Prefabs, зберігати їх у ресурсах, редагувати та застосовувати під час виконання гри. Ці навички дозволяють ефективно організовувати ігрові сцени та знижувати трудомісткість роботи.

На завершальних етапах курсу увага приділяється дизайну користувацького інтерфейсу. Учні вивчають компоненти Canvas, RectTransform, Image, Text та Button, а також їхні властивості. Вони створюють базовий інтерфейс гри, додають кнопки управління, лічильники балів і спливаючі меню. Наприкінці курсу передбачена розробка фінального проєкту, який інтегрує всі отримані знання та вміння: від створення сцени і об'єктів до написання логіки та дизайну UI.

Цей навчальний план спрямований на розвиток технічних і творчих здібностей учнів, формування практичних навичок роботи з ігровим рушієм Unity і створення готових ігрових проєктів.

Методичний комплекс має низку ключових особливостей, які роблять його максимально ефективним для освоєння основ розробки комп'ютерних ігор. Він створений з урахуванням потреб учнів, які лише починають знайомитися з Unity, і побудований так, щоб забезпечити системне, доступне та гнучке навчання.

Одна з основних особливостей — орієнтованість на самостійне навчання. Це



досягається завдяки чіткій структурі матеріалів, яка дозволяє учням працювати у власному темпі та обирати зручний для себе час і спосіб опрацювання теми. Кожен розділ починається з теоретичної частини у вигляді презентації, що охоплює ключові аспекти теми. Презентації містять лаконічні пояснення, схеми, приклади і короткі визначення, які роблять інформацію зрозумілою навіть без сторонньої допомоги. Це дозволяє учням самостійно формувати базові знання і впевнено переходити до виконання завдань.

Ще однією важливою особливістю є практична спрямованість комплексу. Домашні завдання розроблені таким чином, щоб учні могли отримати практичний результат після кожного завершеного етапу. Наприклад, початкові завдання можуть включати створення базових ігрових об'єктів (кубів, сфер, платформ), налаштування фізичних параметрів або простих взаємодій між об'єктами. Згодом складність завдань поступово зростає, включаючи програмування логіки за допомогою C#, створення системи анімації, інтеграцію користувацького інтерфейсу. Такий підхід дозволяє учням бачити практичний результат своєї роботи, що є потужною мотивацією продовжувати навчання.

Інтерактивний підхід методичного комплексу проявляється у тісному зв'язку між теорією та практикою. Кожна презентація не лише подає базові знання, але й демонструє їх застосування у реальних ситуаціях. Наприклад, у темі "Ієрархія об'єктів" презентація пояснює, як створювати вкладені об'єкти, а домашнє завдання пропонує учням реалізувати власну структуру ієрархії, працюючи з конкретними прикладами. Це дозволяє учням одразу застосувати отримані знання, що сприяє їх глибшому засвоєнню та розвитку практичних навичок.

Методичний комплекс також вирізняється гнучкістю у використанні. Завдяки поділу матеріалів на завершені модулі, він легко адаптується до різних форматів навчання. Учні можуть виконувати завдання індивідуально, у власному темпі, або працювати в парах чи групах під керівництвом викладача. Викладач, у свою чергу, може використовувати презентації як основу для своїх пояснень, а домашні завдання — для перевірки знань і закріплення матеріалу. Це дозволяє інтегрувати методичний комплекс у навчальний процес у різних формах, залежно від потреб і можливостей учнів.

Ще одна особливість — орієнтація на кінцевий результат. Комплекс

побудований так, щоб учні не лише вивчали окремі теми, але й поступово створювали свій власний ігровий проєкт. Це може бути простий платформер, головоломка або інтерактивний симулятор. Такий підхід дає можливість учням формувати портфоліо, що буде корисним у майбутньому для вступу до профільних навчальних закладів або участі в конкурсах.

Крім цього, методичний комплекс враховує сучасні тенденції в освіті. Він використовує доступні інструменти, такі як Unity, які є стандартом у галузі розробки ігор. Це дає учням змогу не лише отримувати теоретичні знання, але й знайомитися з реальними інструментами, які використовуються в індустрії.

Таким чином, методичний комплекс є не лише джерелом знань, але й потужним інструментом для розвитку практичних навичок, мотивації та творчого мислення. Завдяки поєднанню чіткої структури, практичності та гнучкості у використанні він забезпечує всебічну підтримку у процесі навчання і допомагає учням успішно опанувати основи розробки комп'ютерних ігор.

Презентації є основним джерелом теоретичного матеріалу у методичному комплексі. Вони виконують роль наочних посібників, які структуровано подають базові поняття, принципи та методи роботи в Unity. Кожна презентація створена так, щоб учні могли швидко ознайомитися з основними концепціями теми, зрозуміти їх суть і отримати чітке уявлення про застосування цих знань на практиці.

Структура презентацій розроблена таким чином, щоб полегшити засвоєння матеріалу. Вони починаються з короткого вступу, де пояснюється, чому тема є важливою для розробки ігор, та які практичні завдання можна виконати, володіючи цією інформацією. Основна частина поділяється на логічні блоки, що послідовно розкривають ключові аспекти теми. Наприклад, у презентації про ієрархію об'єктів спочатку пояснюється, що таке ієрархія, потім показується, як об'єкти впливають один на одного, і завершується практичними прикладами, які демонструють її застосування у проєктах.

Презентації насичені візуальними матеріалами: схемами, графіками, скріншотами з Unity та діаграмами. Це робить інформацію більш доступною та зрозумілою, навіть для тих, хто раніше не працював з розробкою ігор. Наприклад, візуалізація, що показує, як глобальна позиція об'єкта відрізняється від локальної, дозволяє учням швидше зрозуміти принцип роботи трансформацій у просторі.

Кожна презентація також містить приклади застосування теорії. Якщо тема стосується програмування, в презентації будуть наведені прості фрагменти коду з поясненнями, як вони працюють і які результати дають. У темах, пов'язаних із графікою або фізикою, додаються реальні приклади з Unity, які показують, як виглядають налаштування матеріалів, освітлення або взаємодія об'єктів під час гри. Це дає змогу учням одразу пов'язувати теоретичні знання з їхнім практичним застосуванням.

Презентації не лише інформативні, але й адаптивні. Вони створені з урахуванням різного рівня підготовки учнів. Наприклад, складні теми, такі як робота з анімаціями чи програмування взаємодії об'єктів, пояснюються покроково, щоб кожен учень міг легко слідувати логіці викладеного матеріалу. Завдяки цьому учні, які мають різний рівень попередніх знань, можуть однаково успішно опановувати курс.

Презентації також орієнтовані на інтерактивність. Вони закінчуються блоком із практичними рекомендаціями або завданнями, які допомагають одразу перевірити розуміння матеріалу. Наприклад, після теоретичного блоку про фізику об'єктів учням пропонується створити сцену, де об'єкти падають, стикаються та відскакують, використовуючи налаштування фізичних компонентів, описаних у презентації.

Таким чином, презентації є не лише засобом подачі інформації, але й інструментом для організації навчального процесу. Вони допомагають учням глибше зрозуміти матеріал, пов'язують теорію з практикою та забезпечують основу для самостійної роботи з завданнями курсу. Завдяки своєму структурованому та візуально привабливому формату презентації є зручним і ефективним ресурсом для вивчення основ розробки ігор в Unity.

Домашні завдання в методичному комплексі є практичною частиною навчання і виконують кілька ключових функцій: закріплення теоретичних знань, розвиток практичних навичок і створення власних ігрових проєктів. Вони розроблені таким чином, щоб учні могли поступово рухатися від базових завдань до більш складних, при цьому крок за кроком формуючи розуміння принципів роботи в Unity і створюючи функціональні елементи ігор.

Завдання охоплюють усі аспекти, які розглядаються в презентаціях, і дають можливість одразу застосувати отримані знання на практиці. Наприклад, після

вивчення основ роботи з компонентами учні отримують завдання створити просту сцену з декількома об'єктами, налаштувати для них ієрархію та додати базові компоненти, такі як Transform, Rigidbody і Collider. Це дозволяє їм зрозуміти, як працюють об'єкти та їх взаємодія у сцені.

Поступово завдання стають складнішими, охоплюючи такі теми, як налаштування фізики, створення анімацій, робота зі світлом та текстурами, написання простих скриптів на C#. Наприклад, учням може бути запропоновано створити платформу, яка рухається в просторі за допомогою коду, або розробити систему освітлення для сцени з реалістичними тінями. Завдання формулюються таким чином, щоб підштовхнути учнів до аналізу та самостійного пошуку рішень, одночасно даючи достатньо підказок для подолання складнощів.

Особливо важливим елементом домашніх завдань є їх практична результативність. Кожне завдання спрямоване на створення конкретного результату, який можна використати у фінальному проєкті або як окремий елемент портфоліо. Наприклад, завдання з анімації може включати створення персонажа, що рухається певним чином, а завдання з роботи з UI – створення інтерфейсу з кнопками, що реагують на дії користувача. Це забезпечує учням відчуття прогресу та мотивацію до подальшого навчання.

Домашні завдання також містять творчий компонент. Учні часто пропонується не лише виконати конкретну інструкцію, але й адаптувати її до власного бачення чи додати індивідуальні елементи. Наприклад, при створенні ігрової сцени вони можуть обирати власний стиль оформлення, текстури чи навіть сюжет. Це розвиває креативність і дозволяє учням виразити себе через ігровий дизайн.

Важливою особливістю завдань є їх гнучкість. Вони створені з урахуванням різного рівня підготовки учнів, тому містять як обов'язкові базові етапи, так і додаткові завдання для тих, хто хоче глибше зануритися в тему. Наприклад, учням може бути запропоновано додати до створеної сцени більш складні взаємодії між об'єктами, використовуючи написані вручну скрипти або готові компоненти Unity.

Домашні завдання також включають чіткі інструкції та рекомендації, що допомагають учням зрозуміти, з чого почати та які кроки необхідно виконати для досягнення результату. У разі складнощів учні можуть звертатися до презентацій, які

надають теоретичну основу, або додаткових ресурсів, посилання на які включені у завдання.

Таким чином, домашні завдання є невід'ємною частиною навчання, що дозволяє учням не лише закріпити отримані знання, але й отримати практичний досвід роботи в Unity. Вони формують уявлення про процес створення ігор, допомагають розвивати креативність і технічні навички, а також готують до реалізації власних ігрових проєктів.

Результати навчання після проходження курсу з розробки комп'ютерних ігор у Unity відображаються у формуванні ключових теоретичних знань і практичних навичок, необхідних для створення базових ігрових проєктів. Учні отримують розуміння основних принципів роботи з ігровим рушієм Unity, зокрема, організації проєкту, роботи з об'єктами, компонентами та сценами. Це дає змогу їм самостійно створювати базову структуру гри.

Після завершення курсу учні вмітимуть працювати з основними елементами Unity, такими як Game Objects, компоненти Transform, Rigidbody, Collider та інші. Вони розумітимуть, як створювати ієрархію об'єктів, налаштовувати їхні властивості та забезпечувати взаємодію між ними. Це закладає базу для подальшого вивчення складніших аспектів розробки, таких як оптимізація ігрових процесів або створення інтерактивних сцен.

Окремо слід зазначити формування практичних навичок програмування на C#. Учні вивчають основи написання скриптів для управління ігровою логікою, руху об'єктів, обробки вводу користувача та інших важливих функцій. Це дозволяє створювати не лише статичні сцени, а й інтерактивні елементи, що забезпечують динаміку гри. Наприклад, після проходження курсу учень може написати скрипт для переміщення платформи чи стрибків персонажа, налаштувати камеру для слідування за об'єктом або створити просту фізичну симуляцію.

Курс сприяє розвитку навичок роботи з графікою, текстурами та освітленням. Учні дізнаються, як створювати реалістичні сцени, використовуючи матеріали, тіні, джерела світла та текстури. Вони зможуть налаштувати сцену так, щоб вона виглядала природно, і при цьому розумітимуть принципи оптимізації графіки для забезпечення плавної роботи гри.

Одним із ключових результатів навчання є вміння працювати з

користувацьким інтерфейсом. Учні вчаться створювати інтерактивні елементи, такі як кнопки, текстові поля, лічильники балів або меню, що дозволяє додати до гри функціональний UI. Вони розуміють, як розташовувати елементи на екрані, налаштовувати їхній вигляд і взаємодію з користувачем.

На завершення курсу учні отримують навички роботи з анімаціями та інтеграції їх у свої проєкти. Вони дізнаються, як створювати ключові кадри, налаштовувати переходи між анімаціями та використовувати компонент Animator для керування анімацією персонажа чи інших об'єктів.

Важливою складовою результатів навчання є створення власного проєкту. Учні працюють над завершеним ігровим проєктом, який включає всі аспекти, вивчені під час курсу: від налаштування сцени та інтеграції графіки до написання скриптів і розробки ігрового інтерфейсу. Цей проєкт стає демонстрацією їхніх знань і практичних умінь, а також основою для подальшого портфоліо.

У результаті учні отримують не лише технічні знання, але й навички проєктного мислення, що дозволяє їм ефективно організовувати процес створення ігор. Вони стають здатними самостійно створювати невеликі ігрові проєкти, розуміють, як масштабувати свої знання для складніших завдань, і мають мотивацію до подальшого розвитку у сфері ігрової розробки.

### **2.3. Рекомендації щодо використання навчально методичного комплексу в навчальному процесі**

Навчально-методичний комплекс з розробки комп'ютерних ігор у Unity створений для забезпечення якісного навчального процесу, спрямованого на формування у учнів базових знань та практичних навичок у сфері ігрової розробки. Завдяки чіткій структурі, інтеграції теоретичних матеріалів та практичних завдань, комплекс є універсальним інструментом, що підходить як для групового навчання під керівництвом викладача, так і для самостійної роботи.

Цей комплекс включає презентації для вивчення теоретичного матеріалу та домашні завдання, які дозволяють учням одразу застосовувати отримані знання на практиці. Його можна адаптувати під різні формати навчання, забезпечуючи поступове засвоєння основних тем, таких як робота з ігровими об'єктами, сценою, фізикою, анімацією, програмуванням та дизайном інтерфейсу.

Рекомендації щодо використання комплексу допоможуть ефективно інтегрувати його в навчальний процес, враховуючи особливості організації занять, потреби учнів та навчальні цілі. Дотримання цих рекомендацій дозволить максимально використати потенціал матеріалів, сприяючи підвищенню рівня знань і практичних навичок учнів.

Розподіл матеріалу на модулі є ключовим етапом організації навчального процесу, оскільки дозволяє структуровано подати великий обсяг інформації, забезпечуючи логічну послідовність її вивчення. Кожен модуль представляє собою окрему тему або групу взаємопов'язаних тем, які охоплюють конкретний аспект розробки ігор в Unity. Такий підхід дозволяє учням поступово засвоювати матеріал, рухаючись від базових понять до складніших практичних завдань.

На початку курсу рекомендується зосередитися на ознайомленні з Unity, його інтерфейсом, структурою проєкту та базовими поняттями, такими як сцени та об'єкти. Перший модуль має забезпечити учнів розумінням основних принципів роботи в середовищі Unity, що стане фундаментом для подальшого навчання. Презентації цього модуля можуть включати огляд компонентів Unity, пояснення ієрархії об'єктів, а також демонстрацію інтерфейсу та базових інструментів.

Наступні модулі можуть бути присвячені окремим аспектам роботи з Unity, наприклад, фізиці, програмуванню скриптів, освітленню чи анімації. Кожен модуль поділяється на теоретичну частину, що пояснює ключові концепції, та практичну, яка включає домашні завдання для закріплення матеріалу. Наприклад, модуль з фізики може початися з пояснення фізичних компонентів, таких як Rigidbody і Colliders, а закінчитися практичним завданням зі створення сцени, де об'єкти взаємодіють відповідно до законів фізики.

Ключовим елементом такого розподілу є поступовість і взаємозв'язок матеріалу. Кожен наступний модуль повинен спиратися на знання, отримані в попередньому, забезпечуючи природний перехід до складніших завдань. Наприклад, після вивчення основ програмування учні можуть перейти до створення інтерактивності в іграх, а потім до роботи з користувацьким інтерфейсом.

Окремі модулі варто присвячувати комплексним темам, таким як створення фінального проєкту. Наприкінці курсу учні можуть об'єднати всі отримані знання для розробки власної гри, інтегруючи сцену, об'єкти, скрипти, анімації та UI в єдиний

проект. Такий підхід дозволяє не лише повторити вивчені теми, але й наочно продемонструвати результати навчання.

Розподіл матеріалу на модулі забезпечує логічну структуру навчання, робить матеріал більш доступним для засвоєння та дозволяє ефективно поєднувати теорію з практикою. Це створює сприятливі умови для учнів, допомагаючи їм послідовно освоювати нові знання та навички.

Використання презентацій у навчальному процесі є важливим елементом навчально-методичного комплексу, оскільки вони забезпечують чітку та структуровану подачу теоретичного матеріалу. Презентації розроблені так, щоб пояснювати складні поняття зрозумілою мовою, використовуючи візуалізацію, приклади та ключові терміни. Це робить їх зручним інструментом як для викладача, так і для учнів, які можуть використовувати їх як довідковий матеріал.

На початку кожного заняття презентації слугують основою для ознайомлення учнів із новою темою. Вони допомагають викладачеві структурувати подачу матеріалу та зосередити увагу на ключових аспектах. Наприклад, при вивченні теми "Ієрархія об'єктів" презентація може містити визначення ієрархії, приклади її застосування та схеми, що пояснюють взаємодію об'єктів у різних контекстах. Завдяки цьому учні отримують чітке розуміння основних понять перед тим, як перейти до практичної частини.

Важливою особливістю презентацій є використання візуального матеріалу. Схеми, діаграми, скріншоти з Unity та покрокові ілюстрації значно полегшують сприйняття інформації. Наприклад, замість сухого пояснення, що таке Collider, презентація може показати його у дії через анімовану схему або зображення з поясненнями. Це сприяє глибшому розумінню матеріалу навіть для учнів, які раніше не працювали з подібними концепціями.

Презентації також відіграють роль посібників, які можна використовувати для самостійного вивчення. Учні, які не встигли зрозуміти матеріал під час заняття, можуть звернутися до презентацій, щоб повторити ключові моменти. Оскільки презентації побудовані у зрозумілій і логічній формі, вони дозволяють швидко знайти необхідну інформацію та пов'язати її з практичними завданнями.

Після ознайомлення з теоретичною частиною презентації допомагають перейти до практичних завдань. Наприклад, після пояснення теми "Фізика в Unity" у



презентації може бути запропонований приклад налаштування RigidBody, який учні зможуть одразу реалізувати в домашньому завданні. Такий зв'язок між теорією та практикою допомагає краще засвоїти матеріал і застосувати його в реальних проєктах.

Використання презентацій сприяє підвищенню ефективності навчання, адже вони структурують процес отримання знань, забезпечують наочність і дозволяють учням самостійно опанувати матеріал. Вони є універсальним інструментом, який можна адаптувати під різні потреби учнів і формати навчання, забезпечуючи якісну підтримку вивчення кожної теми курсу.

Домашні завдання є основою практичного закріплення знань, отриманих у ході вивчення теоретичного матеріалу. Вони розроблені таким чином, щоб учні могли одразу застосувати на практиці те, що вивчили під час заняття, поступово рухаючись від простих завдань до створення власного ігрового проєкту. Завдяки цьому учні не лише розвивають технічні навички, але й формують уявлення про процес розробки ігор.

Кожне завдання логічно пов'язане з темою, розглянутою у презентаціях. Наприклад, якщо урок стосується роботи з ієрархією об'єктів, домашнє завдання може включати створення сцени, де кілька об'єктів впорядковані у вигляді "батько-дитина", з подальшим налаштуванням локальних і глобальних трансформацій. Завдяки такому підходу учні краще розуміють, як теоретичні поняття застосовуються у реальній розробці ігор.

Домашні завдання розроблені таким чином, щоб відповідати рівню підготовки учнів. На початкових етапах завдання спрямовані на засвоєння базових концепцій, таких як створення простих об'єктів, налаштування їхніх властивостей або використання базових компонентів Unity (Transform, RigidBody, Collider). У міру засвоєння матеріалу складність завдань зростає. Наприклад, учні переходять до програмування простих взаємодій між об'єктами, використання анімацій та створення користувацького інтерфейсу.

Важливою частиною завдань є їхня практична результативність. Кожне завдання побудоване так, щоб у результаті учень створював функціональний елемент гри, який може бути використаний у фінальному проєкті. Наприклад, завдання з фізики може полягати у створенні сцени, де об'єкти падають і стикаються відповідно

до налаштувань фізичних компонентів. Такі завдання дозволяють учням бачити конкретний результат своєї роботи, що мотивує їх до подальшого навчання.

Крім стандартних завдань, учням часто пропонуються творчі завдання, які стимулюють їхню креативність. Наприклад, при створенні сцени з освітленням учням може бути запропоновано налаштувати джерела світла так, щоб вони відповідали заданій атмосфері (ніч, день, загадковий підземний світ). Це дозволяє учням застосовувати свої знання у нестандартних ситуаціях, розвивати фантазію і вчитися знаходити власні рішення.

Домашні завдання також включають додаткові пояснення та рекомендації, які допомагають учням зрозуміти, з чого почати роботу та які інструменти використовувати. У разі складнощів учні можуть звертатися до теоретичних матеріалів у презентаціях або до інших ресурсів, посилення на які додаються до завдань. Це дозволяє розвивати навички самостійного вирішення проблем і знаходження необхідної інформації.

Завдяки домашнім завданням учні поступово формують власний ігровий проєкт, який інтегрує всі теми курсу: від створення сцени та об'єктів до реалізації анімацій, програмування взаємодій та налаштування інтерфейсу. Це допомагає не лише закріпити отримані знання, але й створити щось унікальне, що стане результатом їхньої праці та предметом гордості. У процесі виконання завдань учні не лише опановують технічні аспекти, а й отримують досвід організації роботи, планування та вирішення завдань, що є важливими навичками для майбутньої професійної діяльності.

Самостійна робота учнів є важливим елементом навчального процесу в рамках курсу з розробки комп'ютерних ігор. Вона спрямована на закріплення знань, отриманих під час занять, розвиток навичок самостійного вирішення завдань та формування вміння організувати власний навчальний процес. Завдяки методичному комплексу учні отримують необхідні матеріали, які дозволяють ефективно працювати з теоретичними та практичними завданнями.

Основою самостійної роботи є презентації та домашні завдання, які дають змогу учням крок за кроком опановувати нові знання та застосовувати їх у практичних проєктах. Презентації виконують роль довідкового матеріалу, до якого учні можуть звертатися під час виконання завдань. Завдяки зрозумілій структурі

презентацій учні можуть швидко знаходити необхідну інформацію, повторювати ключові поняття та пов'язувати їх із практичними завданнями. Наприклад, при виконанні завдання з налаштування фізики у сцені учень може повернутися до презентації, щоб уточнити значення параметрів Rigidbody або Collider.

Самостійна робота стимулює учнів до аналізу та пошуку рішень. Завдання побудовані так, щоб учень мав можливість експериментувати з інструментами Unity, пробувати різні варіанти виконання завдань і бачити результати своїх дій. Наприклад, налаштування освітлення чи створення текстур дає широкий простір для творчості, що допомагає учням розвивати не лише технічні, але й художні навички.

Важливим аспектом самостійної роботи є гнучкість у плануванні часу. Учні можуть самостійно визначати, коли і як довго працювати над завданнями, що сприяє формуванню відповідальності та вмінню розподіляти власний час. Це особливо важливо для старшокласників, які вже мають певний досвід навчання та потребують більше автономії в опануванні нових знань.

Для підтримки самостійної роботи у методичному комплексі передбачено додаткові ресурси. У завданнях можуть бути посилання на відеоуроки, документацію Unity чи сторонні матеріали, які допомагають краще зрозуміти складні аспекти курсу. Це дозволяє учням розвивати навички самоосвіти, які є важливими у сучасному освітньому середовищі.

Самостійна робота також сприяє розвитку критичного мислення. Учні аналізують свої помилки, знаходять способи їх виправлення та вдосконалюють свої проєкти. Наприклад, у процесі тестування створеної сцени вони можуть виявити, що фізика об'єктів працює некоректно, і повернутися до налаштувань компонентів, щоб виправити проблему. Такий підхід формує навички вирішення реальних технічних завдань, що є важливим для подальшого професійного розвитку.

Отже, самостійна робота учнів не лише закріплює знання, але й сприяє формуванню навичок самонавчання, креативного підходу до виконання завдань і вміння організувати свій час. Це робить її важливим компонентом курсу, який забезпечує глибоке засвоєння матеріалу і сприяє успішному виконанню підсумкового проєкту.

Інтеграція завдань у єдиний проєкт є важливим аспектом курсу, який дозволяє учням зрозуміти, як різні елементи розробки ігор взаємодіють між собою, і навчитися

об'єднувати їх у цілісний продукт. Цей підхід забезпечує не лише глибоке засвоєння матеріалу, але й формування системного мислення, яке необхідне для роботи над реальними ігровими проєктами.

Протягом курсу кожне завдання стає частиною майбутньої гри, яку учні створюють поступово. Наприклад, на початковому етапі учні працюють над створенням базової сцени, додаючи прості об'єкти та налаштовуючи їхню ієрархію. Далі вони вчаться додавати фізичні компоненти, такі як Rigidbody або Colliders, що дозволяє об'єктам взаємодіяти за законами фізики. Кожен із цих елементів стає фундаментом для більш складних завдань, таких як створення інтерактивності чи додавання анімацій.

Поступове додавання нових елементів допомагає учням побачити, як різні аспекти розробки ігор поєднуються між собою. Наприклад, після написання скрипту для руху об'єкта учні можуть налаштувати камеру, яка слідкуватиме за цим об'єктом у просторі. Додавання освітлення та текстур зробить сцену більш реалістичною, а створення інтерфейсу дозволить гравцю взаємодіяти з грою. Такий підхід не лише сприяє кращому розумінню окремих тем, але й демонструє учням важливість комплексного підходу до розробки.

Інтеграція завдань у єдиний проєкт також мотивує учнів, оскільки вони бачать, як їхня робота поступово перетворюється на завершений продукт. Замість виконання ізольованих завдань, які можуть здаватися безцільними, учні створюють щось конкретне, що викликає відчуття досягнення. Наприклад, завершена гра з усіма елементами, розробленими протягом курсу, стає не лише навчальним результатом, але й предметом гордості, який можна показати друзям, родині чи використовувати у портфоліо.

Такий підхід також дозволяє розвивати у учнів навички проєктного менеджменту. У ході виконання завдань вони вчаться планувати свою роботу, розподіляти час між різними етапами проєкту та знаходити способи вирішення складних технічних задач. Наприклад, якщо під час інтеграції різних компонентів виникає конфлікт (наприклад, невідповідність налаштувань фізики та анімації), учні аналізують проблему та шукають оптимальне рішення.

На завершальному етапі курсу учні отримують завдання завершити свій проєкт, об'єднавши всі створені елементи в одну гру. Це включає не лише технічну

інтеграцію, але й поліпшення дизайну, оптимізацію роботи сцени, тестування гри на помилки та створення фінального інтерфейсу. Завдяки цьому вони отримують цілісний досвід розробки гри, який можна порівняти з реальними робочими процесами у професійній сфері.

Інтеграція завдань у єдиний проєкт дозволяє учням глибше зрозуміти принципи розробки ігор, побачити зв'язок між різними аспектами курсу та розвинути вміння працювати над масштабними задачами. Це забезпечує їх не лише технічними знаннями, але й важливими навичками організації роботи, креативного мислення та вирішення проблем. У результаті учні виходять за межі навчання окремих інструментів і отримують повноцінний досвід створення завершеного продукту.

Індивідуальний підхід у рамках навчального процесу з використанням методичного комплексу є важливим аспектом, що дозволяє максимально ефективно адаптувати матеріали до потреб і рівня підготовки кожного учня. Завдяки гнучкості комплексу, який містить чітко структуровані презентації та різнорівневі завдання, викладач має змогу створити комфортні умови для навчання як для початківців, так і для більш досвідчених учнів.

Для учнів, які тільки починають знайомство з розробкою ігор, особливо важливо приділяти увагу поясненню базових концепцій. Наприклад, замість складних завдань їм можна запропонувати простіші вправи, такі як створення базових об'єктів і налаштування їхніх властивостей у Unity. Під час виконання завдань учням можна надавати більше підказок і прикладів, щоб вони могли поступово вивчати матеріал і набиратися впевненості. Також для початківців доцільно проводити більше часу на поясненні інтерфейсу Unity, принципів роботи з ієрархією об'єктів і налаштування базових компонентів.

Для учнів із вищим рівнем підготовки завдання можна ускладнювати, додаючи творчі або технічно насичені елементи. Наприклад, якщо базове завдання передбачає налаштування руху об'єкта за допомогою скрипту, більш досвідченим учням можна запропонувати додати складнішу логіку, як-от обробку кількох станів об'єкта чи реакцію на взаємодію з іншими об'єктами. Такий підхід стимулює інтерес учнів, які вже мають базові знання, і дозволяє їм розвивати свої навички.

Індивідуальний підхід також проявляється у виборі темпу навчання. Деякі учні засвоюють матеріал швидше, ніж інші, тому вони можуть виконувати завдання

наперед або отримувати додаткові творчі завдання. Інші, навпаки, можуть потребувати більше часу для виконання завдань або повторення теоретичного матеріалу. Викладачеві важливо підтримувати та мотивувати таких учнів, пояснюючи складні моменти більш детально або надаючи додаткові приклади.

Завдяки гнучкій структурі курсу викладач може адаптувати домашні завдання залежно від зацікавленості учнів. Наприклад, якщо хтось виявляє особливий інтерес до дизайну, йому можна запропонувати завдання, що зосереджуються на налаштуванні текстур, матеріалів і освітлення сцени. Для тих, хто цікавиться програмуванням, можна створити додаткові завдання, які передбачають написання скриптів для складних взаємодій у грі. Це дозволяє розвивати індивідуальні сильні сторони кожного учня.

Індивідуальний підхід також сприяє формуванню у учнів почуття відповідальності за власний навчальний процес. Вони отримують можливість самостійно обирати складність завдань і напрямок, у якому хочуть розвиватися. Викладач у такому випадку виступає не лише як джерело знань, але й як ментор, який допомагає учням знаходити найкращі шляхи для досягнення їхніх цілей.

У результаті індивідуальний підхід створює умови для комфортного та продуктивного навчання, де кожен учень отримує необхідну підтримку і виклики, які відповідають його рівню. Це допомагає не лише ефективно засвоювати матеріал, але й розвивати вміння працювати самостійно, вирішувати проблеми та досягати поставлених цілей.

Оцінювання знань і вмінь учнів є важливим компонентом навчального процесу, який дозволяє визначити рівень засвоєння матеріалу, виявити сильні та слабкі сторони кожного учня і допомогти їм покращити свої результати. В рамках курсу з розробки комп'ютерних ігор у Unity оцінювання має бути спрямоване на перевірку як теоретичних знань, так і практичних навичок, необхідних для створення ігрових проєктів.

Одним із ключових елементів оцінювання є перевірка виконання домашніх завдань. Завдання розроблені таким чином, щоб поступово підвищувати рівень складності, тому їх виконання демонструє, наскільки учень освоїв відповідний матеріал. Наприклад, у початкових завданнях викладач може оцінювати точність виконання простих інструкцій, таких як створення об'єктів чи налаштування

компонентів, тоді як у пізніших завданнях акцент робиться на складніших аспектах, таких як написання скриптів чи інтеграція кількох елементів у єдиний проєкт.

Практичне оцінювання також включає аналіз проміжних результатів. Наприклад, під час виконання проєкту учні можуть періодично демонструвати свої напрацювання, щоб отримати зворотний зв'язок від викладача. Це дозволяє не лише оцінити їхній прогрес, але й допомогти виявити та виправити можливі помилки на ранніх етапах. Викладач може звертати увагу на те, як учні реалізують ідеї, наскільки правильно використовують компоненти Unity і наскільки ефективно працюють із кодом.

Теоретичне оцінювання також є важливим і може проводитися у формі тестів або усних опитувань. Тести допомагають перевірити знання основних термінів, принципів роботи Unity, розуміння ролі компонентів і процесів. Наприклад, учням можна запропонувати пояснити, як працює RigidBody або для чого використовується MonoBehaviour. Таке оцінювання дозволяє викладачеві переконатися, що учні розуміють основи, які вони застосовують у своїх практичних завданнях.

Особливе місце в оцінюванні займає фінальний проєкт, який об'єднує всі знання і навички, здобуті під час курсу. Завершений проєкт демонструє, наскільки учень зміг інтегрувати теоретичні знання з практичними вміннями, а також його здатність організувати роботу над великим завданням. Під час оцінювання фінального проєкту враховуються різні аспекти, зокрема технічна реалізація (коректна робота скриптів, налаштування компонентів), дизайн (оформлення сцени, вибір текстур і освітлення) та функціональність (інтерактивність, зручність користування).

Оцінювання також має включати елемент самоаналізу. Учням можна запропонувати пояснити, як вони вирішували ті чи інші завдання, які труднощі виникали і як вони їх подолали. Це допомагає не лише оцінити їхні результати, але й зрозуміти, як учні підходять до розв'язання проблем, що є важливим для їхнього подальшого розвитку.

Важливо, щоб оцінювання було не лише формальним, а й мотивуючим. Викладач повинен не лише вказувати на помилки, але й підкреслювати досягнення учнів, заохочувати їх до подальшої роботи. Це сприяє формуванню впевненості у своїх силах і бажанню вдосконалювати свої навички. Таким чином, оцінювання стає

інструментом не лише перевірки знань, але й підтримки учнів у їхньому розвитку.

Гнучкість використання навчально-методичного комплексу є однією з його ключових переваг, що дозволяє адаптувати матеріали до різних форматів навчання, рівня підготовки учнів та умов проведення занять. Така універсальність забезпечується чіткою структурою матеріалів, які складаються з презентацій і домашніх завдань, розподілених за модулями. Це створює можливості для їх застосування як у групових заняттях під керівництвом викладача, так і для індивідуального самостійного навчання.

Методичний комплекс можна використовувати в традиційному форматі, коли викладач проводить заняття, пояснюючи теоретичний матеріал із презентацій і супроводжуючи його демонстрацією у Unity. У цьому випадку викладач має змогу детально розглядати складні теми, показувати наочні приклади та відповідати на запитання учнів у реальному часі. Після цього учням можна запропонувати виконувати домашні завдання, які дозволяють закріпити отримані знання через практичну діяльність.

Комплекс також підходить для дистанційного навчання. Завдяки чіткій структурі матеріалів учні можуть працювати з презентаціями та виконувати завдання у зручний для себе час. Викладач може контролювати процес через онлайн-платформи, організовувати консультації, перевіряти виконані завдання та давати зворотний зв'язок. Такий підхід дозволяє організувати ефективний навчальний процес навіть за відсутності можливості проводити заняття у класі.

Для індивідуального навчання комплекс є ідеальним інструментом, оскільки його матеріали побудовані так, щоб учень міг самостійно розібратися в теоретичній частині та застосувати знання на практиці. Презентації виконують роль довідкового матеріалу, до якого учень може звертатися під час виконання завдань. Завдання подані в доступному форматі, з чіткими інструкціями та покроковими рекомендаціями, що дозволяє учням самостійно виконувати їх без постійної підтримки викладача.

Гнучкість використання також проявляється у можливості адаптувати матеріали до різного рівня підготовки учнів. Для новачків можна спростити завдання, зосередившись на базових аспектах, таких як створення об'єктів, робота з компонентами та налаштування сцени. Більш досвідченим учням можна



запропонувати складніші завдання, що включають написання скриптів, створення інтерактивності або оптимізацію ігрових сцен. Викладач може комбінувати завдання, додаючи творчі елементи або розширюючи практичну частину за бажанням учнів.

Комплекс також дозволяє організовувати роботу як у групах, так і індивідуально. Наприклад, у групових проєктах учні можуть розподіляти ролі між собою, працюючи над різними частинами гри (графіка, програмування, анімація). Це сприяє розвитку командних навичок і розуміння того, як організовується спільна робота у розробці ігор.

Завдяки такій гнучкості навчально-методичний комплекс може бути використаний у різних умовах, відповідаючи потребам учнів і викладачів. Він дозволяє організувати як стандартний курс із чітким розкладом занять, так і модульне навчання, де учні опрацьовують теми у власному темпі. Це робить його універсальним інструментом, придатним для сучасного освітнього середовища.

### **Висновки до другого розділу**

Розробка факультативного курсу з основ створення комп'ютерних ігор є значним кроком у модернізації навчального процесу старшої школи, спрямованого на розвиток цифрових і творчих навичок учнів. Структурований методичний комплекс, побудований за модульним принципом, дозволяє послідовно опанувати ключові аспекти роботи з сучасним ігровим рушієм Unity, зокрема програмування, фізичну взаємодію об'єктів, створення графіки та анімації, а також розробку користувацького інтерфейсу.

Однією з головних переваг цього комплексу є його орієнтованість на інтеграцію теоретичних знань із практичними завданнями. Такий підхід дозволяє учням закріплювати нові знання через створення реальних ігрових сцен і проєктів. Інтерактивні презентації, чітко сформульовані завдання та гнучкість у використанні матеріалів забезпечують зручність як для самостійного навчання, так і для роботи під керівництвом викладача.

Особливе значення має поступове виконання домашніх завдань, які спрямовані на створення цілісного ігрового проєкту. Це не лише допомагає учням засвоювати технічні аспекти розробки, але й розвиває їхні навички планування, командної роботи та креативного мислення. Підсумковий проєкт стає демонстрацією

отриманих знань і цінним елементом для подальшого портфолію.

Запропоновані рекомендації щодо використання комплексу гарантують його універсальність, адаптацію до рівня підготовки учнів та можливість впровадження у різних навчальних форматах. Такий курс здатен не лише сприяти підвищенню рівня цифрової грамотності, а й допомогти школярам визначити свої інтереси та потенціал для майбутньої професійної діяльності.

## ВИСНОВКИ

Проведене дослідження дозволило зробити висновки про перспективність впровадження факультативного курсу з розробки комп'ютерних ігор для учнів ліцеїв. У сучасному цифровому світі, де інформаційні технології швидко змінюють всі аспекти нашого життя, така ініціатива є актуальною і своєчасною. Геймдев, як одна з найдинамічніших сфер ІТ-індустрії, надає молоді унікальні можливості для реалізації свого творчого й технічного потенціалу.

Сучасний стан шкільних навчальних програм з інформатики не дозволяє інтегрувати в них розробку комп'ютерних ігор, але повністю можливо реалізувати її у вигляді розробленого в роботі факультативу. Це сприятиме розвитку критичного мислення, логіки, креативності, а також здатності до розв'язання складних завдань. Учні отримають змогу опанувати сучасні програмні інструменти, зокрема Unity та мову програмування C#, які є стандартами в ігровій індустрії. Курс підтримує міждисциплінарний підхід до навчання, інтегруючи знання з математики, фізики, мистецтва, літератури та інших дисциплін, що сприяє всебічному розвитку особистості учнів ліцеїв.

Методологічна основа курсу базується на модульному підході, який забезпечує поступове освоєння матеріалу: від базових понять до складніших практичних завдань. Учні починають із вивчення інтерфейсу Unity, основ роботи з об'єктами та компонентами, поступово переходячи до програмування, анімації, роботи зі світлом, текстурами та розробки користувацького інтерфейсу. Наприкінці курсу вони створюють власний ігровий проєкт, який стає демонстрацією їхніх знань і навичок.

Особливістю курсу є його інтерактивний і практичний характер, що дає можливість учням одразу застосовувати знання на практиці. Домашні завдання структуровані так, щоб учні могли поступово створювати функціональні елементи гри, які об'єднуються у фінальному проєкті. Такий підхід дозволяє не лише закріпити теоретичні знання, але й побачити результати своєї роботи, що стимулює подальший інтерес до навчання.

У методичних рекомендаціях до курсу враховано сучасні тенденції в освіті, такі як використання інтерактивних презентацій, індивідуалізований підхід до навчання та можливість адаптації курсу до різного рівня підготовки учнів. Це

забезпечує універсальність комплексу, який можна використовувати як у традиційному, так і в дистанційному форматах навчання.

Практична значущість дослідження полягає в тому, що запропонований курс можна інтегрувати в навчальний процес профільних класів ліцеїв, забезпечуючи учням фундаментальні знання з програмування та розробки ігор. Отримані навички не лише допоможуть учням у подальшому навчанні, але й стануть конкурентною перевагою на ринку праці.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. T. A. Vakaliuk, V. V. Kontsedailo, D. S. Antoniuk, O. V. Korotun, I. S. Mintii, A. V. Pikilnyak, Using game simulator Software Inc in the Software Engineering education, in: A. E. Kiv, M. P. Shyshkina (Eds.), Proceedings of the 2nd International Workshop on Augmented Reality in Education, Kryvyi Rih, Ukraine, March 22, 2019, volume 2547 of CEUR Workshop Proceedings, CEUR-WS.org, 2019, pp. 66–80. URL: <https://ceur-ws.org/Vol-2547/paper05.pdf>.
2. O. M. Haranin, N. V. Moiseienko, Adaptive artificial intelligence in RPG-game on the Unity game engine, CEUR Workshop Proceedings 2292 (2018) 143–150.
3. O. O. Katsko, N. V. Moiseienko, Development computer games on the Unity game engine for research of elements of the cognitive thinking in the playing process, CEUR Workshop Proceedings 2292 (2018) 151–155.
4. T. Barnes, H. Richter, E. Powell, A. Chaffin, A. Godwin, Game2learn: Building cs1 learning games for retention, ACM SIGCSE Bulletin 39 (2007) 121–125. doi:10.1145/1269900.1268821.
5. K. Claypool, M. Claypool, Teaching software engineering through game design, ACM SIGCSE Bulletin 37 (2005) 123–127. doi:10.1145/1151954.1067482.
6. B. B. Morrison, J. A. Preston, Engagement, ACM SIGCSE Bulletin 41 (2009) 342–346. doi:10.1145/1539024.1508990.
7. T. E. Roden, R. LeGrand, Growing a computer science program with a focus on game development, in: Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE'13, ACM Press, 2013. doi:10.1145/2445196.2445362.
8. K. Sung, Computer games and traditional CS courses, Communications of the ACM 52 (2009) 74–78. doi:10.1145/1610252.1610273.
9. O. M. Haranin, O. O. Katsko, N. V. Moiseienko, Developer software tools in a course “Development of computer games”, New computer technology 15 (2017) 160–163.
10. T. A. Vakaliuk, V. Kontsedailo, D. Antoniuk, O. Korotun, S. Semerikov, I. S. Mintii, Using Game Dev Tycoon to Create Professional Soft Competencies for Future EngineersProgrammers, in: O. Sokolov, G. Zholtkevych, V. Yakovyna, Y. Tarasich, V. Kharchenko, V. Kobets, O. Burov, S. Semerikov, H. Kravtsov (Eds.), Proceedings of the 16th International Conference on ICT in Education, Research and Industrial

- Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, Kharkiv, Ukraine, October 06-10, 2020, volume 2732 of CEUR Workshop Proceedings, CEUR-WS.org, 2020, pp. 808–822. URL: <https://ceur-ws.org/Vol-2732/20200808.pdf>.
11. Q. Brown, F. Lee, S. Alexandre, Emphasizing soft skills and team development in an educational digital game design course, in: Proceedings of the 4th International Conference on Foundations of Digital Games, FDG '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 240–247. doi:10.1145/1536513.1536557.
  12. Y. Rankin, A. Gooch, B. Gooch, The impact of game design on students' interest in CS, in: Proceedings of the 3rd international conference on Game development in computer science education - GDCSE'08, ACM Press, 2008. doi:10.1145/1463673.1463680.
  13. Програма профільного вивчення інформатики 10 – 11 класи інформаційно-технологічного профілю. – URL: <https://mon.gov.ua/static-objects/mon/sites/1/zagalna%20serednya/programy-10-11-klas/prof-riven.pdf> (дата звернення: 11.11.2023).
  14. Руденко В. Д., Речич Н. В., Потієнко В. О. Інформатика (профільний рівень): підручник для 10 кл. закладів загальної середньої освіти. Харків: Ранок, 2018. 256 с.
  15. Руденко В. Д., Речич Н. В., Потієнко В. О. Інформатика (профільний рівень): підручник для 11 кл. закладів загальної середньої освіти. Харків : Ранок, 2019. 256 с.
  16. Морзе Н., Нанаєва Т., Пасічник О. Стан та перспективи навчання інформатики в закладах загальної середньої освіти України, *Інформаційні технології і засоби навчання*, 92(6), pp. 1–20. 2022. doi:10.33407/itlt.v92i6.5138.
  17. K. Becker, J. R. Parker, Serious Games + Computer Science = Serious CS, *Journal of Computing Sciences in Colleges* 23 (2007) 40–46. doi:10.5555/1292428.1292436.
  18. Сайт Карнегі-Мелонського університету URL: <https://www.cmu.edu/pre-college/academic-programs/game-academy.html> (дата звернення 19.04.2024).
  19. Сайт GameLab Масачусетського інституту технологій URL: <https://gamelab.mit.edu/> (дата звернення 20.04.2024).
  20. Сайт ІТ-школи Robocode URL: <https://robocode.ua/kursy-unity-ua#programs>

(дата звернення 18.05.2024).

21. Сайт ІТ-школи IT-generation URL: <https://it-generation.com.ua/navchalni-programi/unity/> (дата звернення 18.05.2024). (дата звернення 16.05.2024).
22. Сайт ІТ-школи GO-ITeens URL: <https://goiteens.com/course/game-dev/> (дата звернення 16.05.2024).
23. Сайт навчального центру «ІТ-Столиця» URL: <https://itstolytsa.ua/nashy-kursy/kursy-dlya-detej/razrobotka-igr-na-unity3d> (дата звернення 18.05.2024).
24. U. Technologies, Unity Real-Time Development Platform | 3D, 2D VR & AR Engine, 2021. URL: <https://unity.com>.
25. R. M. Jones, Design and implementation of computer games, ACM SIGCSE Bulletin 32 (2000) 260–264. doi:10.1145/331795.331866.
26. I. Parberry, T. Roden, M. B. Kazemzadeh, Experience with an industry-driven capstone course on game programming, in: Proceedings of the 36th SIGCSE technical symposium on Computer science education - SIGCSE'05, ACM Press, 2005. doi:10.1145/1047344.1047387.
27. E. Sweedyk, R. M. Keller, Fun and games, ACM SIGCSE Bulletin 37 (2005) 138–142. doi:10.1145/1151954.1067485.
28. R. Coleman, M. Krembs, A. Labouseur, J. Weir, Game design & programming concentration within the computer science curriculum, ACM SIGCSE Bulletin 37 (2005) 545–550. doi:10.1145/1047124.1047514.
29. B. Clark, J. Rosenberg, T. Smith, S. Steiner, S. Wallace, G. Orr, Game development courses in the computer science curriculum, Journal of Computing Sciences in Colleges 23 (2007) 65–66. doi:10.5555/1292428.1292440.
30. N. Fachada, N. Códices, Top-down design of a CS curriculum for a computer games BA, in: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, ACM, 2020. doi:10.1145/3341525.3387378.
31. I. Parberry, M. B. Kazemzadeh, T. Roden, The art and science of game programming, in: Proceedings of the 37th SIGCSE technical symposium on Computer science education SIGCSE'06, ACM Press, 2006. doi:10.1145/1121341.1121500.
32. D. Rocco, D. Yoder, Design of a media and gaming sequence for graduates in applied CS, Journal of Computing Sciences in Colleges 22 (2007) 131–137.
33. O. V. Prokhorov, V. O. Lisovichenko, M. S. Mazorchuk, O. H. Kuzminska,

Implementation of digital technology for student involvement based on a 3D quest game for career guidance and assessing students' digital competences, *Educational Technology Quarterly* 2022 (2022) 366–387. doi:10.55056/etq.430.

34. L. Werner, S. Campe, J. Denner, Children learning computer science concepts via Alice game-programming, in: *Proceedings of the 43rd ACM technical symposium on Computer Science Education - SIGCSE'12*, ACM Press, 2012. doi:10.1145/2157136.2157263.
35. J. Distasio, T. Way, Inclusive computer science education using a ready-made computer game framework, in: *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE'07*, ACM Press, 2007. doi:10.1145/1268784.1268820.
36. G. A. Shultz, The story engine concept in CS education, *Journal of Computing Sciences in Colleges* 20 (2004) 241–247. doi:10.5555/1040231.1040263.
37. J.-M. Vanhatupa, Game engines in game programming education, in: *Proceedings of the 11th Koli Calling International Conference on Computing Education Research - Koli Calling'11*, ACM Press, 2011. doi:10.1145/2094131.2094156.
38. A. Estey, J. Long, B. Gooch, A. A. Gooch, Investigating studio-based learning in a course on game design, in: *Proceedings of the Fifth International Conference on the Foundations of Digital Games - FDG'10*, ACM Press, 2010. doi:10.1145/1822348.1822357.