

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
магістра

(назва освітньо-кваліфікаційного рівня)

студента *Сердюка Владислава Юрійовича*
(ПІБ)

академічної групи *122М-19-2*
(шифр)

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

на тему: *Методи, алгоритми та інформаційна технологія для вивчення
правил дорожнього руху за допомогою впровадження елементів ігровізації.*

В.Ю. Сердюк

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинг овою	інституці йною	
розділів кваліфікаційної роботи				
спеціальний	Проф. Мороз Б. І.			
економічний	Доц. Касьяненко Л.В.			
Рецензент				
Нормоконтролер	Доц. Сироткіна О.І.			

Дніпро
2020

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

Завідувач кафедри

Програмного забезпечення комп'ютерних систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

« »

20 20 Року

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра

спеціальності 122 Комп'ютерні науки
 (код і назва спеціальності)

студенту 122М-19-2 Сердюку Владиславу Юрійовичу
 (група) (прізвище та ініціали)

Тема кваліфікаційної роботи Методи, алгоритми та інформаційна технологія
для вивчення правил дорожнього руху за допомогою впровадження елементів
ігрофікації.

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 22.10.2020. №888-с

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – процес впровадження ігрофікації в інформаційну технологію для вивчення правил дорожнього руху.

Предмет досліджень – методи, алгоритми та інформаційна технологія для вивчення правил дорожнього руху.

Мета роботи – підвищення ефективності вивчення правил дорожнього руху шляхом впровадження ігрофікації.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна результатів кваліфікаційної роботи полягає в удосконаленні методів та алгоритмів вивчення правил дорожнього руху за допомогою впровадження ігрофікації.

Практична цінність результатів полягає у тому, що результати роботи, отримані в ході дослідження, можуть застосовуватися як і в розробці ігрофікованих систем, так і в практиці викладання дисциплін, пов'язаних з ігрофікацією.

4 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Аналіз теми та постановка задачі	14.09.2020-25.09.2020
Побудова системи симуляції проблемних випадків на дорозі та теоретичних питань	26.09.2020-01.11.2020
Створення автоматизованої ігрової системи	02.11.2020-10.12.2020

Завдання видав

(підпис)

Мороз Б.І.

(прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Сердюк В.Ю.

(прізвище, ініціали)

Дата видачі завдання: 14.09.2020

Термін подання кваліфікаційної роботи до ЕК 11.12.2020

РЕФЕРАТ

Пояснювальна записка: 85с., 32 рис., 3 дод., 70 джерел.

Об'єкт дослідження: процес впровадження ігрофікації в інформаційну технологію для вивчення правил дорожнього руху.

Предмет дослідження: методи, алгоритми та інформаційна технологія для вивчення правил дорожнього руху.

Мета магістерської роботи: підвищення ефективності вивчення правил дорожнього руху шляхом впровадження ігрофікації.

Методи дослідження. Для вирішення поставлених задач використані методи: об'єктно-орієнтоване програмування, ігрофікація, побудова адаптивних систем.

Наукова новизна результатів дипломної роботи полягає в удосконаленні методів та алгоритмів вивчення правил дорожнього руху за допомогою впровадження ігрофікації.

Практична цінність результатів полягає у тому, що результати роботи, отримані в ході дослідження, можуть застосовуватися як і в розробці ігрофікованих систем, так і в практиці викладання дисциплін, пов'язаних з ігрофікацією.

У розділі «Економіка» проведені розрахунки трудомісткості розробки програмного забезпечення, витрат на створення ПЗ і тривалості його розробки, а також проведені маркетингові дослідження ринку збуту створеного програмного продукту.

Список ключових слів: ігрофікація, теорія ігор, C#, Unity, інформаційна система, правила дорожнього руху, адаптивна система.

ABSTRACT

Explanatory note: 85 p., 32 fig., 3 applications, 70 sources.

Object of research: Process of implementation gamification in road rules learning.

Subject of research: Methods, algorithms and informational system of road rules learning.

Purpose of Master's thesis: improving of efficiency of road rules learning by integrating gamification elements.

Research methods. To perform the tasks, we used methods of the theory of computing system, object-oriented programming, gamification, and empirical generalization based on data.

Originality of research is in the improvement of the road rules learning system by adding gamification elements.

Practical value of the results consists the experimental data obtained in the reseach can be used both in the development of gamification systems and in the practice of teaching disciplines related to gamification.

In the Economics section we calculated the complexity of software development and the costs of software development, the duration of the actual development is calculated.

Keywords: gamification, theory of games, C#, Unity, computing system, road rules.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПДР – правила дорожнього руху;

АТІ – aptitude-treatment interaction;

ЗВ – зворотній відлік;

БД – бали досвіду;

АНС – адаптивна навчальна система.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ.....	12
1.1. Адаптивні системи навчання та їх архітектура	12
1.1.1. Модель користувача.....	13
1.1.2. Підходи до формування адаптивних систем.....	15
1.1.3. Форми адаптації.....	17
1.2. Приклади адаптивних навчальних систем.....	19
1.3. Вибір ігрового рушія	21
1.3.1. Існуючі ігрові рушії.....	22
1.4. Існуючі системи вивчення правил дорожнього руху.....	24
1.5. Висновки до першого розділу.....	27
РОЗДІЛ 2. ІГРОФІКАЦІЯ ЯК СПОСІБ	
МОТИВУВАННЯ КОРИСТУВАЧА.....	28
2.1. Основи ігрофікації.....	28
2.2. Елементи ігрофікації.....	32
2.3. Приклади використання ігрофікації.....	37
2.4. Переваги ігрофікації.....	45
2.5. Критика ігрофікації.....	47
2.6. Висновки до другого розділу.....	51
РОЗДІЛ 3. РОЗРОБКА АДАПТИВНОЇ СИСТЕМИ ВИВЧЕННЯ ПРАВИЛ	
ДОРОЖНЬОГО РУХУ З ЕЛЕМЕНТАМИ ІГРОФІКАЦІЇ	52
3.1. Блок-схема алгоритму.....	52
3.2. Структура інформаційної системи вивчень правил дорожнього руху.....	53
3.2.1. Модуль візуалізації задач правил дорожнього руху.....	55
3.2.2. Адаптивний модуль.....	56
3.2.3. Модуль ігрофікаційних елементів.....	57

	8
3.3. Заміри модулю зацікавленості користувача.....	58
3.4. Заміри прогресу вивчення правил дорожнього руху.....	59
3.5. Висновки до третього розділу.....	60
РОЗДІЛ 4. ЕКОНОМІКА.....	62
4.1. Розрахунок вартості програмного забезпечення.....	63
4.2. Витрати на створення програмного забезпечення.....	65
4.3. Маркетингові дослідження ринку збуту розробленого програмного продукту.....	66
4.4. Оцінка економічної ефективності впровадження програмного забезпечення.....	68
ВИСНОВКИ.....	69
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71
Додаток А. ЛІСТИНГ ПРОГРАМИ.....	77
Додаток Б. ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ.....	84
Додаток В. ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ.....	85

ВСТУП

Актуальність дослідження. В останній час все більше і більше світових компаній починають використовувати елементи ігрофікації в різних сферах. Адаптивність систем навчання стає не лише опціональною особливістю навчальної системи, а обов'язковою в будь-якій системі спрямованій на навчання. Ігри існують з часів виникнення нашої цивілізації. Навіть відеоігри вже існують понад 40 років і представляють індустрію, яка приносить більше 70 мільярдів доларів прибутків на рік. Мільйони людей у всіх куточках Землі проводять по 2-3 години в день за іграми на консолі, персональних ком'ютерах та мобільних телефонах. Багато адаптивних навчальних середовищ показали ефективність при використанні, але вони були швидко покинуті користувачами із-за відсутності чіткої мотивації.

Незважаючи на високий інтерес до цієї індустрії в останній час ігрофікація є доволі молодого сферою досліджень. Термін «ігрофікація» був введений лише в 2002 році Ніком Пелінгом. Розробкою теоретичних підходів і практичних методів для вирішення даного кола проблем займалися такі вчені як Томас Малон, Річард Бартл, Стівен Драпер, Нік Пелінг, Кевін Вермахт та багато інших.

За даними ВООЗ, ситуація з ДТП погіршується. Смертність унаслідок аварій на дорогах зросла до 1,35 млн людей на рік, або в середньому близько 3 700 летальних випадків щодня.

Крім того, десятки мільйонів дістають поранення чи інвалідність щороку або їхнє життя змінюється через отримані травми. Ці втрати істотно впливають на життя сім'ї та суспільства загалом.

Існує багато причин для такої тенденції: швидка урбанізація, погані стандарти безпеки, недотримання правил дорожнього руху, водії під час кермування відволікаються чи перевтомлюються, інші сідають за кермо під впливом наркотиків або алкоголю, перевищують швидкість, нехтують пасками безпеки та захисними шоломами.

Тож, якщо ми поєднаємо методи мотивування користувачів з нагальною проблемою, то ми зможемо зменшити кількість ДТП тим, що більше людей в світі будуть знати ПДР завдяки інформаційній системі, яка їх спонукає до вивчення цих правил.

Мета дослідження: Підвищення ефективності вивчення правил дорожнього руху шляхом впровадження ігрофікації.

Завдання дослідження:

1. Викласти принципи побудови адаптивних систем;
2. Дослідити особливості адаптивних систем;
3. Дослідити елементи ігрофікації;
4. Побудувати інформаційну систему вивчення правил дорожнього руху;
5. Впровадити елементи ігрофікації;
6. Перевірити ефективність впроваджених елементів.

Об'єкт дослідження: процес впровадження ігрофікації вивчення правил дорожнього руху.

Предмет дослідження: методи, алгоритми та інформаційна технологія для вивчення правил дорожнього руху.

Методи дослідження. Для вирішення поставлених задач використані методи: ігрофікації, об'єктно-орієнтоване програмування, побудови адаптивних систем.

Наукова новизна результатів дипломної роботи полягає в удосконаленні методів та алгоритмів вивчення правил дорожнього руху за допомогою впровадження ігрофікації.

Практичне значення. Проведене дослідження буде корисно для науковців, розробників, аспірантів і студентів, що спеціалізуються або цікавляться проблематикою ігрофікації. Результати дослідження можуть застосовуватися як в практиці викладання, так і для організації самостійного навчання.

Особистий внесок автора:

1. Наукові результати роботи отримані автором самостійно;
2. Вибір методів досліджень і технологій реалізації;
3. Реалізація тренажеру правил дорожнього руху засобами ігрового рушія Unity;
4. Розробка теоретичної частини роботи, в якій досліджені і систематизовані знання про існуючі підходи до адаптивних систем навчання та ігрофікації;
5. Оцінка отриманих результатів.

Структура і обсяг роботи. Робота складається з вступу, трьох розділів і висновків. Містить 85 сторінок, в тому числі 60 сторінок тексту основної частини з 32 рисунками, списку використаних джерел з 70 найменуваннями на 6 сторінках, 3 додатка на 8 сторінках.

РОЗДІЛ 1

АНАЛІЗ ТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Адаптивні системи навчання та їх архітектура

Адаптивні освітні системи відстежують важливі характеристики учня та вносять відповідні корективи в навчальне середовище для підтримки та вдосконалення навчання. Метою адаптивних освітніх систем, в контексті цього розділу, є створення навчально-обґрунтованого та життєздатного середовища, яке підтримує навчання для студентів з різними здібностями, вадами розвитку, інтересами, походженням та іншими характеристиками. Завдання досягнення цієї мети значною мірою залежить від точного визначення характеристик конкретного учня або групи учнів - таких як тип і рівень знань, навичок, рис особистості, афективних станів - а потім визначення способу старіння інформації для вдосконалення студента навчання.

Архітектура адаптивних систем складається з чотирьох компонентів (Рис. 1.1.) [1]:

1. Медіа модуль містить всю навчальну інформацію та всі пояснення щодо цієї інформації.
2. Експертний модуль визначає структуру навчання.
3. Адаптаційний модуль – центральний компонент який дає ефект адаптації. Він містить правила вибору контенту та концепту. Ми застосовуємо правила вибору контенту для вибору тих ресурсів з медіа модуля, які зараз найкраще підходять. Правила вибору концепту використовують, щоб вибрати доречні концепти з експертного модуля.
4. Модель користувача містить інформацію про користувача.

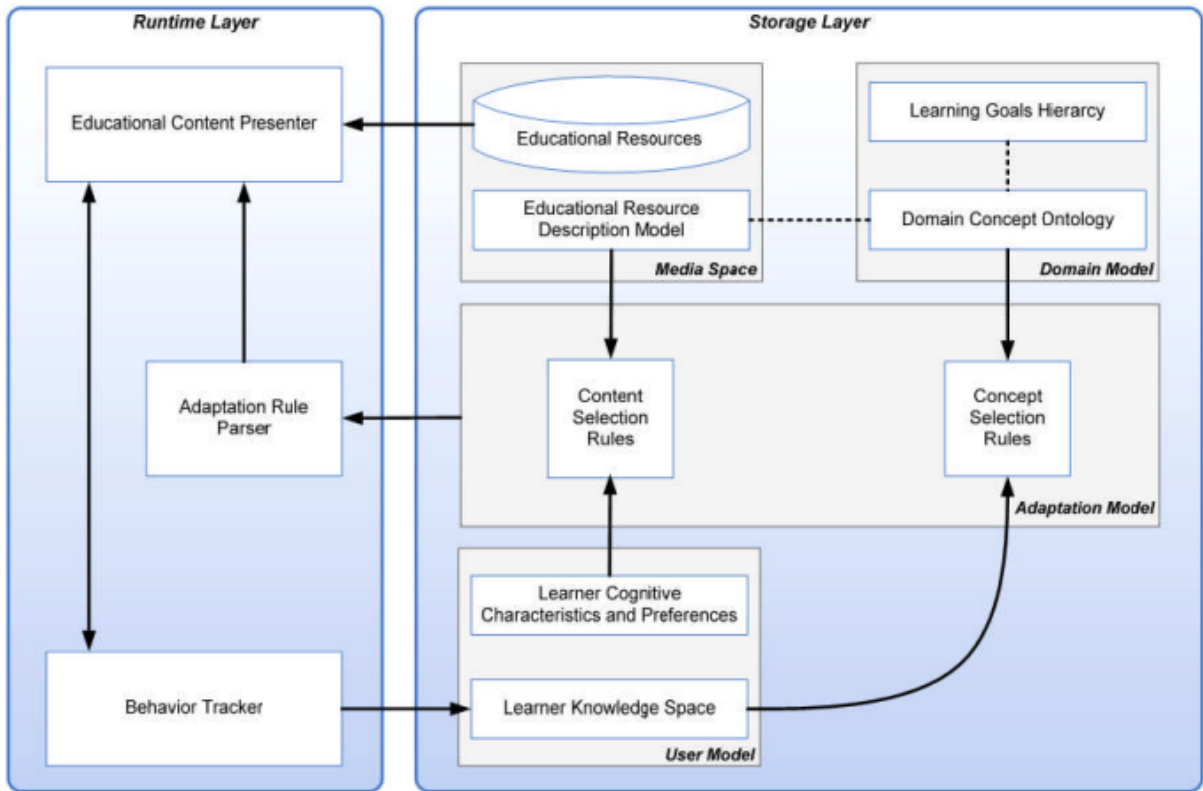


Рис. 1.1. Архітектура адаптивних систем

Загалом, архітектура адаптивних систем має два шари [2]:

1. Шар виконання – відповідає за презентування навчального матеріалу для користувача та спостерігає за учнем для удосконалення інформації про нього;
2. Шар зберігання є основним рушієм, який контролює процес адаптації.

1.1.1. Модель користувача

Модель користувача є однією з найважливіших елементів адаптивної системи. За допомогою цього елемента система розуміє рівень кваліфікації користувача.

Найпростішим способом визначення рівня кваліфікації студента є метод, що застосовується в САТ (комп'ютеризоване адаптивне тестування). У САТ випробовуваний представляє запитання, які вибираються на основі рівня

складності щодо передбачуваного рівня кваліфікації випробуваного. По мірі проходження тесту комп'ютер коригує оцінку випробуваного на основі їх відповідей, безперервно допрацьовуючи оцінку, вибираючи питання з вузького діапазону складності.

Алгоритм оцінки за стилем SAT простий у реалізації. Великий пул питань накопичується та оцінюється відповідно до складності за допомогою експертного аналізу, експериментів або їх поєднання. Потім комп'ютер виконує те, що є, по суті, двійковим пошуком, завжди задаючи випробуваному запитання, яке знаходиться на півдорозі між тим, що комп'ютер вже визначив як максимальний та мінімально можливий рівень кваліфікації суб'єкта. Потім ці рівні пристосовуються до рівня складності запитання, призначаючи мінімальний, якщо випробуваний відповідав правильно, і максимальний, якщо випробуваний відповідав неправильно. Очевидно, що для розробки сценаріїв, коли відповідь випробуваного не вказує на їх справжній рівень кваліфікації, а просто збігається, потрібно вбудувати певний запас помилок. Задавання кількох запитань з одного рівня складності значно зменшує ймовірність оманливої відповіді, а збільшення діапазону понад передбачуваний рівень кваліфікації може компенсувати можливі помилкові оцінки.

Подальшим продовженням виявлення слабких сторін з точки зору концепцій є програмування студентської моделі для аналізу неправильних відповідей. Це особливо стосується питань із множинним вибором.

Розглянемо наступний приклад:

Завдання: Спростити вираз: $2x^2 + x^3$

а) Не можна спростити;

б) $3x^5$;

в) ...;

г) ...;

Очевидно, що студент, який відповідає (б), додає показники ступеня та не розуміє поняття подібних термінів. У цьому випадку неправильна відповідь дає додаткове розуміння, крім простого факту, що вона є неправильною.

1.1.2. Підходи до формування адаптивних систем навчання

З існуючих підходів виділяють 3 головних:

1. Макроадаптивний підхід;

Цей підхід реалізовує адаптивність, дозволяючи різні альтернативи для вибору цілей навчання, змісту навчальної програми та системних систем, заснованих на профілі студента та характеристиках. Ці характеристики є наступними: стилі навчання, цілі навчання студента, системи навчання, рівні досягнень, рівні деталізації тощо. Ці характеристики по-різному впливають на адаптивні системи електронного навчання, такі як діагностування специфічних потреб у навчанні та надання навчальних приписів для них, визначення передумов для змісту навчання, адаптація до стилів навчання студентів та досягнення різних типів навчання. Цілі відповідно до індивідуальних потреб або здібностей студента. Потім заздалегідь сплановані адаптивні стратегії електронного навчання розробляються експертами для різних категорій учнів [6].

2. Підхід АТІ (Aptitude-treatment interaction);

Підхід АТІ пропонує різні типи інструктажів та/або різні типи засобів масової інформації для різних студентів, і вона базується на ідеї, що якби здібності учнів поєднувалися з правильними відношенням, прогнозування результатів навчання було б більш ефективним [7]. Метою підходу АТІ є пошук зв'язків між навчанням та здібностями. З цією метою один із аспектів підходу АТІ стосується місця контролю над процесом навчання, наприклад, ефективніше обмежити контроль для студентів з низьким рівнем попереднього знання. Класи схильностей у дослідженні АТІ, визначені кількома дослідженнями, включають інтелектуальні здібності (такі як міркування та схильність до математики), когнітивні стилі та стилі навчання та (попередні) знання студентів. Крім того, мета когнітивні здібності вважаються важливими в підході АТІ, і дослідники вивчають їх вплив на такі змінні, як зворотний зв'язок та контроль. Нещодавно дослідники визнали здатність когнітивної обробки як

вплив на здатність; таким чином, були розроблені нові адаптивні системи електронного навчання, які включають у свої конструкції наслідки теорії когнітивного навантаження.

3. Мікроадаптивний підхід;

Цей підхід діагностує конкретні навчальні потреби студента під час навчання та, отже, забезпечує відповідні інструкції та тактику для цих потреб. Тобто мікроадаптивні навчальні моделі в основному покладаються на завдання, а не на попередні завдання. Інтелектуальні навчальні системи (ІТС) є прикладом цього підходу. У випадку макроадаптивного навчання диференціація навчальних операцій часто використовується на більших сегментах навчання. Навпаки, мікроадаптивна модель використовує часовий характер вразливостей та характеристик, особливо тих, що динамічно змінюються, таких як афективні стани, помилки відповіді та затримки реакції. Моніторинг поведінки та ефективності користувача може бути використаний для оптимізації навчальних приписів (таких як лікування та послідовності) за оновленою шкалою. Більшість мікроадаптивних моделей коригують навчальний зміст (структуру, виклад, кількість) під час навчання на основі кількісного відображення учнів.

Цей підхід базується на 2 основних процесах:

1. Діагностичний процес: він виявляє характеристики учня (схильності, здібності, мотивація, знання, уподобання, стиль навчання);
2. Процес оптимізації: він оптимізує процес взаємодії користувача з системою при адаптації змісту навчання і послідовності завдань.

На додаток до вищезазначених, деякі дослідники також пропонують інший підхід, конструктивістський - підхід до співпраці. Як випливає з назви, цей підхід підтримує конструктивістське навчання шляхом включення в розроблені відповідні механізми подання знань, міркувань та прийняття рішень та / або співпраці за допомогою адаптивного групування.

1.1.3. Форми адаптації

Відповідно до [3], існує дві форми адаптації: адаптивна презентація та адаптивна навігація:

1. Адаптивна презентація(Рис. 1.2)[3] - адаптує інформацію, яку треба показати користувачу;

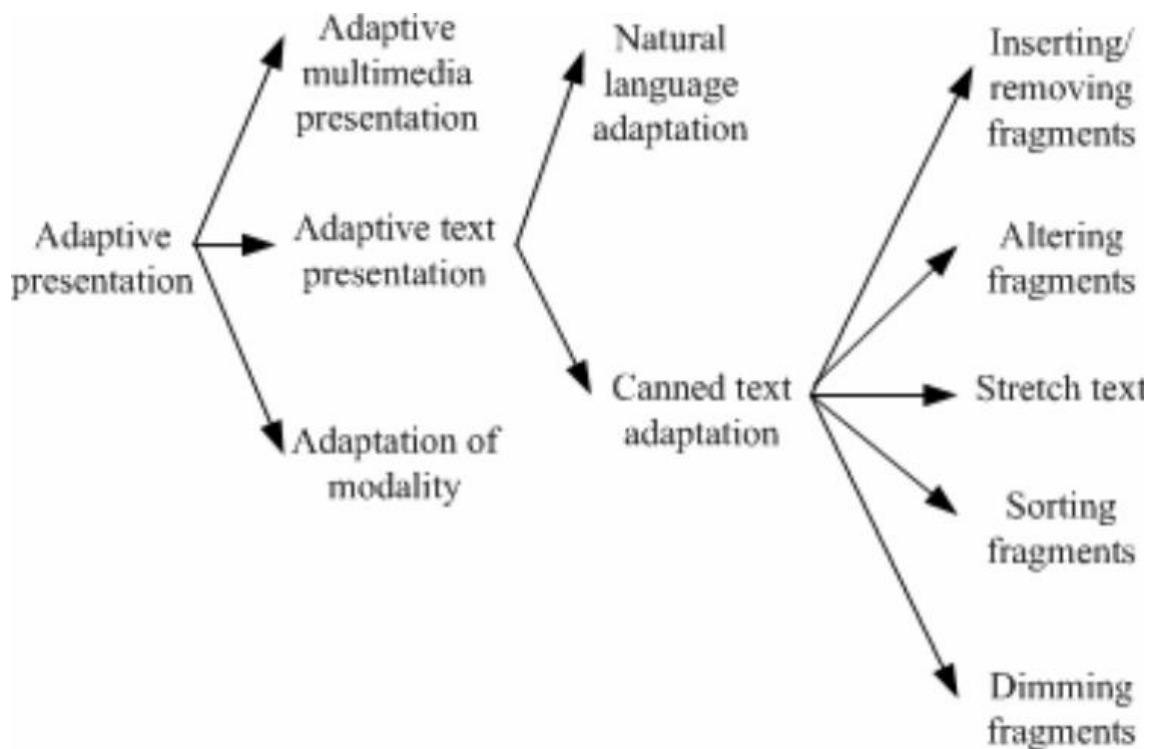


Рис.1.2. Адаптивна презентація

Ідея різних методів адаптивної презентації полягає в адаптації змісту сторінки, на яку здійснюється доступ певним користувачем до поточних знань, цілей та інших характеристик користувача. Для прикладу, кваліфікований користувач може отримати більш детальну та глибоку інформацію, будучи початківцем користувачі можуть отримати додаткові пояснення. У системах гіпермедіа вмістом звичайної сторінки може бути не лише текст, як у класичних гіпертекстових системах, але й набір різноманітних мультимедійних елементів. Таким чином ми могли б розрізнити адаптивне подання тексту та адаптивне мультимедіа.

2. Адаптивна навігація(Рис 1.3.)[3] стосується маніпулювання посиланнями, тим самим користувач сам вибирає, яку інформацію йому покаже система.

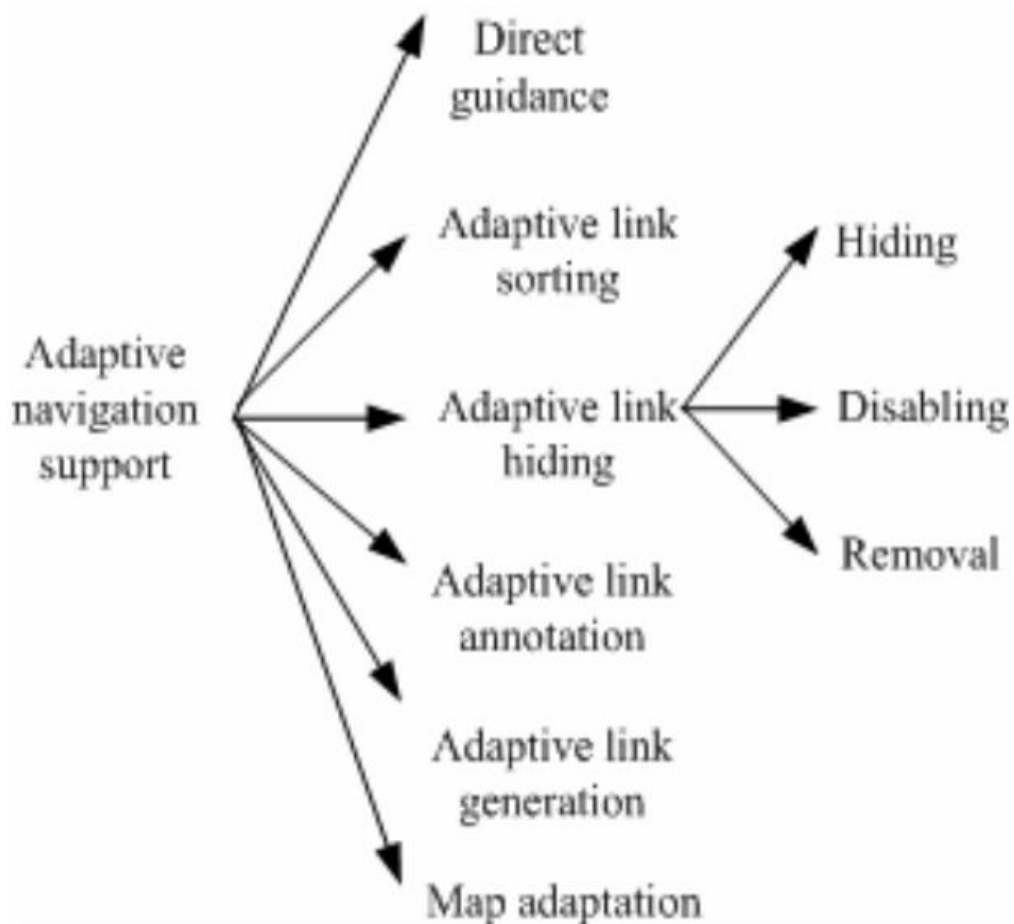


Рис.1.3. Адаптивна навігація

Ідея адаптивних методів навігації полягає в тому, щоб допомогти користувачам знайти свої шляхи в гіперпросторі шляхом адаптації способу подання посилань на цілі, знання та інші характеристики окремого користувача. Хоча цей напрямок досліджень є дуже новим, існує ряд цікавих методик, які вже запропоновані та впроваджені.

1.2. Приклади адаптивних навчальних систем

Макроадаптивний є найстарішим із чотирьох вже згаданих підходів. Ранніми макроадаптивними прикладами адаптивних систем навчання (до Другої світової війни) є “Індивідуальні навчальні плани” (ПНП), План Далтона та План WinnetkaPlan. Протягом 60-х та 70-х років вдосконалення процедур аналізу завдань, поява критеріального тестування та теорія ієрархії навчання Ганьє, ймовірно, вплинули на розвиток індивідуальних систем навчання. Перша комп'ютерна система технологічного навчання (TEL) була запрограмована логікою для автоматизованих навчальних операцій (PLATO). PLATO, розроблена в 1960-х і 1970-х роках, вважається однією з найуспішніших систем в історії TEL[8]. Система управління навчанням Платона, еволюція оригінальної системи PLATO, може забезпечити деяку рудиментальну адаптацію: при оволодінні всіма цілями модуль був досягнутий, студент міг перейти до наступного модуля. Перша система, навмисно розроблена для адаптивного навчання, була розроблена Россом та Моріссоном, включаючи прогнозування потреб учнів у навчанні; однак ці потреби можна було діагностувати лише на етапі перед викладанням, а не під час навчання. На додаток до цих макроадаптивних характеристик, ця система також поєднувала деякі особливості, пов'язані з мікроадаптивним підходом.

Керрієранд Йонассен[9] запропонував восьмикрокову модель для надання практичних вказівок щодо застосування моделі АТІ до дизайну навчальних програм, які насправді нагадував навчальну модель дизайну Діка і Кері. Крім того, вони запропонували методи чотирьох адаптацій, засновані на концептуальному відображенні між навчальними стратегіями та характеристиками студентів:

1. коригувальні дії для надання додаткових інструкцій;
2. використання великих літер / переважних дій для надання інструктажу на спосіб, що відповідає бажаному студентом режиму сприйняття чи міркування;

3. компенсаційний, для заміщення деяких вимог до обробки навчальної задачі (наприклад, здатність до когнітивної обробки);
4. виклик для мотивації учнів використовувати та розробляти нові режими обробки.

Парк та Лі стверджували[10], що, хоча, здається, ця модель має високу практичну цінність без теоретично узгоджених та емпірично простежуваних зв'язків між різними змінними учнів та без чітко визначених типів та рівнів вимог до навчання та навчальних стратегій для різних завдань, одне лише застосування цієї моделі навряд чи дасть кращі результати, аніж не адаптивні навчальні системи, що є очевидним. Одна з останніх тенденцій щодо підходу АТІ орієнтована на пізнавальне навантаження учня як параметр адаптації.

Наприклад, е-TPCK - це система адаптивного навчання, яка була розроблена та розроблена спеціально для стимулювання постійного вдосконалення вчителів знань про те, як навчати певній темі за допомогою конкретних засобів ІКТ. Підкреслюється, що е-TPCK був розроблений для того, щоб бути пізнавальним партнером для навчання викладачів будівельних лісів шляхом персоналізації вмісту, представленого їм у формі сценаріїв дизайну, що застосовуються до ІКТ. Серед іншого, система стосується вибору адаптивного сценарію проектування із спільним контролем між користувачем та системою та навчальним шляхом її користувачів на основі оцінок їх сприйнятих розумових зусиль щодо представленого їм сценарію проектування.

Ранні приклади мікроадаптивного підходу базуються на запрограмованих інструкціях, а найновіші моделі застосовують методи штучного інтелекту. В Kickmeier-Rust[11] та інших, мікроадаптивність дозволила моніторинг та інтерпретацію поведінки учня у віртуальному світі гри, а саме ELEKTRA, шляхом персонального аналізу наданих вказівок та зворотного зв'язку значущим чином, які потрапили в потоки гри, не погіршуючи занурення або перебіг досвід гри. Мікроадаптивність була використана для інтерпретації поведінки учня при вирішенні проблем таким чином, щоб він відповідав ідеї

гри. З цією метою відповідна модель домену включала стани компетенції, стани проблем та можливі шляхи навчання.

1.3. Вибір ігрового рушія для розробки.

Ігровий рушій – це середовище для розробки програмного забезпечення, призначене для створення відеоігор. Розробники використовують ігрові рушії для побудови ігор для консолей, мобільних пристроїв та персональних комп'ютерів. Основна функціональність, яку зазвичай надає ігровий движок, включає в себе механізм візуалізації ("візуалізатор") для 2D або 3D графіки, фізичний механізм або виявлення зіткнень (і відповідь на зіткнення), звук, сценарії, анімацію, штучний інтелект, мережу, потокове передавання, пам'ять управління, створення потоків, підтримка локалізації, графік сцени та може включати підтримку відео для кінематики. Реалізатори часто економлять на процесі розробки ігор, повторно використовуючи / адаптуючи, значною мірою, один і той же ігровий движок для створення різних ігор або при перенесенні ігор на різні платформи.

Ігрові рушії - це багаторазові компоненти, які розробники використовують для побудови фреймворку. Це дає їм більше часу, щоб зосередитись на таких унікальних елементах, як моделі персонажів, текстури, взаємодія об'єктів тощо. Однак, як і раніше, існує велика кількість великих компаній і навіть незалежних команд, які створюють свої двигуни. Для цього потрібно використовувати інтерфейси програм, такі як DirectX, OpenGL та XNA, а також комерційні та відкриті бібліотеки, що містять фізику, графіку сцен та бібліотеки графічного інтерфейсу.

Створення одного з найкращих двигунів для відеоігор є непростим завданням, але часом необхідним, якщо гра досить інша, і жоден існуючий движок не буде працювати.

1.3.1. Існуючі ігрові рушії

З існуючих ігрових рушіїв виділяють наступні:

Одним з найпопулярніших і широко використовуваних ігрових рушіїв є Unreal Engine 4 (Рис. 1.4.) від Epic Games. Оригінальна версія була випущена в 1998 році, але незважаючи на це навіть через 17 років вона продовжує використовуватися для найбільших ігор щороку. Головною мовою програмування є C++. Серед відомих назв, зроблених за допомогою Unreal Engine, - Star Wars Jedi: Fallen Order, Gears of War 4, Fortnite, Dead by Daylight.

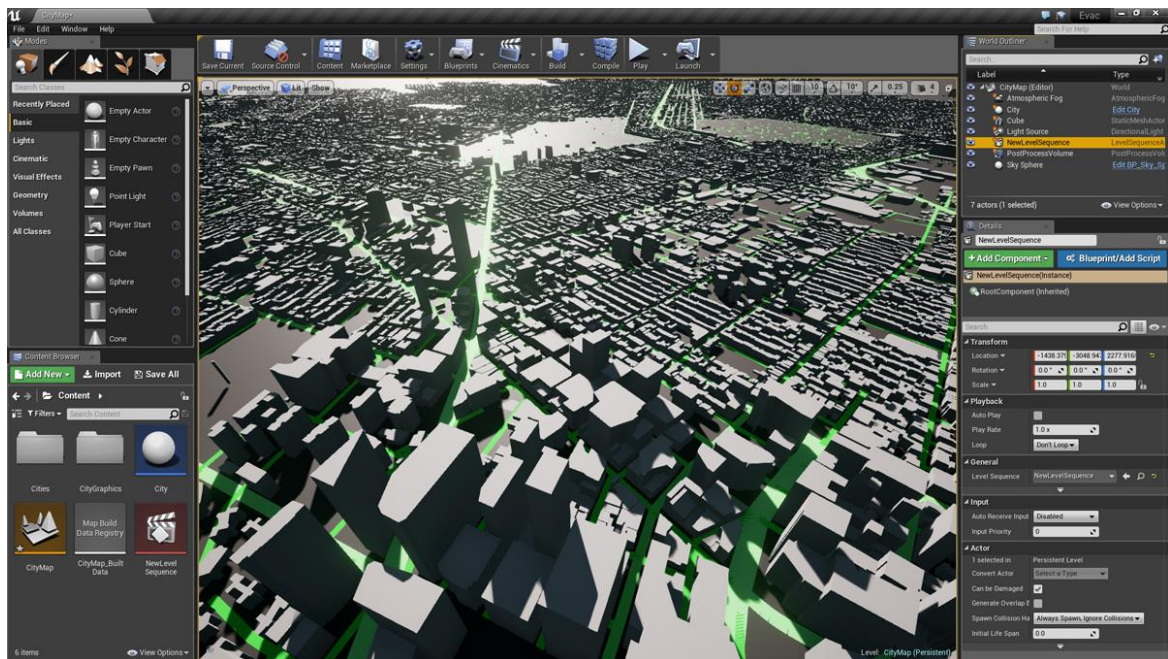


Рис.1.4 Інтерфейс Unreal Engine 4

Найбільшим плюсом Unreal Engine є його здатність модифікуватися настільки, що з ігор можна зробити дуже унікальний досвід. Останню версію, Unreal Engine 4, вважають найпростішою у використанні.

Серед головних плюсів Unreal Engine:

1. Відкритий код ігрового рушія;
2. Готові ассети;
3. Blueprints;

4. Фотореалістичні зображення;
5. Кросс-платформенність.

Один з найпростіших в використанні є Unity (Рис. 1.5), мультиплатформенний ігровий рушій, який дозволяє легко створювати інтерактивний 3D-вміст. Багато незалежних розробників використовують Unity за чудову функціональність, високоякісний вміст та можливість використовувати його майже для будь-якого типу гри. Серед останніх відомих назв, створених спільно з Unity, є Fall Guys, Syberia 3, Hearthstone. Головною мовою програмування є C# та Javascript.

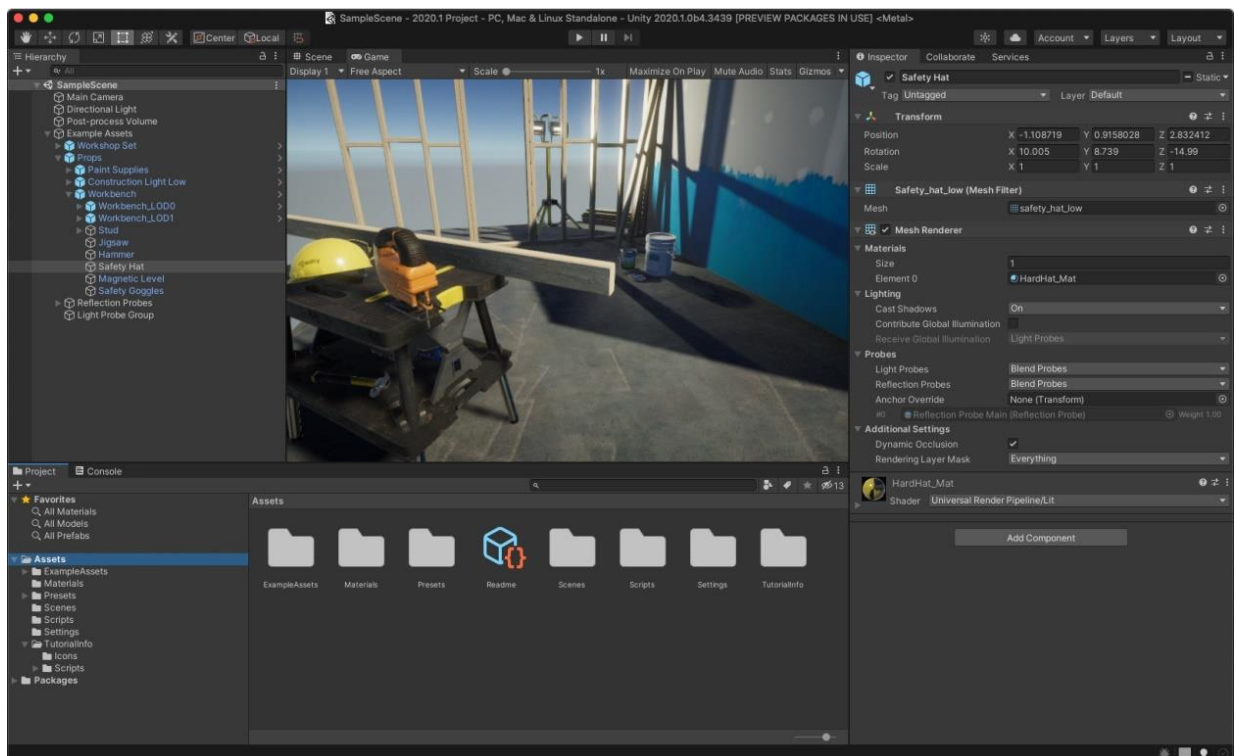


Рис. 1.5. Інтерфейс Unity

Серед головних плюсів Unity:

1. Мультиплатформенність;
2. Ефективність та надійність;
3. Система підтримки мультиплеєру;
4. Гарні візуальні ефекти;
5. Аналітика.

Löve – міжплатформений ігровий рушій з відкритим вихідним кодом для розробки 2D-відеоігор. Механізм написаний на C ++ і використовує Lua як мову сценаріїв. Видається під ліцензією zlib. LÖVE досі підтримується оригінальними розробниками. Цей двигун часто зустрічається у композиціях змагань з розробки відеоігор, таких як міжнародний конкурс Ludum Dare. Проте Löve нам не підходить тому що нам треба 3D аспект в нашій інформаційній технології.

1.4. Існуючі системи вивчення правил дорожнього руху

Найпершим тренажером, який слід розглянути є «Онлайн-тренажер із ПДР» від Кабінету міністрів України (Рис. 1.6.).

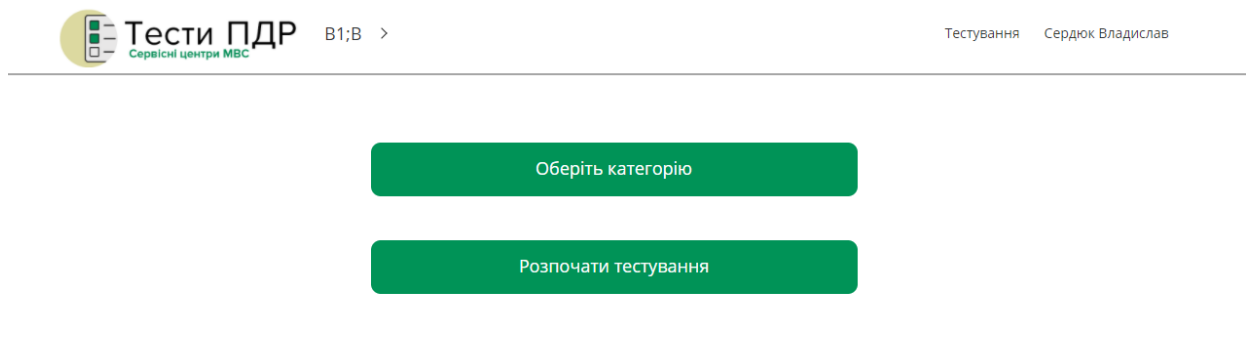


Рис 1.6. Онлайн-тренажер із ПДР від Кабінету міністрів України

Адаптивність системи мінімалізована до вибору категорії. Проходження тесту закінчується після 4 неправильних спроб без пояснень сутності помилок, помилки користувача ніяк не аналізуються. Елементів ігрофікації в цій системі немає. Інші браузерні ресурси пропонують лише тести з ПДР, які абсолютно не пристосовуються до відповідей чи знань користувача та не включають елементів ігрофікації.

З україномовних додатків на Андроїд жоден з них не включає в себе адаптивності та елементів ігрофікації (Рис. 1.7.) .

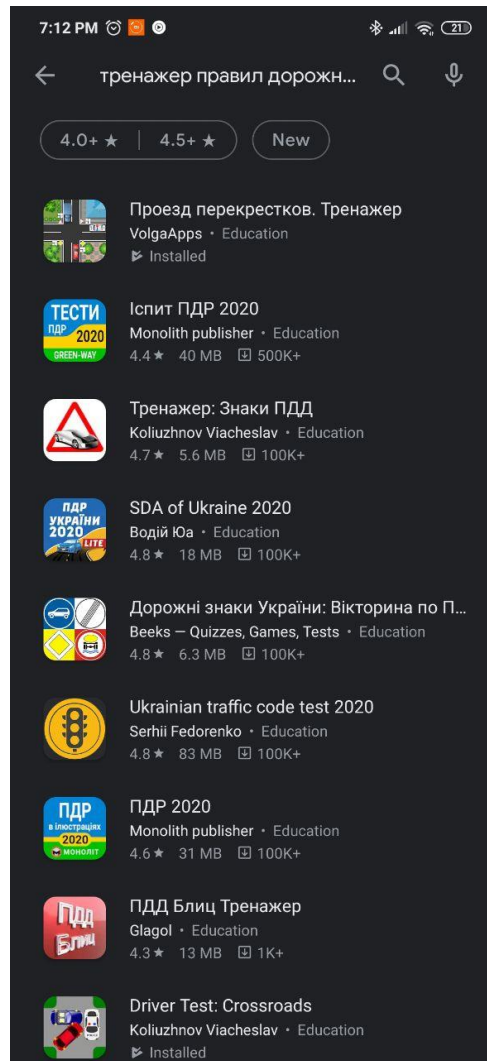


Рис. 1.7. Результати пошуку «тренажер правил дорожнього руху»

З англomовних та російськомовних додатків слід відзначити «Driver Test:Crossroads» від Колюжнова В'ячеслава (Рис. 1.8.) [52]. Проте адаптивність системи до користувача мінімальна та елементи ігрофікації зовсім відсутні. Інтерфейс також не підтримує українську мову та не адаптованих до українських ПДР. Пояснень помилок чи їх аналізу за час користування додатком не було помічено.



Рис. 1.8. Скріншот з додатку «Driver Test: Crossroads»

Інших систем, які займаються навчанням правилам дорожнього руху, які потребували б уваги серед додатків на мобільні пристрої не знайшли.

При пошуку аналогічних програм для операційної системи Windows в Microsoft Store ні платних, ні безоплатних не було знайдено (Рис. 1.9.).

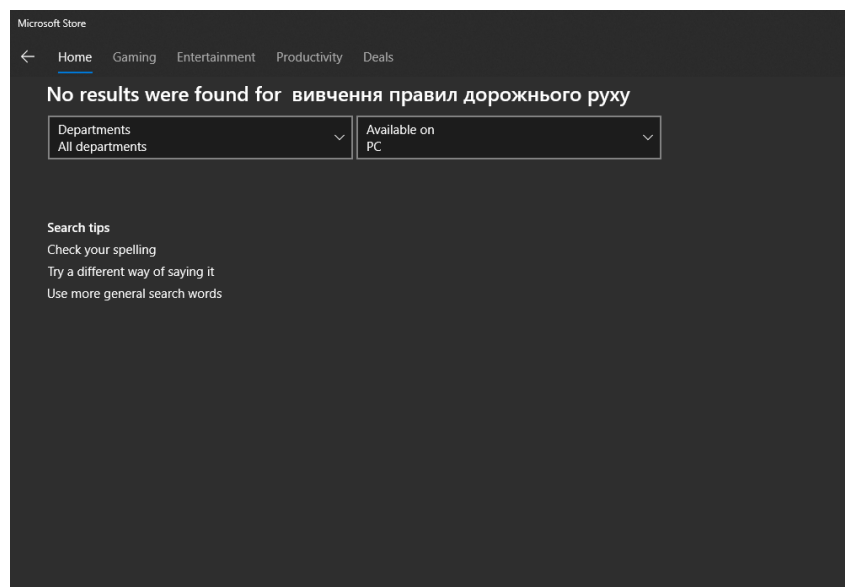


Рис. 1.9. Пошук аналогічних програм в Microsoft Store

1.5. Висновки до першого розділу

Багато навчальних середовищ показали ефективність при використанні, але вони були швидко покинуті користувачами із-за відсутності чіткої мотивації. В навчальних системах, ігрофікація – це спосіб мотивувати учнів та активізувати їхню участь у навчальній діяльності, додаючи елементи ігрофікації. Але мало хто звертає увагу на індивідуальні відмінності між учнями, як гравцями в системі.

З форм адаптацій виділили:

1. Адаптивну презентацію;
2. Адаптивну навігацію.

Для виконання було обрано адаптивну презентацію.

З підходів до формування адаптивних систем було обрано мікроадаптивний підхід.

Серед існуючих ігрових рушіїв виділили найбільш 3 доречних рушія:

1. Unreal Engine 4;
2. Unity;
3. LOVE.

Для розробки інформаційної системи було обрано ігровий рушій Unity.

Проаналізувавши існуючі системи вивчення правил дорожнього руху було виявлено відсутність систем з адаптивною презентацією та примітивність адаптивної навігації. Елементів ігрофікації в цих системах виявлено не було.

РОЗДІЛ 2

ІГРОФІКАЦІЯ ЯК СПОСІБ МОТИВУВАННЯ КОРИСТУВАЧА

2.1. Основи ігрофікації

Останні роки показали швидке розповсюдження масового програмного забезпечення, яке надихається відеоіграми. Зазвичай термін «ігрофікація» пов'язують із значним набором існуючих концепцій та досліджень взаємодії людини з комп'ютером та ігрових досліджень, таких як серйозні ігри, загальнодоступні ігри, ігри з альтернативною реальністю або грайливий дизайн. Однак не зрозуміло, як «ігрофікація» пов'язана з ними та чи позначає воно нове явище.

Ігрофікація – це застосування елементів ігрового дизайну та ігрових принципів у неігрових контекстах[12]. Його також можна визначити як сукупність видів діяльності та процесів для вирішення проблем, використовуючи або застосовуючи характеристики ігрових елементів. У ігрофікації зазвичай використовуються елементи ігрового для поліпшення залучення користувачів, організаційна продуктивність, потік, навчання, краудсорсинг, збереження знань, набір та оцінка працівників, простота використання, корисність систем, фізичні вправи, порушення правил дорожнього руху, апатія виборців та інше. Збірник досліджень з питань ігрофікації показують, що більшість досліджень з питань ігрофікації чинять позитивний вплив на людей[13]. Однак існують індивідуальні та контекстуальні відмінності.

Методи ігрофікації мають на меті використати природні бажання людей до спілкування, навчання, оволодіння, змагання, досягнень, статусу, самовираження, альтруїзму чи закритості або просто їхньої реакції на формування ситуації як гри. Ранні стратегії ігрофікації використовують винагороду для гравців, які виконують бажані завдання або змагання, щоб залучити гравців. Типи винагород включають бали, значки або рівні

досягнення, заповнення індикатора прогресу, або надання користувачеві віртуальної валюти. Визначення винагороди за виконання завдань видимим для інших гравців або надання дошок лідерів – це способи заохочення гравців до змагань.

Інший підхід до ігрофікації – зробити так, щоб існуючі завдання були більш схожими на ігри[14]. Деякі прийоми, що використовуються у цьому підході, включають додавання значущого вибору, використання навчальних посібників, збільшення викликів та додавання розповіді.

Хоча термін "ігрофікація" вперше з'явився в Інтернеті в контексті комп'ютерного програмного забезпечення в 2008 році, він набув популярності лише в 2010 році. Ще до того, як цей термін увійшов у вживання, інші сфери запозичення елементів у відеоігор були поширеними. Термін "ігрофікація" вперше набув широкого вживання в 2010 році в більш конкретному сенсі, що стосується включення соціальних аспектів / аспектів ігор у програмне забезпечення (Рис 2.1) .

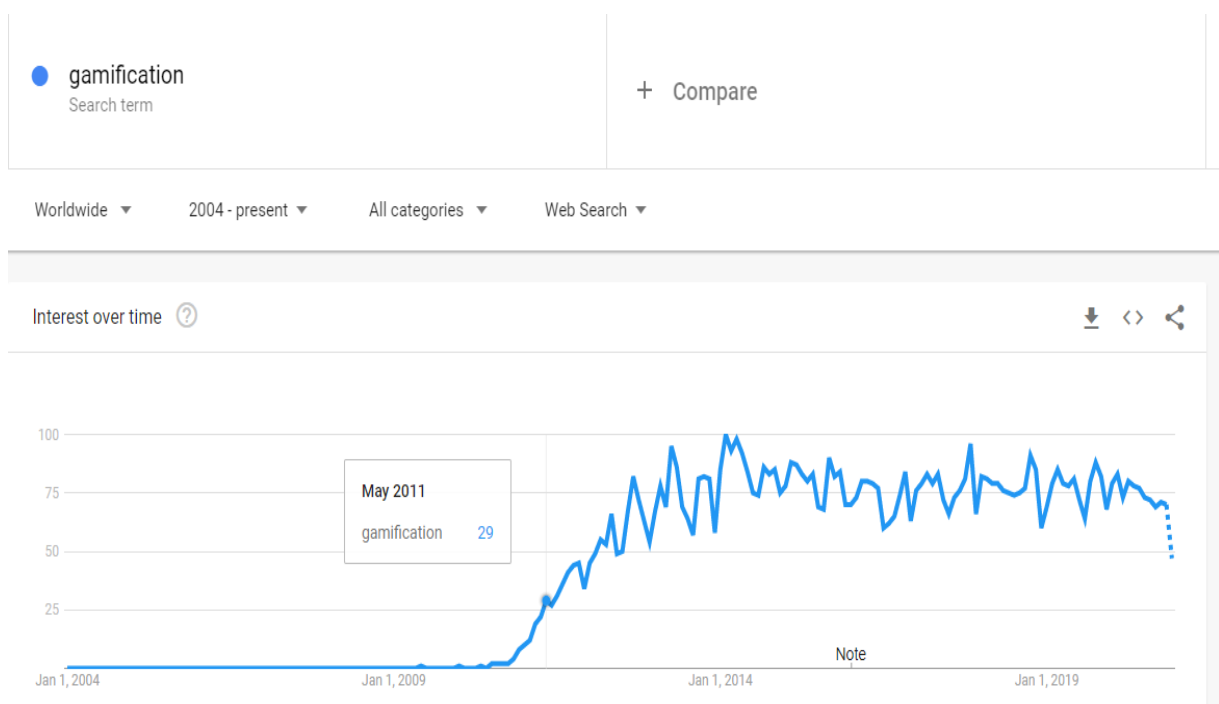


Рис. 2.1. Тренди пошукової системи «Google» для терміну «ігрофікація»

Ця техніка привернула увагу венчурних капіталістів, один з яких сказав, що вважає ігрофікацію найбільш перспективною сферою ігор. Інший зауважив, що половина всіх компаній, які шукають фінансування для споживчих програмних програм, згадували дизайн ігор у своїх презентаціях.

Деякі дослідники вважають гейміфікацію тісно пов'язаною з попередньою роботою з адаптації елементів і технік ігрового дизайну до неігрового контексту. Нельсон доводить зв'язок як з радянською концепцією соціалістичної конкуренції, так і з американською тенденцією управління «задоволенням від роботи»[15]. Фукс зазначає[16], що ігрофікація може бути зумовлена новими формами сміливих інтерфейсів. Ігрофікаторні конференції також включили ігрове моделювання із більшою силою.

На додаток до компаній, які використовують цю техніку, низка підприємств створила ігрові платформи. У жовтні 2007 року Bunchball (Рис 2.2.) , за підтримки Adobe Systems Incorporated, стала першою компанією, яка надала ігрову механіку як послугу, на Dunder Mifflin Infinity, сайті спільноти для телешоу The Office на телеканалі NBC. Серед клієнтів Bunchball - Playboy, Chiquita, Bravo та The USA Network .



Рис. 2.2. Bunchball, як перша комерційна ігрофікаційна компанія

Компанія Badgevillе, яка пропонує послуги з ігрофікації, була розпочата наприкінці 2010 року і за перший рік своєї діяльності залучила 15 мільйонів доларів венчурного фінансування. Серед великих корпоративних фірм SAP AG, Microsoft, IBM, SAP, LiveOps, Deloitte та інші компанії почали використовувати ігрофікацію в різних додатках та процесах.

Gamification 2013, захід, що вивчає майбутнє гейміфікації, відбувся в університеті Ватерлоо, Стратфорд, в жовтні 2013 року.

Завдяки дедалі більшому прийняттю ігрофікації та її природі як агрегатора даних, до ігрофікації можуть застосовуватися численні юридичні обмеження. Деякі посиляються на використання віртуальних валют та віртуальних активів, закони про конфіденційність даних та захист даних або законодавство про працю[17].

Використання віртуальних валют (Рис 2.3.), на відміну від традиційних платіжних систем, не регулюється. Правова невизначеність, пов'язана із схемами віртуальних валют, може становити виклик для органів державної влади, оскільки ці схеми можуть використовуватись злочинцями, шахраями та відмивачами грошей для здійснення своєї незаконної діяльності[18].



Рис. 2.3. Віртуальна валюта в грі «FarmVille»

2.2. Елементи ігрофікації

Елементи ігрового дизайну є основними будівельними елементами програм ігрофікації. Серед типових елементів дизайну гри – бали, значки, таблиці лідерів, графіки ефективності, сюжет, час, аватари та товариші по команді[19].

Бали є основними елементами безлічі ігор та ігрових додатків[20]. Зазвичай вони винагороджуються за успішне виконання певних дій у ігрофікованому середовищі, і вони служать для чисельного відображення прогресу гравця. Різні види балів можна диференціювати, напр. бали досвіду (Рис. 2.4.), бали, які можна викупити, або репутаційні бали, як і різні цілі, яким служать бали. Однією з найважливіших цілей пунктів є надання зворотного зв'язку. Бали дозволяють виміряти поведінку гравців у грі, і вони слугують постійним і негайним зворотним зв'язком та нагородою.



Рис 2.4. Бали досвіду на офіційному сайті УЄФА

Значки досягнень визначаються як наочне зображення досягнень[21], і їх можна заробляти та збирати в середовищі ігрофікації. Вони підтверджують досягнення гравців, символізують їх заслуги і наочно демонструють досягнення ними рівнів чи цілей. Заробляння значка може залежати від конкретної

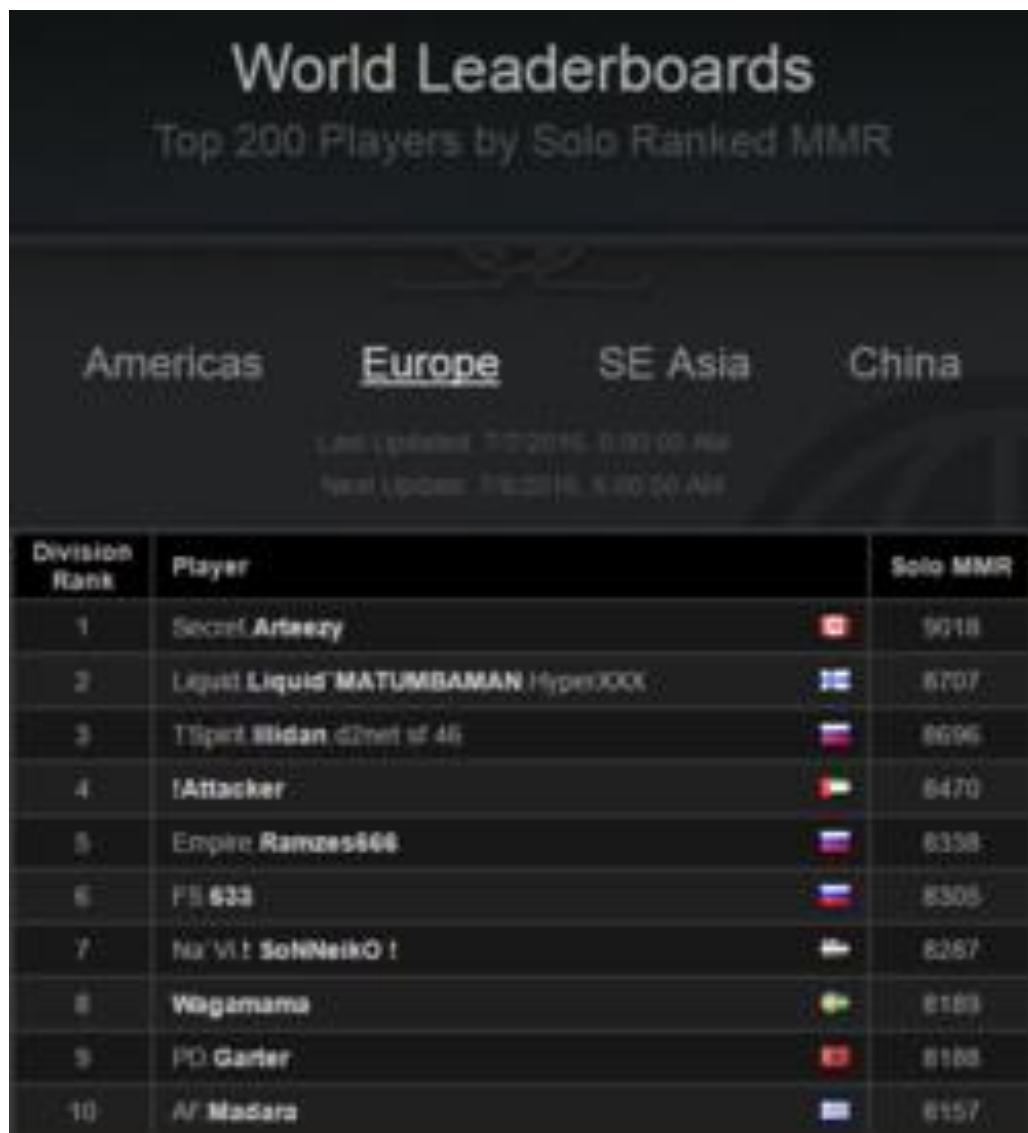
кількості балів або від певної діяльності в грі. Значки мають багато функцій, слугуючи цілями, якщо передумови їх перемоги відомі гравцеві або як віртуальні символи стану. Так само, як бали, значки також забезпечують зворотний зв'язок, вказуючи на те, як гравці зробили завдання. Значки можуть впливати на поведінку гравців, змушуючи їх вибирати певні маршрути та завдання, щоб заробити значки, пов'язані з ними. Окрім того, оскільки значки символізують членство в групі тих, хто володіє цим значком, вони також можуть чинити соціальний вплив на гравців та співавторів, особливо якщо вони рідкісні або їх важко заробити. Як приклад, компанія Playstation спільно з розробниками ігор створили систему значків, які показують наскільки людина пройшла ту чи іншу гру. Виділяють чотири види значків (Рис 2.5.):

1. Бронзовий – за проходження гри на 30%
2. Срібний – за проходження на 60%;
3. Золотий – за проходження на 90%;
4. Платиновий – за повне проходження гри.



Рис. 2.5. Значки проходження гри на Playstation

Таблиці лідерів (Рис. 2.6.) класифікують гравців відповідно до їх відносного успіху, вимірюючи їх за певним критерієм успіху[22]. Таким чином, таблиці лідерів можуть допомогти визначити, хто найкраще працює у певній діяльності, і, отже, є конкурентними показниками прогресу, які пов'язують власні результати гравця з результатами інших. Однак мотиваційний потенціал таблиць лідерів неоднозначний.



The image shows a screenshot of the Dota 2 World Leaderboards interface. The title is "World Leaderboards" and the subtitle is "Top 200 Players by Solo Ranked MMR". The "Europe" division is selected. The table below lists the top 10 players with their ranks, names, and MMR values.

Division Rank	Player	Solo MMR
1	Secret_Artezy	8718
2	Liquid Liquid MATUMBAMAN HyperXXX	8707
3	Tispit Bidan dZnet w 46	8696
4	lAttacker	8470
5	Empire Ramzes666	8358
6	PS 633	8306
7	Na'Vi t SolNeiko 1	8267
8	Wagamama	8189
9	PO_Garber	8168
10	Al Madara	8157

Рис. 2.6. Таблиця лідерів в грі Dota 2

Вербах та Хантер[21] розглядають їх як ефективних мотиваторів, якщо до наступного рівня чи позиції залишилось лише кілька очок, але як демотиватори, якщо гравці опиняються в нижній частині таблиці лідерів.

Конкуренція, спричинена таблицями лідерів, може створити соціальний тиск для підвищення рівня залучення гравця і, отже, може мати конструктивний вплив на участь та навчання. Однак ці позитивні наслідки конкуренції більш вірогідні, якщо відповідні конкуренти мають приблизно однаковий рівень продуктивності[23].

Графіки ефективності (Рис 2.7.), які часто використовуються в симуляторах або стратегічних іграх, надають інформацію про ефективність гравців порівняно з їх попередніми показниками під час гри. Таким чином, на відміну від таблиць лідерів, графіки продуктивності не порівнюють результативність гравця з іншими гравцями, а натомість оцінюють власну ефективність гравця з часом. На відміну від соціального еталонного стандарту таблиць лідерів, графіки ефективності базуються на індивідуальному еталонному стандарті. Графічно відображаючи продуктивність гравця протягом певного періоду, вони зосереджуються на вдосконаленнях. Теорія мотивації постулює, що це виховує орієнтацію на оволодіння наукою, що особливо корисно для навчання[20].

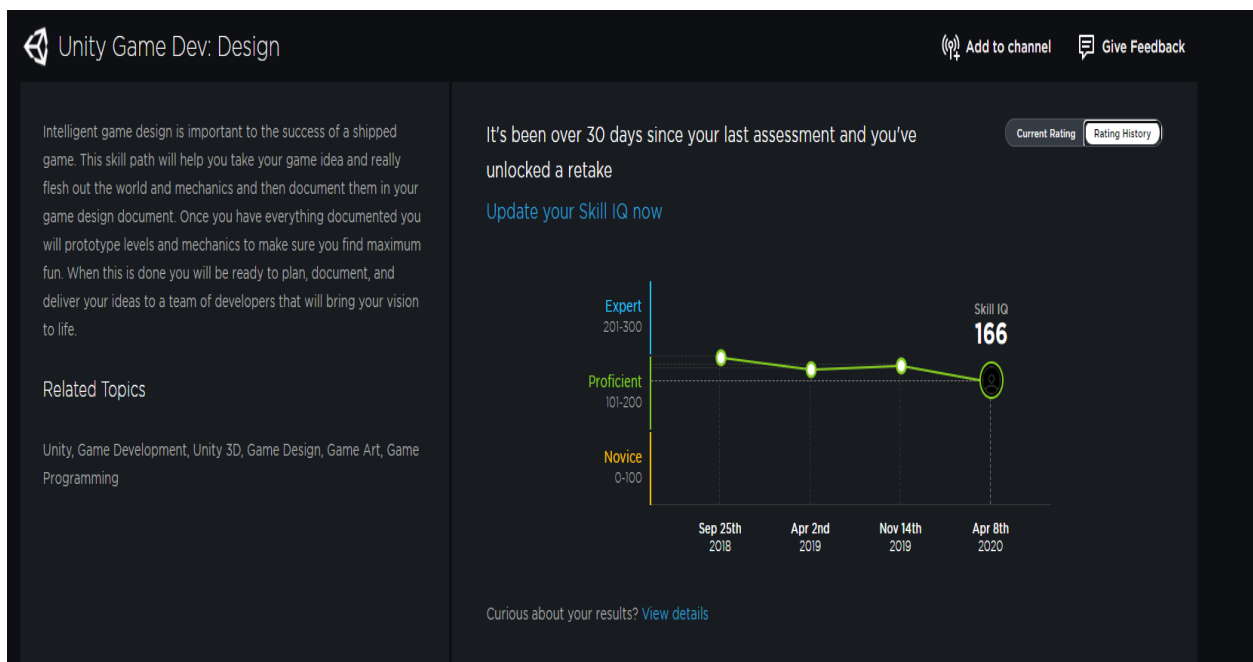


Рис. 2.7. Графік ефективності на навчальному сайті Pluralsight

Сюжет – це елемент ігрофікації, який не стосується продуктивності гравця. Розповідний контекст, в який може бути вбудований ігрофікований додаток, контекстуалізує дії та персонажів у грі та надає їм значення поза простими пошуками очок та досягнень. Історія може передаватися за назвою гри (наприклад, *Space Invaders*) або за допомогою складних сюжетних ліній, характерних для сучасних рольових відеоігор (наприклад, *The Elder Scrolls Series*). Наративні контексти можуть бути орієнтовані на реальні, неігрові контексти або виступати аналогіями реальних умов. Останні можуть збагатити нудний, ледь стимулюючий контекст, а отже, надихати та мотивувати гравців, особливо якщо історія відповідає їхнім особистим інтересам. Таким чином, сюжет також є важливою частиною в програмах ігрофікації, оскільки вони можуть змінити значення справжніх дій.

Зворотний відлік – поширений елемент у настільних іграх, таймери (відлік загального часу) та годинник зворотного відліку створюють відчуття актуальності. Навіть використовуючи розклад подій, наприклад, перш ніж робити В і С, я повинен виконати А, допомагає зосередити увагу учня на завданні.

Аватари (Рис. 2.8.) – це візуальні зображення гравців в ігровому або ігровому середовищі. Зазвичай їх вибирає або навіть створює гравець [24]. Аватари можуть бути спроектовані просто як просто піктограма, або вони можуть бути складно-анімованими, тривимірними зображеннями. Їх основною формальною вимогою є те, що вони безпомилково ідентифікують гравців та виділяють їх серед інших аватарів, якими керує людина чи комп'ютер. Аватари дозволяють гравцям прийняти або створити іншу ідентичність і, у спільних іграх, стати частиною спільноти.



Рис. 2.8. Аватар в системі XBOX

Соціалізація - товариші по команді, будь то інші реальні гравці або віртуальні персонажі, що не грають, можуть викликати конфлікт, конкуренцію чи співпрацю. Останньому можна сприяти, особливо шляхом введення команд, тобто шляхом створення визначених груп гравців, які працюють разом задля спільної мети.

2.3. Приклади використання ігрофікації

Ігрофікація застосовувалася майже до всіх аспектів життя. Приклади ігрофікації у бізнес-контексті включають армію США, яка використовує військовий симулятор армії Америки як інструмент вербування, та гру кренделів M & M, запущену в 2013 році для посилення маркетингової кампанії компанії з кренделями, створивши цікавий спосіб «стимулювати зацікавлення користувачів». Інший приклад можна побачити в американській системі освіти. Студенти класифікуються у своєму класі на основі їх заробленого середнього балу (GPA), який можна порівняти з отриманням високих балів у відеоіграх[25]. Студенти також можуть отримувати заохочення, такі як почесне згадування в списку декана, почесний список та стипендії, які еквівалентні підвищенню рівня персонажа відеоігор або зароблянню віртуальної валюти або інструментів, що збільшують успіх в грі. Процеси подання заявок на роботу

іноді використовують ігрофікацію як спосіб найму працівників, оцінюючи їх придатність за допомогою анкет та міні-ігор, що імітують фактичне робоче середовище цієї компанії.

Ігрофікація широко застосовується в маркетингу. Понад 70% опитаних у 2013 році компаній Forbes Global 2000 заявили, що планують використовувати гейміфікацію для цілей маркетингу та утримання клієнтів [26]. Наприклад, у листопаді 2011 року австралійське партнерство з питань мовлення та онлайн-медіа Yahoo!7 запустило свій мобільний додаток Fango/SAP (Рис. 2.9.), який телеглядачі використовують для взаємодії з шоу за допомогою таких методів, як опитування та значки.

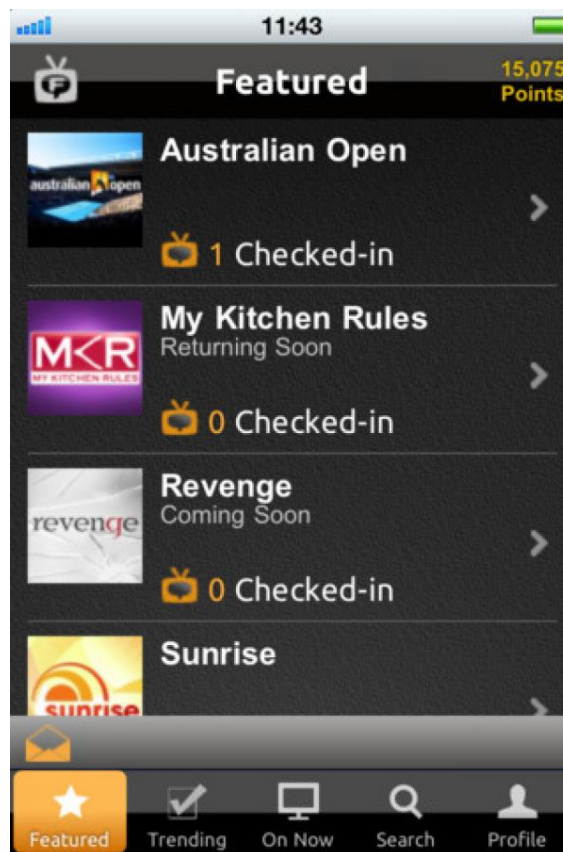


Рис. 2.9. Скріншот головного меню мобільного додатку Fango

Ігрофікація також застосовується в програмах лояльності клієнтів. У 2010 році Starbucks надав власні значки Foursquare людям, які зареєструвалися у різних місцях, і запропонував знижки тим, хто найчастіше реєструвався в

окремому магазині. Як загальне правило, маркетинг ігрофікації або ігровий маркетинг зазвичай підпадає під чотири основні категорії:

1. Брендіфікація (внутрішньоігрова реклама): повідомлення, зображення чи відео, що рекламують бренд, товар чи послугу в рамках візуальних компонентів гри. За даними NBCNews, творці ігор Electronic Arts використовували "Madden 09" та "Burnout Paradise" для просування рекламних щитів, що заохочують гравців голосувати.

2. Transmedia: результат взяття медіа-власності та її поширення на інший носій як для рекламних цілей, так і для монетизації. Класичним прикладом є "007: GoldenEye" від Nintendo (рис 2.10.). Відеогра, створена для реклами спочатку названого фільму. Зрештою, рекламна гра принесла більше грошей, ніж спочатку названий фільм.

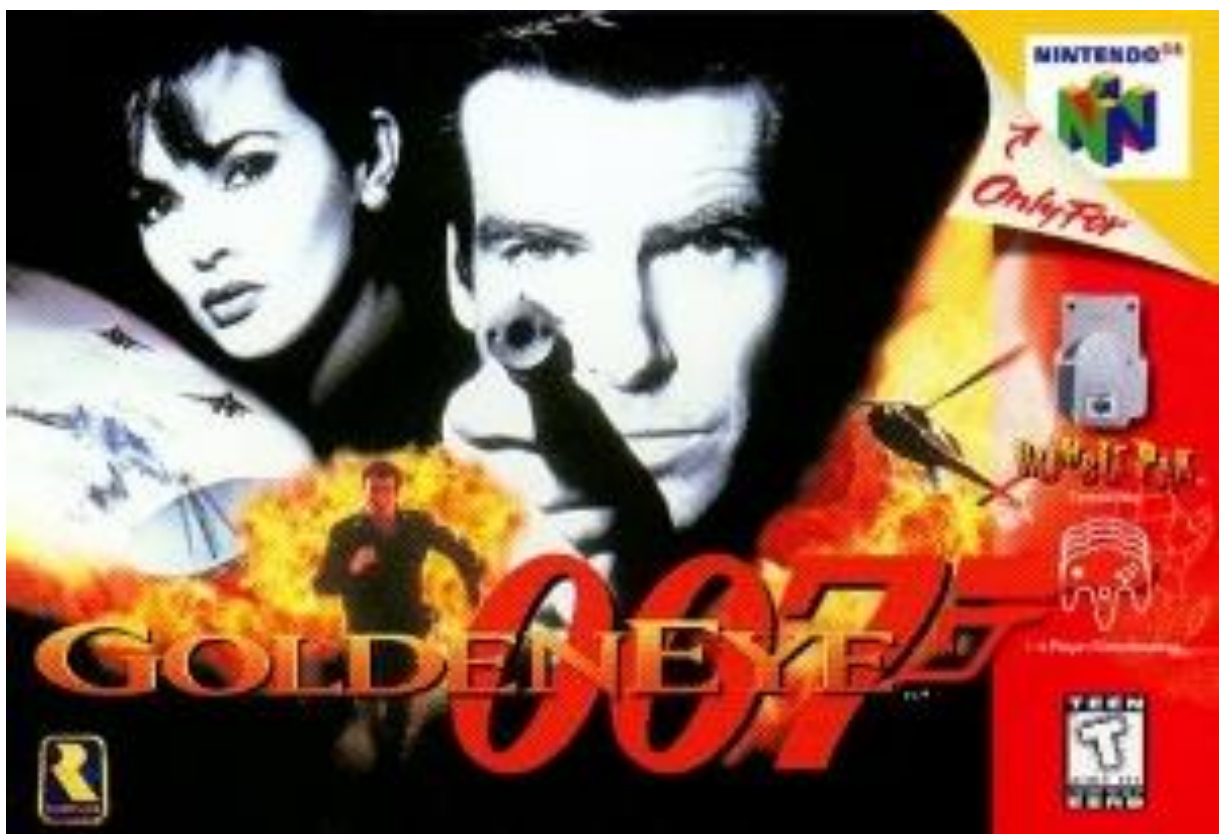


Рис. 2.10. Постер гри «007: Golden Eye»

3. Наскрізна лінія (TTL) та нижча за лінію (BTL): текст над, збоку або нижче основного екрану гри (також відомий як iFrame), рекламні зображення або текст. Прикладом цього може бути "I love Bees"(Рис 2.11).

4. Рекламні ігри: Зазвичай це ігри, засновані на популярних шаблонах мобільних ігор, таких як «Candy Crush» або «Temple Run». Потім ці ігри відтворюються на платформах, таких як WIX, із програмним забезпеченням, подібним до Gamify, з метою просування брендів, продуктів та послуг. Зазвичай, щоб заохотити залучення, лояльність та освіту продуктів. Зазвичай вони включають соціальні таблиці лідерів та нагороди, які рекламуються через соціальні медіа-платформи, такі як 10 найкращих ігор Facebook.

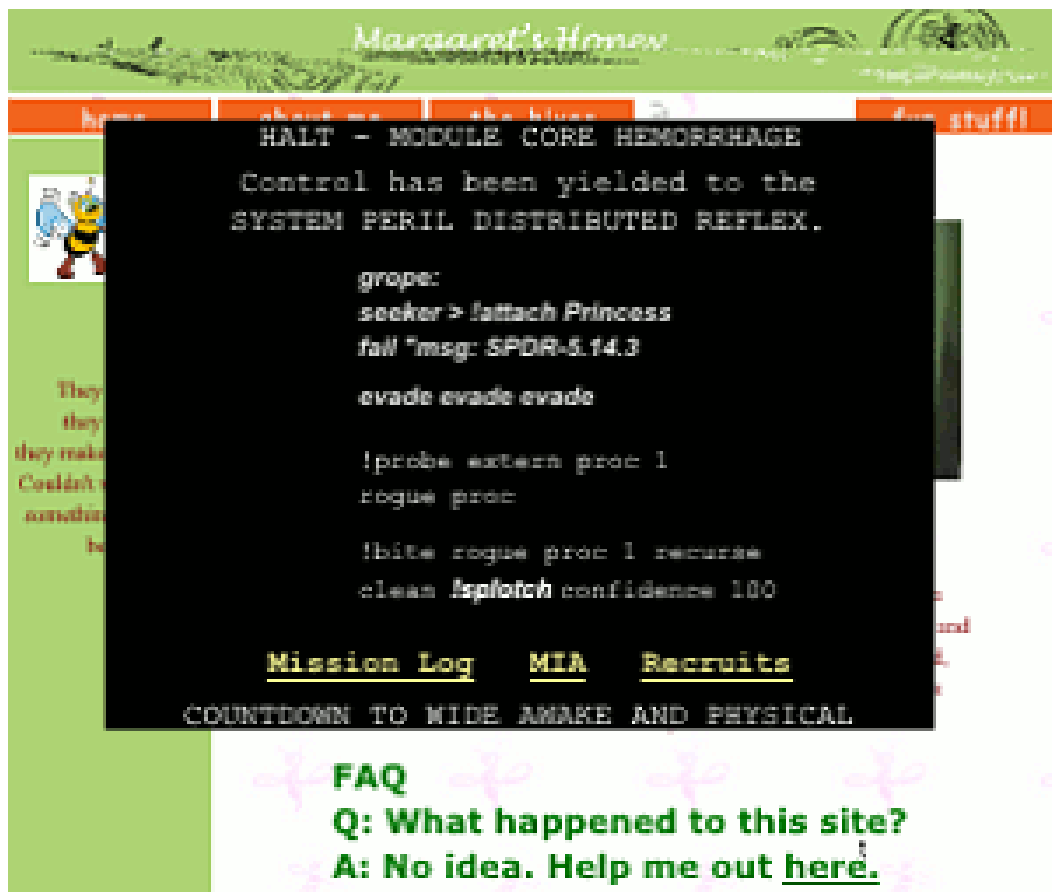


Рис. 2.11. Сайт "I love Bees"

Ігрофікація також була використана як інструмент для залучення клієнтів та для заохочення бажаної поведінки веб-сайтів[27]. Крім того, ігрофікація застосовується для збільшення залученості на сайтах, побудованих на послугах

соціальних мереж. Наприклад, у серпні 2010 року розробник веб-сайтів DevHub оголосив про збільшення кількості користувачів, які виконали свої онлайн-завдання з 10% до 80% після додавання елементів гейміфікації. На веб-сайті програмування запитань-відповідей Stack Overflow користувачі отримують бали та / або значки за виконання різноманітних дій, включаючи поширення посилань на запитання та відповіді через Facebook та Twitter. Доступна велика кількість різних значків, і коли бали репутації користувача перевищують різні пороги, користувач отримує додаткові привілеї, включаючи привілеї модератора.

Ігрофікація може бути використана для ідеації (структурований мозковий штурм для створення нових ідей). Дослідження в MIT Sloan показало, що ігри на ідеї допомогли учасникам генерувати більше і кращих ідей, і порівняло це з оцінкою впливу академічних робіт за кількістю цитат, отриманих у подальших дослідженнях[28].

Такі програми, як Fitocracy (Рис. 2.12.) та QUENTIQ (Dacadoo), використовують гейміфікацію, щоб заохотити своїх користувачів до більш ефективних фізичних вправ та покращення загального стану здоров'я. Користувачі отримують різну кількість балів за діяльність, яку вони виконують у своїх тренуваннях, і отримують рівні на основі зібраних балів. Користувачі також можуть виконувати квести (набори пов'язаних видів діяльності) та отримувати значки досягнень для досягнення фітнесу. Місяць здоров'я додає аспекти соціальних ігор, дозволяючи успішним користувачам відновлювати бали тим користувачам, які не досягли певних цілей. Дослідники громадського здоров'я вивчали використання ігрофікації для самоконтролю хронічних захворювань та загальних психічних розладів, профілактики ПСШ, та профілактики та контролю інфекцій.

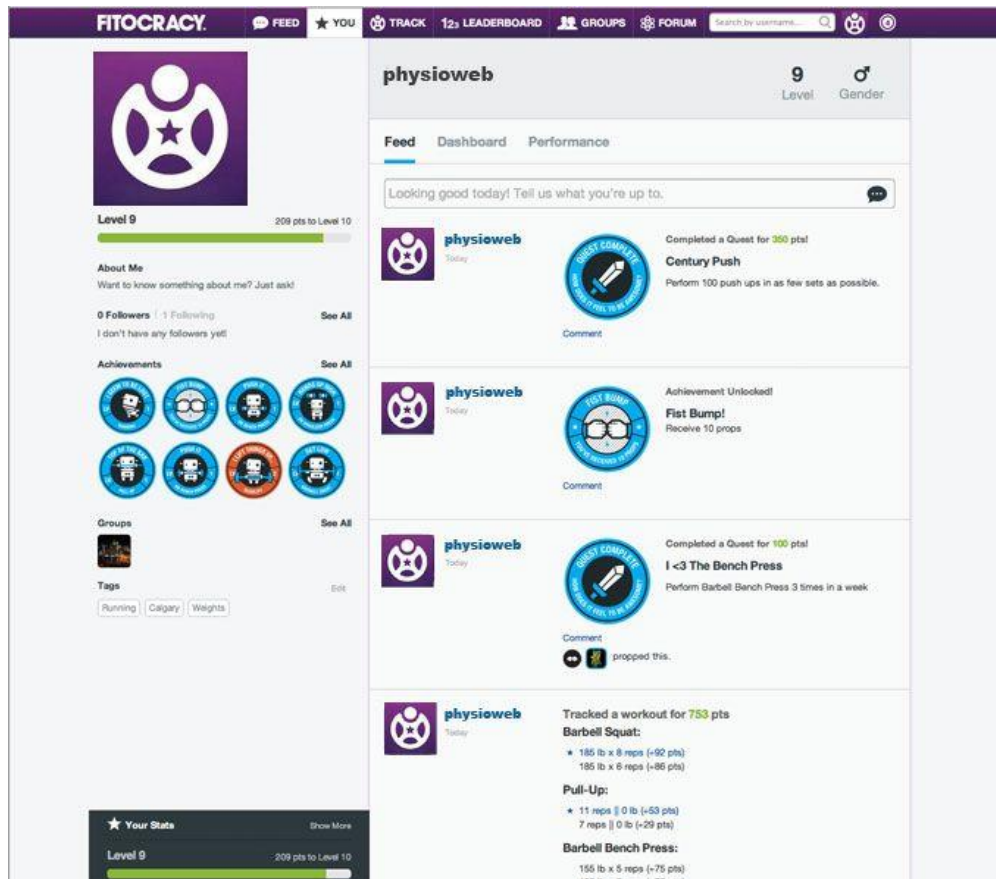


Рис. 2.12. Мобільний додаток Fitocracy

В огляді програм охорони здоров'я в Apple App Store 2014 року понад 100 додатків показали позитивну кореляцію між використовуваними елементами гейміфікації та високими рейтингами користувачів. MyFitnessPal було названо додатком, який використовував більшість елементів ігрофікації[29].

Рецензенти популярної локаційної гри Pokémon Go високо оцінили гру за сприяння фізичним вправам. Террі Шварц (IGN) сказала, що це «таємно найкраща програма для вправ», і що це змінило її повсякденну прогулянку[30]. Патрік Аллен (Лайфхакер) написав статтю з порадами щодо того, як потренуватися за допомогою Pokémon Go. Джулія Беллуз (Vox) заявила, що це може бути «найбільшою ненавмисною примхою для здоров'я коли-небудь», написавши, що одним із результатів гри, якого розробники, можливо, не уявляли, є те, що «це, здається, змушує людей рухатися»[31].

Одне з досліджень показало, що користувачі робили додаткові 194 кроки на день, коли вони починали користуватися програмою, приблизно на 26%

більше, ніж зазвичай[32]. Ingress – це подібна гра, яка також вимагає від гравця фізичної активності.

«Zombies, run» гра, в якій гравець намагається пережити зомбі-апокаліпсис через низку місій, вимагає від гравця (фізичного) бігу, збору предметів, щоб допомогти місту вижити, і прослуховування різних аудіо-розповідей, щоб розкрити таємниці.

Гейміфікація була використана для спроби підвищити продуктивність праці працівників у галузі охорони здоров'я, фінансових послуг, транспорту, уряду та інших. Загалом, ігрофікація підприємств відноситься до робочих ситуацій, коли ігрове мислення та інструменти, засновані на іграх, використовуються стратегічно для інтеграції з існуючими бізнес-процесами або інформаційними системами. І ці методи використовуються для сприяння досягненню позитивних результатів для працівників та організації[33].

Освіта та навчання – це сфери, де також спостерігався інтерес до гейміфікації. Гейміфікація привертає увагу в контексті освіти та навчання, оскільки вона пропонує різноманітні переваги, пов'язані з результатами навчання[34]. Гельмс[35] розробив таксономію навчальних ігрових елементів на основі восьми широких категорій та 26 унікальних ігрових елементів. Автори пов'язували кожен ігровий елемент із позитивними та негативними навчальними результатами.

Корпорація Майкрософт випустила гру Ribbon Hero 2 як доповнення до пакета продуктивності Office, щоб допомогти навчити людей ефективно використовувати її. Цей проект був описаний Microsoft як один з найпопулярніших проектів, який коли-небудь випускав його підрозділ Office Labs. Департамент освіти Нью-Йорка за фінансування Фонду Макарура та Фонду Білла та Мелінди Гейтс створив школу під назвою Quest to Learn Center RD навколо ігрового навчання, прагнучи зробити освіту більш привабливою та актуальною для сучасних дітей[36]. SAP використовував ігри для навчання своїх співробітників щодо стійкості. Американські військові та Unilever також використовували гейміфікацію під час навчання. Академія Khan є прикладом

використання прийомів ігрофікацію в Інтернет-освіті. У серпні 2009 року Гбанга запустив освітню гру Gbanga ZooH для зоопарку в Цюріху, яка пропонувала учасникам активно рятувати тварин, що перебувають під загрозою зникнення, і фізично повернути їх у зоопарк[37]. Гравці підтримували віртуальні місця проживання через кантон Цюріх для залучення та збору видів тварин, що перебувають під загрозою зникнення. Існує певне свідчення того, що гейміфікація може бути особливо мотиваційною для студентів з дислексією в освітніх ситуаціях.

Також зростає інтерес до використання гейміфікації в науках та освіті в галузі охорони здоров'я, де інтерактивні опитування, пригодницькі ігри та інші способи можуть застосовуватися для посилення взаємодії та інтерактивності із змістом курсу. У 2016 році Соуза гейміфікував 100-студентський вступний курс програмної інженерії програмного забезпечення у Федеральному університеті Мінас-Жерайс (Бразилія), представивши значки та таблиці лідерів[38].

Ігрофікація використовується в корпоративному навчанні, щоб мотивувати співробітників застосовувати те, що вони засвоїли на навчанні, до своєї роботи, теоретично це повинно покращити ефективність роботи. Згідно з дослідженням, проведеним Badgeville, 78% працівників використовують мотивацію, засновану на іграх, на роботі, і майже 91% заявляють, що ці системи покращують свій досвід роботи за рахунок підвищення залученості, обізнаності та продуктивності[39].

Традиційно дослідники думали, що спонукання до використання комп'ютерних систем переважно керуються зовнішніми цілями[40]. Однак у багатьох сучасних системах їх використання зумовлене, головним чином, внутрішньою мотивацією. Приклади таких систем, що використовуються в основному для виконання власних спонукань користувачів, включають онлайн-ігри, віртуальні світи, інтернет-магазини, навчання, інтернет-знайомства, сховища цифрової музики, соціальні мережі тощо. Такі системи є чудовими кандидатами для подальшої «гейміфікації» у своєму дизайні. Більше того, навіть традиційні управлінські інформаційні системи (наприклад, ERP, CRM)

піддаються «гейміфікації», так що все частіше слід враховувати як зовнішні, так і внутрішні мотивації.

Як ілюстрацію, Microsoft оголосила про плани використовувати прийоми ігрових процесів для проектування операційної системи Windows Phone 7. Хоча підприємства стикаються з проблемами створення мотиваційних ігрових стратегій[41], ключовим питанням є те, що робить ефективну гейміфікацію.

Одним з важливих типів технологічного проектування в гейміфікації є дизайн, орієнтований на гравця. На основі методології проектування, орієнтованої на користувача, його основна мета полягає у сприянні більшій взаємозв'язку та позитивній зміні поведінки між технологічними споживачами. Він має п'ять кроків, які допомагають користувачам комп'ютерів зв'язуватися з іншими людьми в Інтернеті, щоб допомогти їм досягти цілей та інших завдань, які їм потрібно виконати.

П'ять кроків:

1. особа чи компанія повинна знати свого гравця (цільову аудиторію);
2. визначити свою місію (свою мету);
3. зрозуміти мотивацію людини (особистість, бажання та активатори цільової аудиторії);
4. застосувати механіку (бали, значки, таблиці лідерів тощо);
5. керувати, контролювати та вимірювати спосіб використання їх механіки, щоб переконатися, що це допомагає їм досягти бажаного результату своєї мети, а також, що їх мета конкретна і реалістична.

2.4. Переваги ігрофікації

Ініціативи ігрофікації в контексті навчання визнають, що велика кількість дітей шкільного віку грає у відеоігри, що формує їхню ідентичність як людей та учнів. Поки світ ігор раніше були сильно перекошені до гравців чоловічої статі, останні статистичні дані показують, що трохи більше половини гравців відеоігор - це чоловіки: у США 59% чоловіків, 41% жінок та 52% чоловіків,

48% жінок у Канаді. В рамках ігор та інших цифрових медіа студенти відчують можливості для самостійності, компетентності та спорідненості, і ці можливості є тим, чого вони очікували від таких середовищ. Забезпечення цих самих можливостей у класі - це спосіб визнати реальність учнів та визнати, що ця реальність впливає на те, ким вони є. Включення елементів з ігор у сценарії класів - це спосіб надати студентам можливість діяти самостійно, проявляти компетентність та вчитися у стосунках з іншими. Ігрові елементи - це звична мова, якою розмовляють діти, і додатковий канал, за допомогою якого вчителі можуть спілкуватися зі своїми учнями.

Ігровий дизайнер Джейн Макгонігал характеризує гравців відеоігор як невідкладних оптимістів, які є частиною соціальної тканини, займаються блаженною продуктивністю та шукають епічного значення. Якщо вчителі можуть успішно організувати свої аудиторії та навчальну програму, щоб включити елементи ігор, що сприяють такій впевненості, меті та інтегрованому почуттю місії, студенти можуть захопитися навчанням та співпрацею так, що вони не захочуть зупинятися. Динамічне поєднання внутрішніх та зовнішніх мотиваторів є потужною силою, яка, якщо освітній контекст може адаптуватися до відеоігор, може збільшити мотивацію студентів та навчання учнів.

Деякі з потенційних переваг успішних ініціатив ігрофікації в навчанні включають:

1. надання студентам права власності на їх навчання;
2. можливості для ідентичності шляхом прийняття альтернативних «я»;
3. свобода невдач і спробувати ще раз без негативних наслідків;
4. шанси збільшити задоволення та радість у класі;
5. можливості диференційованого навчання;
6. зробити навчання видимим;
7. забезпечення керованого набору підзадач та завдань надихаючи студентів відкрити внутрішні мотиватори для навчання;
8. мотивація студентів з дислексією з низьким рівнем мотивації.

Говорячи про те, як відеоігри створюють для гравців дедалі складніші завдання, дизайнер ігор Емі Джо Кім припустила, що кожен навчальний сценарій може бути створений таким чином. Ця ігрова механіка, яка передбачає відстеження навчання гравців у грі та відповідь, підвищуючи рівень складності завдань у потрібний момент, утримує гравців від зайвого розчарування надто складними завданнями, а також не дає гравцям нудьгувати занадто легкі завдання. Цей темп сприяє постійному залученню та зацікавленню, що може означати, що учні зосереджені на навчальних завданнях, можуть потрапити в стан потоку або глибоко захоплені навчанням.

У великому систематичному огляді літератури щодо застосування гейміфікації у вищих навчальних закладах було визначено такі позитивні ефекти як залучення студентів, старання, результативність та задоволення, хоча вони опосередковані контекстом та дизайном.

2.5. Критика ігрофікації

Дослідник Гамбурзького університету Себастьян Детердінг охарактеризував початкові популярні стратегії ігрофікації як не веселі та створюють штучне відчуття досягнень. Він також каже, що ігрофікація може спонукати до ненавмисної поведінки[14].

У огляді 132 найпопулярніших програм для здоров'я та фітнесу в магазині додатків Apple у 2014 році, використовуючи гейміфікацію як метод модифікації поведінки, автори дійшли висновку, що «незважаючи на включення принаймні деяких компонентів гейміфікації, середні бали інтеграція компонентів гейміфікації все ще була нижчою за 50 відсотків. Це також стосувалося включення ігрових елементів та використання конструкцій теорії поведінки у стані здоров'я, таким чином, демонструючи відсутність дотримання будь-якого чіткого галузевого стандарту ефективних ігор, гейміфікації чи теорії поведінки в галузі охорони здоров'я і програми для фітнесу»[29].

Стурбованість також була висловлена в дослідженні 2016 року, в якому аналізувалися результати результатів від 1298 користувачів, які змагались у гейміфікованих та стимульованих завданнях фізичних вправ під час носіння пристроїв, що носяться. У цьому дослідженні автори припустили, що дані можуть бути сильно перекошені когортами вже здорових користувачів, а не передбачуваною аудиторією учасників, які потребують поведінкового втручання[42].

Ігрові дизайнери, такі як Джон Радофф і Маргарет Робертсон, також критикували гейміфікацію як виключення таких елементів, як розповідь історій та досвід, та використання простих систем винагород замість справжньої ігрової механіки.

Практикуючі гейміфікатори зазначають, що, хоча початкові популярні конструкції насправді в основному поклалися на спрощений підхід до винагороди, навіть ті, що призвели до значних поліпшень у короткостроковій взаємодії. Це було підтверджено першим всебічним дослідженням у 2014 році, яке прийшло до висновку, що збільшення елементів гейміфікації корелювало зі збільшенням балів мотивації, але не з оцінкою здатності чи можливості / тригера.

Це ж дослідження закликала до стандартизації в галузі додатків щодо принципів гейміфікації для підвищення ефективності програм охорони здоров'я на результати здоров'я користувачів.

Професор Массачусетського технологічного інституту Кевін Славін описав бізнес-дослідження гейміфікації як помилкові та оманливі для тих, хто не знайомий з іграми. Хізер Чаплін, пишучи «Шифер», описує гейміфікацію як «нібито популістську ідею, яка насправді приносить користь корпоративним інтересам над інтересами простих людей». Джейн Макгонігал дистанціювала свою роботу від ярлика "гейміфікація", перелічуючи винагороди за межами ігрового процесу як центральну ідею ігрофікації та виділяючи ігрові додатки, де сам геймплей є винагородою під терміном "ігровий дизайн".

"Ігрофікація" як термін також зазнала критики. Ян Богост назвав цей термін маркетинговою примхою і запропонував "експлуатаційний посуд" як більш підходящу назву для ігор, що використовуються в маркетингу. Інші думки щодо термінологічної критики дають підстави вважати, що термін ігрофікація має сенс.

Гейміфікація навчання зазнала критики за використання зовнішніх мотиваторів, яких, на думку деяких викладачів, слід уникати, оскільки вони можуть зменшити внутрішню мотивацію до навчання. Ця ідея базується на дослідженнях, які вперше з'явилися на початку 1970-х, а нещодавно були популяризовані Даніелем Пінком. Вчителі можуть не визнати, що зовнішні мотиватори вже працюють у типовому класі, або вони можуть побажати мінімізувати зовнішню мотивацію.

Деякі вчителі можуть критикувати ігрофікацію за те, що вона не так серйозно підходить до освіти. Це може бути результатом історичної різниці між роботою та грою, що продовжує уявлення про те, що клас не може бути місцем для ігор чи місцем для розваг. Ігри також страждають від помилкових уявлень про те, що вони легкі, не мають значення для навчання і застосовуються лише для дуже маленьких дітей. Ці негативні враження від гри можуть призвести до підозр щодо цінності ігрових елементів, які сприяють розвазі та відчуттю грайливості в контексті навчання.

Вчителі, які критикують гейміфікацію навчання, можуть відчувати, що їм не варто витрачати час на реалізацію ігрових ініціатив, або тому, що вони самі потягнуті з кількістю обов'язків, які вони вже мають, або тому, що вони побоюються, що навчальна програма може не бути охопленим, якщо витрачається будь-який час, присвячений чомусь іншому, крім участі у цій навчальній програмі. Ігрофікацію навчання також критикують як неефективну для певних учнів та для певних ситуацій. Прихильники гейміфікації ніколи не стверджували, що гейміфікація є такою панацеєю, визнаючи, що не є відповідною стратегією мотивування кожного учня за будь-яких обставин.

Теоретик відеоігор Ян Богост критикував ігрофікацію за її тенденцію до спрощення, маніпулятивного підходу, який не відображає справжню якість складних мотиваційних ігор. Освітні сценарії, які претендують на ігрофікацію, але використовують лише такі механізми прогресу, як бали, значки та таблиці лідерів, особливо сприйнятливі до такої критики.

Крім усіх цих тенденцій, ігрофікація не є безпрограшною комбінацією і може призвести до деяких побічних ефектів. Поширена критика породила новий вираз, названий "pointification" (або "badgefication"), що описує акт створення винагород як бали чи досягнення (значки) без стратегії чи справжнього значення для гравців.

Хороший приклад – соціальний медіа-блог Tumblr (Рис.2.13.), який намагався залучити своїх користувачів, винагороджуючи тих, хто публікує більше за допомогою балів та таблиць лідерів у системі під назвою "Tumblarity". Ця ігрофікаційна система призводить до збільшення безглузвих блогів користувачами, щоб заробити очки, і змушення користувачів жорстко критикувати процес ігрофікації. Через кілька місяців компанія вирішила закрити ініціативу.

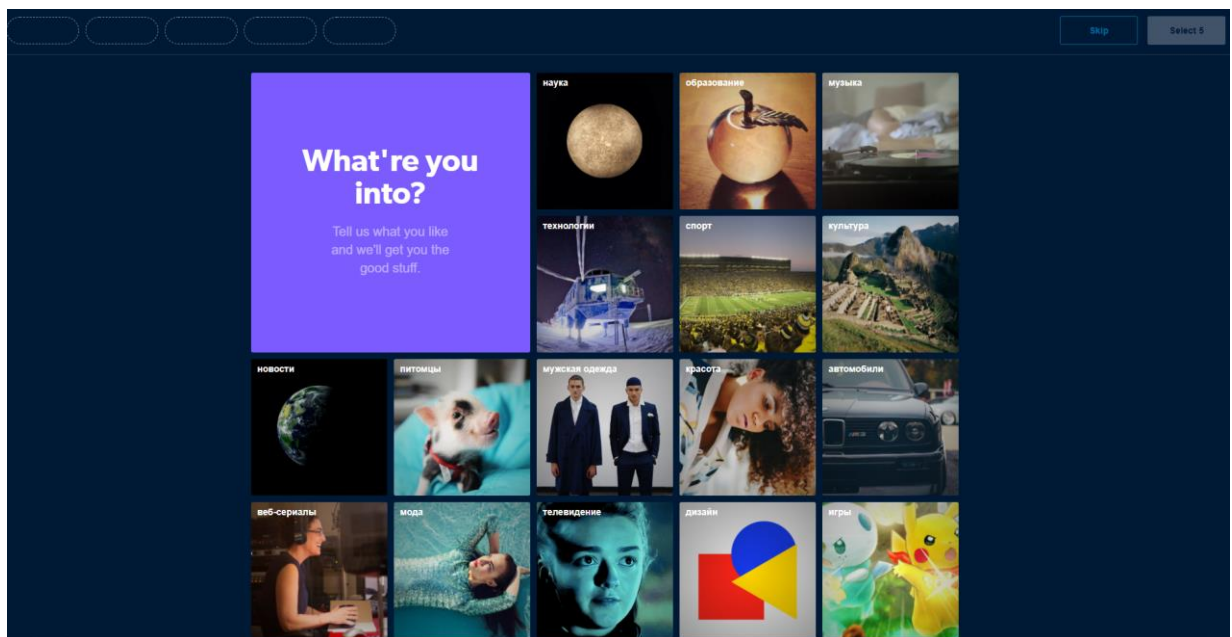


Рис. 2.13. Головна сторінка медіа-блогу Tumblr

2.6. Висновки до другого розділу

В останній час все більше і більше світових компаній починає використовувати багато елементів ігрофікації в різних сферах. Ігри існують з часів виникнення нашої цивілізації. Навіть відеоігри вже існують понад 40 років і представляють індустрію, яка приносить більше 70 мільярдів доларів прибутків на рік. Мільйони людей у всіх куточках Землі проводять по 2-3 години в день за іграми на консолі, персональних ком'ютерах та мобільних телефонах. Ігри популярні серед всіх груп населення незалежно від віку, статі чи соціального положення.

Серед елементів ігрофікації ми виділили наступні:

1. Бали;
2. Значки;
3. Таблиці лідерів;
4. Графіки ефективності;
5. Сюжет;
6. Зворотній відлік;
7. Аватари;
8. Соціалізація.

Розглянули переваги та недоліки ігрофікації та існуючі приклади. Визначили для себе, які саме елементи ігрофікації слід застосувати в розроблюваній системі.

РОЗДІЛ 3

РОЗРОБКА АДАПТИВНОЇ СИСТЕМИ ВИВЧЕННЯ ПРАВИЛ ДОРОЖНЬОГО РУХУ З ЕЛЕМЕНТАМИ ІГРОФІКАЦІЇ

3.1. Блок-схема алгоритму

Для розробленого алгоритму була створена блок-схема (Рис. 3.1):

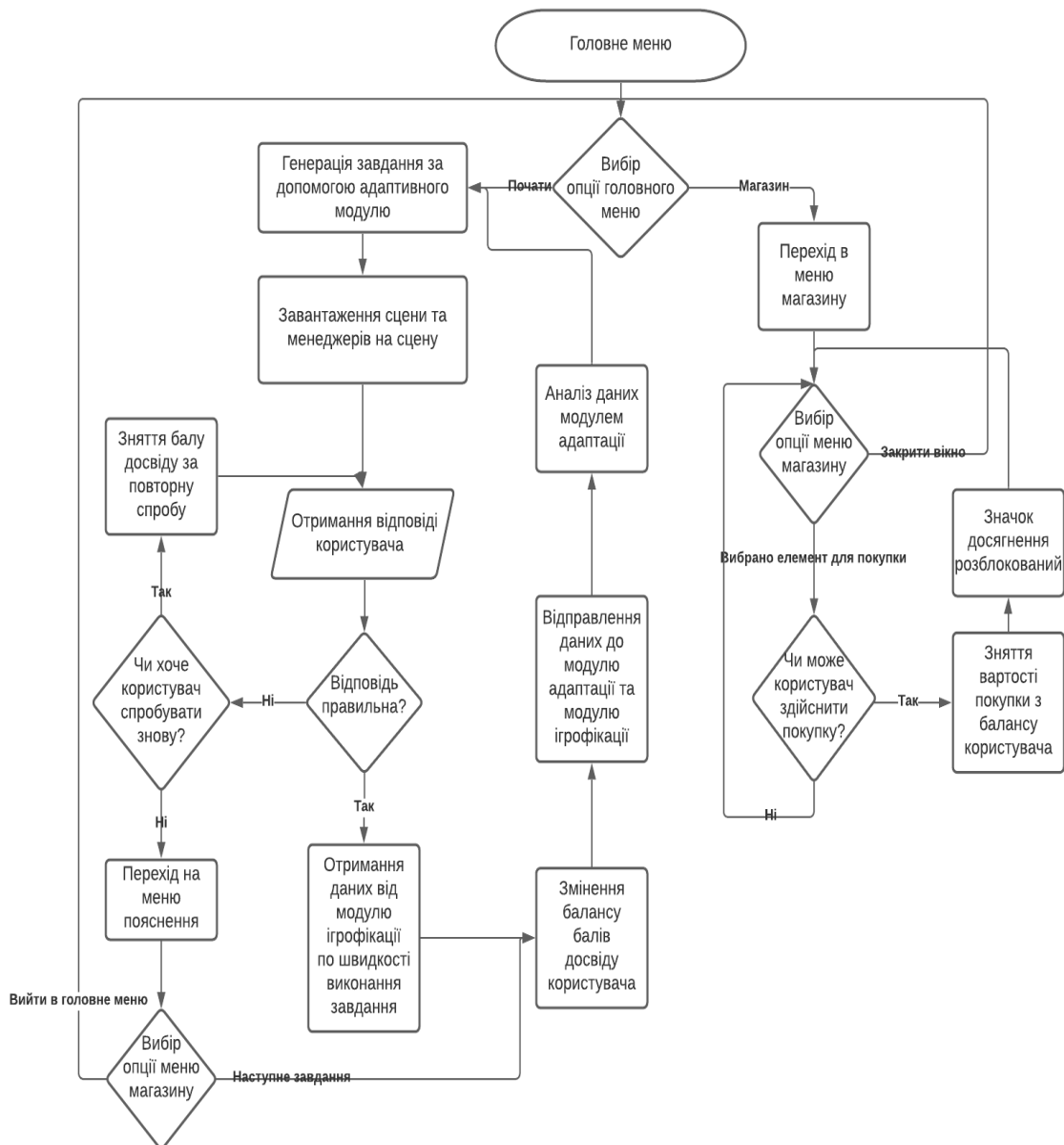


Рис 3.1. Блок-схема алгоритму програми

3.2. Структура інформаційної системи вивчення правил дорожнього руху

Програма складається з п'яти компонентів:

1. Візуалізатор;
2. Адаптивний модуль;
3. Контролюючий модуль;
4. Ігрофікаційний модуль;
5. Модуль зацікавлення.

Після запуску програми, користувача вітає головне меню, з двома опціями (Рис 3.2).

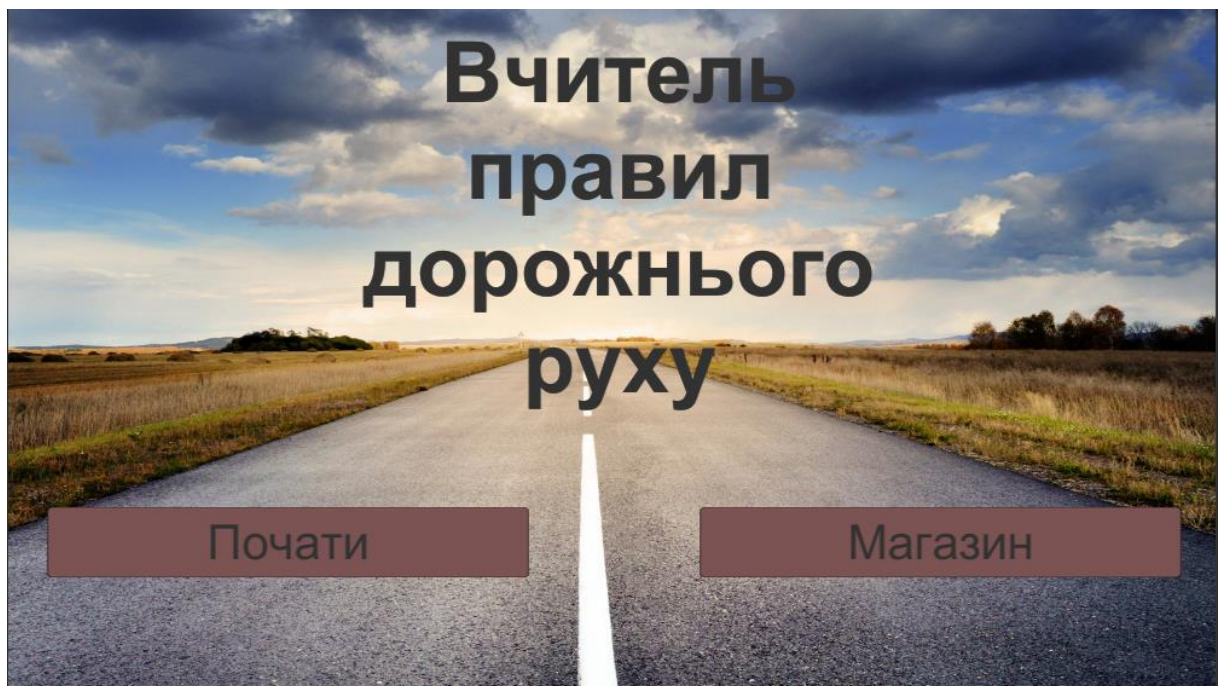


Рис. 3.2. Головне меню програми

Якщо натиснути кнопку «Магазин» (Рис 3.3.), то програма покаже меню вибору різних косметичних елементів за бали досвіду, які він отримує за кожен правильно вирішене завдання.



Рис.3.3. Меню вибору різних косметичних елементів за бали досвіду

Після натискання на кнопки «Почати», система генерує завдання для користувача на основі попередніх відповідей або ж випадкове завдання із переліку тих, які є в програмі, якщо програма запускається вперше.

Якщо користувач вирішує завдання правильно за певний час, то система винагороджує його балами досвіду, генерує наступне завдання та аналізуючи його правильну відповідь складає модель користувача. Якщо відповідь неправильна, то користувачу показується вікно з вибором (Рис 3.4.).

При виборі «Спробувати ще раз» система повторить завдання, проте без зворотного відліку. За кожну спробу користувач втрачає 1 бал досвіду. Якщо вибрати «Показати відповідь», програма покаже вікно з поясненням до певного завдання. Якщо користувач не встигає вирішити завдання за певний відрізок часу, то бали досвіду він не отримує. В середньому на вирішення завдань дається від 10 до 30 секунд в залежності від складності та теми задачі.

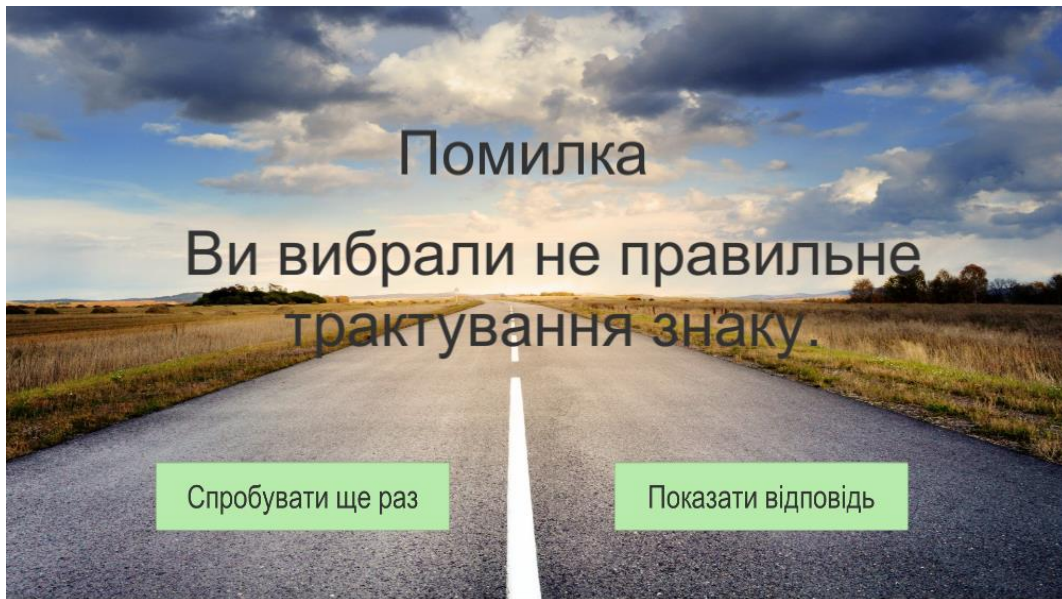


Рис 3.4. Меню вибору після неправильної відповіді

3.2.1. Модуль візуалізації задач правил дорожнього руху

За модуль візуалізації загалом відповідає головний модуль контролюючий хід програми. Після натискання на кнопку «Почати» модуль вибирає з раніше сформованих рівнів, той який адаптивна система вважає найбільш інформативним та цікавим для користувача та передасть модулю ігрофікації данні про те скільки секунд треба виділити користувачу на вирішення завдання та скільки балів досвіду користувач отримає.

Сцени складаються з наступних елементів:

1. Інтерфейс користувача з показом необхідної інформації;
2. Дорожня ситуація або текстова задача, яка буде сформована згідно вхідних даних від головного модулю;
3. Текстури необхідні для формування задачі.
4. Модуль ігрофікації;
5. Адаптивний модуль.

Після відповіді користувача йому буде показано або роз'яснення або кількість отриманого досвіду та в цей же час буде почата предзавантаження наступної задачі. Тож користувач не буде довго чекати, щоб продовжити

процес. В цей же час модуль, який вимірює зацікавленість користувачів записує інформацію для аналізу

За допомогою ігрового рушія Unity було створено 5 сцен (Рис. 3.5.), які будуть завантажені в разі необхідності.

Name	Date modified	Type	Size
Знаки дорожнього руху.unity	05/12/2020 06:11 PM	Unity scene file	6 KB
Знаки дорожнього руху.unity.meta	05/12/2020 06:11 PM	META File	1 KB
Обгін.unity	05/12/2020 06:13 PM	Unity scene file	6 KB
Обгін.unity.meta	05/12/2020 06:13 PM	META File	1 KB
Перехрестя.unity	05/12/2020 06:10 PM	Unity scene file	6 KB
Перехрестя.unity.meta	05/12/2020 06:10 PM	META File	1 KB
Повороти.unity	05/12/2020 06:11 PM	Unity scene file	6 KB
Повороти.unity.meta	05/12/2020 06:11 PM	META File	1 KB
Теоретичні питання.unity	05/12/2020 06:11 PM	Unity scene file	6 KB
Теоретичні питання.unity.meta	05/12/2020 06:11 PM	META File	1 KB

Рис. 3.5. Сформовані сцени для вибору завдання

3.2.2. Адаптивний модуль

Цей модуль має доступ до всієї класифікації завдань правил дорожнього руху та відповідає за створення моделі користувача. Адаптивний модуль збирає данні про:

1. Час виконання завдання;
2. Правильність виконання;
3. Складність завдання;
4. Тему завдання;

Чим більше людина буде помилятися в завданнях однієї і тієї ж теми, тим більше завдань такого типу адаптивний модуль буде показувати користувачу. Якщо людина завдання виконує правильно, то адаптивний модуль буде зменшувати час на виконання завдання до мінімально можливого. Складність

завдань завжди буде підвищуватися. Також модуль відповідає за надання інформації про задачі модулю ігрофікації.

3.2.3. Модуль ігрофікаційних елементів

Для цієї програми зі всіх елементів ігрофікації я вибрав:

1. Бали (Рис 3.6.);



Рис.3.6. Бали досвіду в грі

Бали спонукають користувача виконувати завдання правильно та за певний час. Для балів ми створили магазин, щоб користувач міг бачити свій прогрес відкриваючи більше і більше елементів змінення зовнішнього вигляду машин.

2. Зворотній відлік (Рис 3.7.);

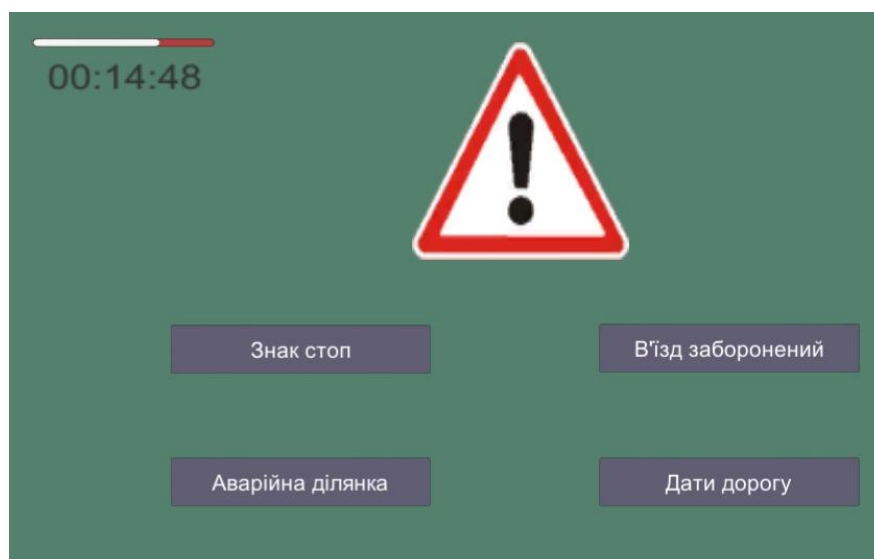


Рис 3.7. Зворотній відлік на рівні з вибором правильної трактовки знаку

Зворотній відлік робить гру напруженішою та цікавішою за допомогою обмеження в часі.

3. Досягнення (Рис 3.8.).



Рис. 3.8. Нові текстури машин, як вид досягнень в грі

Досягнення роблять прогрес більш наочним та змушують користувача заходити частіше, щоб отримати певне досягнення. За допомогою досягнень можна спонукати користувача до певної потрібної нам поведінки.

3.3. Заміри модулю зацікавленості користувача

Під час виконання дипломної роботи ми провели експеримент на не адаптивній системі без елементів ігрофікації. Групі користувачів А, ми встановили цю програму без модулю адаптивності та ігрофікації. Групі користувачів В ми встановили адаптивну систему. Групі користувачів С ми встановили повну версію з елементами ігрофікації. Після тижня з модулю, який вимірює зацікавленість користувачів ми зібрали данні та представили наступним чином (Рис 3.9.):

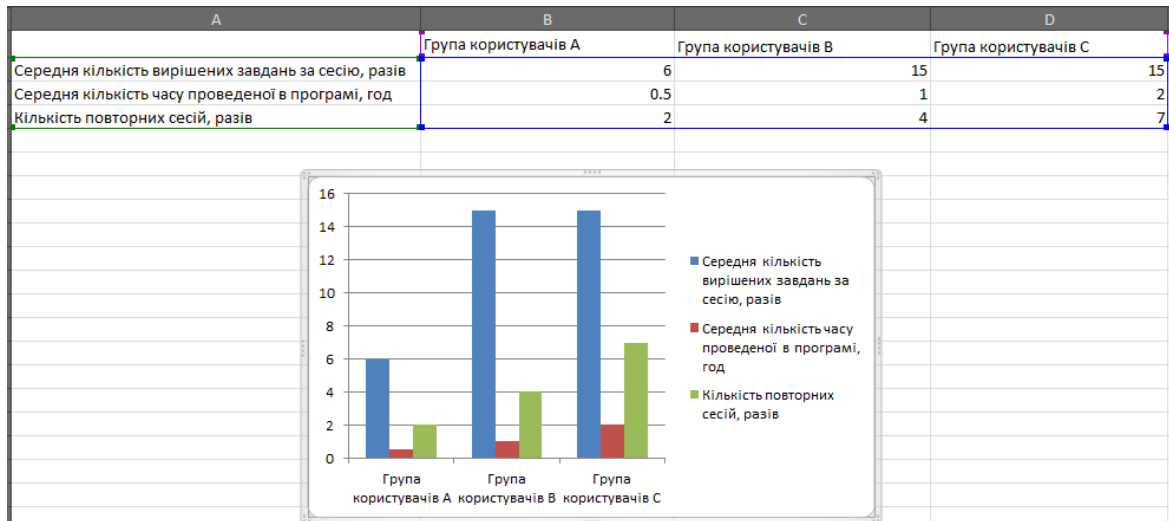


Рис 3.9. Таблиця даних модулю зацікавленості користувача

З цих даних можна зробити наступний висновок:

1. В адаптивних системах кількість вирішених завдань за сесію значно більше ніж в не адаптивних системах;
2. Модуль ігрофікації значно поліпшив кількість повторних сесій, що доволі важливо для нашої системи;
3. Кількість часу показує, що чим більше система адаптується тим більше часу користувач готовий провести в програмі;
4. Елементи ігрофікації вплинули лише на кількість повторних сесій та кількість часу, але на кількості вирішення завдань за сесію не вплинули.

3.4. Заміри прогресу вивчення правил дорожнього руху

Для цього експерименту групі користувачів А, ми надали цю програму для користування на 1 день. Групі користувачів В, ми надали цю програму для користування на 4 дні. Групі користувачів С, ми надали цю програму на 1 тиждень. Кожна група складалася з 10 чоловік. Результати експерименту наступні (Рис. 3.10.):

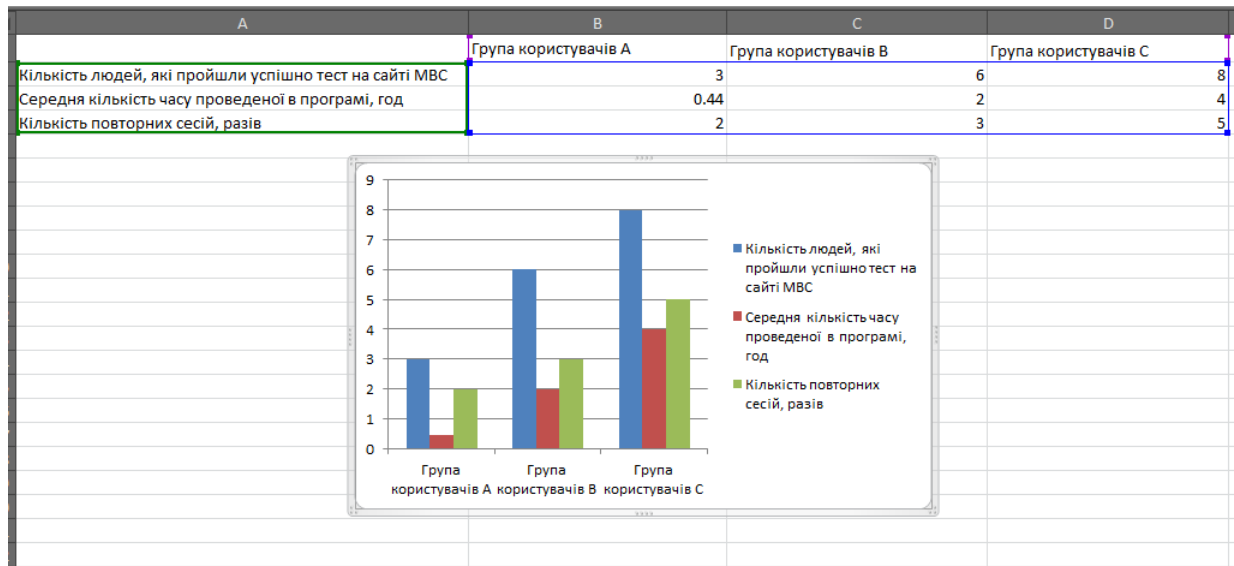


Рис 3.10. Результати експерименту

З цих даних можна зробити наступний висновок:

1. Користувачі у яких було більше всього часу для користування програми склали тести найкраще;
2. Чим більше було часу проведено в програмі тим краще результати на тесті;
3. З плином часу інтерес до програми не впав.

3.5. Висновки до третього розділу

Для даного ПЗ ми вирішили використовувати бали досвіду, значки досягнень та зворотній відлік, адже саме ці елементи можуть не лише мотивувати користувача, а й показувати прогрес не тільки для користувача, а й для адаптаційного модулю.

Програма складається з п'яти компонентів:

1. Візуалізатор;
2. Адаптивний модуль;
3. Контролюючий модуль;
4. Ігрофікаційний модуль;
5. Модуль зацікавлення.

Було проведено два дослідження в яких перевірялась ефективність даного ПЗ в аспектах мотивації та якості знань.

Зі зростаючими можливостями ігрофікації інтерес до неї в навчальному контексті тільки посилюється. В даний час додатків, які мають дійсно чітку та налагоджену ігрофікаційну систему в споживчому сегменті не так багато, проте для великих компаній ця тема стає все більш актуальною та обов'язково знайде свого користувача та замовника.

.

РОЗДІЛ 4 ЕКОНОМІКА

4.1. Розрахунок вартості програмного забезпечення

Одним з головних етапів при розробці ПЗ є визначення трудомісткості та розрахунок витрат на створення програмного продукту. Одним з головних етапів при розробці ПЗ є визначення трудомісткості та розрахунок витрат на створення програмного продукту.

Початкові дані:

- Годинна заробітна плата програміста, грн /год - 200
- Вартість машино-години ЕОМ, грн/год – 40

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_0 + t_u + t_a + t_{\Pi} + t_{\text{відл}} + t_d,$$

де

t_0 - витрати праці на підготовку і опис поставленого завдання (приймається 50);

t_u – витрати праці на дослідження алгоритму вирішення задач;

t_a – витрати праці на розробку блок-схем алгоритму;

t_{Π} – витрати праці на програмування за готовою блок-схемою;

$t_{\text{відл}}$ – витрати праці на відладку програм на ПЕОМ;

t_d – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q * C * (1 + p) ,$$

де

q – Передбачуване число операторів;

C – Коефіцієнт складності програми;

p – Коефіцієнт кореляції програми в ході її розробки.

Наприклад, q буде дорівнювати 1250, C – 1,5, p – 0,2.

$$Q = 1250 * 1,5 * (1 + 0,2) = 2250 \text{ людино-годин,}$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{QB}{(75..85)*k} ,$$

де

B – Коефіцієнт збільшення витрат праці (внаслідок неповного опису завдання, $B = 1,1 \dots 1,4$);

k – Коефіцієнт кваліфікації, обумовлений від стажу роботи з даної спеціальності (якщо стаж роботи менший 2 років, то $k = 1,2$).

$$t_u = 2250 * 1,2 / (80 * 1,2) = 28,1 \text{ людино-годин,}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \dots 25)k} \text{ людино-годин,}$$

$$t_a = \frac{2250}{23 * 1,2} = 81,5 \text{ людино-годин,}$$

Витрати на складання програми по готовій блок-схемі:

$$t_{\Pi} = \frac{Q}{(20...25)k}, \text{ люДИНО-ГОДИН,}$$

$$t_{\Pi} = \frac{2250}{23*1,2} = 81,5 \text{ люДИНО-ГОДИН,}$$

Витрати праці на налагодження програми на ЕОМ:

$$t_{\text{вїдл}} = \frac{Q}{(4..5)k}, \text{ люДИНО-ГОДИН,}$$

$$t_{\text{вїдл}} = \frac{2250}{4*1,2} = 468,8, \text{ люДИНО-ГОДИН,}$$

Витрати праці на підготовку документації:

$$t_{\text{д}} = t_{\text{др}} + t_{\text{до}}, \text{ люДИНО-ГОДИН,}$$

де $t_{\text{др}}$ – трудомісткість підготовки матеріалів і рукопису.

$$t_{\text{др}} = \frac{Q}{(15..20)k}, \text{ люДИНО-ГОДИН,}$$

$$t_{\text{др}} = \frac{2250}{17*1,2} = 110,3, \text{ люДИНО-ГОДИН,}$$

$t_{\text{до}}$ – трудомісткість редагування, печатки й оформлення документації

$$t_{\text{до}} = 0,75 * t_{\text{др}}, \text{ люДИНО-ГОДИН,}$$

$$t_{\text{до}} = 0,75 * 110,3 = 82,7, \text{ людино-годин,}$$

$$t_{\text{д}} = 110,3 + 82,7 = 193, \text{ людино-годин,}$$

Отримуємо трудомісткість розробки програмного забезпечення:

$$t = 50 + 28,1 + 81,5 + 81,5 + 468,8 + 193 = 902,8 \text{ людино-годин.}$$

4.2. Витрати на створення програмного забезпечення

Витрати на створення програмного забезпечення $K_{\text{пз}}$ включають витратина заробітну плату розробників програми ($Z_{\text{зп}}$) і витрати машинного часу, необхідного для налагодження програми на ЕОМ ($Z_{\text{мв}}$).

$$K_{\text{пз}} = Z_{\text{зп}} + Z_{\text{мв}}, \text{ грн,}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{\text{зп}} = t * C_{\text{пр}}, \text{ грн,}$$

де:

t - загальна трудомісткість, людино-годин;

$C_{\text{пр}}$ – середня годинна заробітна плата програміста, грн/година.

$$Z_{\text{зп}} = 902,8 * 200 = 180560 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{\text{МВ}} = t_{\text{в\ddot{и}дл}} * C_{\text{мч}}, \text{ грн,}$$

де:

$t_{\text{в\ddot{и}дл}}$ - трудомісткість налагодження програми на ЕОМ;

$C_{\text{мч}}$ - вартість машино-години ЕОМ.

$$Z_{\text{МВ}} = 468,8 * 40 = 18752 \text{ грн.}$$

Таким чином витрати на створення програмного забезпечення, складуть:

$$K_{\text{пз}} = 180560 + 18752 = 199312 \text{ грн.}$$

Очікувана тривалість розробки ПО:

$$T = \frac{t}{B_k * F_p}$$

де:

B_k - число розробників;

F_p – місячний фонд робочого часу(при 40-ка годинному робочому тижні

$F_p = 176$ годин).

$$T = 902,8 / 1 * 176 = 5 \text{ місяців.}$$

4.3. Маркетингові дослідження ринку збуту розробленого програмного продукту

В процесі дослідження та розробки було створено адаптивну навчальну систему для особистого використання. Метою навчального асистента є покращення навчального процесу користувача та ефективний розподіл його часу на засвоєння правил дорожнього руху.

Принцип взаємодії користувача з системою наступний:

1. Користувач проходить завдання підібрані модулем адаптації.
2. Користувач отримує інформацію при неправильній відповіді.
3. Користувач отримує бали досвіду.
4. Користувач може отримати досягнення всередині інформаційної технології.
5. Коли користувач виконує завдання, то система аналізує яке завдання та за який час він виконав.

Як результат, навчальна технологія зменшує час, що витрачається користувачем на вибір завдань та планування навчання. Таким чином, технологія пришвидшує та покращує навчальний процес користувача.

Навчальний тренажер було розроблено у вигляді гри на Unity, тож він може розповсюджуватися у вигляді комп'ютерного чи мобільного додатку, що встановлюється на пристрій. Головна платформа – це мобільні додатки. Проте головний плюсом є легкий перехід між платформами інструментами Unity. Систему можна монетизувати системою «Google Ads».

Даний тренажер може зацікавити широку аудиторію, але перш за все він орієнтований на користувачів автомобілів та пішоходів.

Цей проект досить специфічний, але він може знайти свою аудиторію. Одним з ефективним способів залучення аудиторії є написання статті на IT ресурсах як Habrhabr або Troger, така практика широко використовується навіть великими компаніями, які хочуть продемонструвати свій продукт. У цій статті буде розкриватись процес розробки системи та використані у ньому методи адаптації, це не тільки безкоштовно, але разом з тим дозволить нам набрати зацікавлену аудиторію, яка буде слідкувати за розвитком проекту, також ми отримаємо корисний досвід та перші думки й коментарі щодо продукту

Можливими конкурентами навчальної технології є ITS (Intelligent tutoring system – інтелектуальні системи навчання) такі як Knewton та Lalilo. У навчального асистента та ITS схожа мета – покращення та персоналізація процесу навчання, за винятком того, що ITS мають більш широкий функціонал

та націлені на застосування у навчальних установах, а не для особистого користування. Сучасних ITS систем небагато й вартість їх рішень неможливо оцінити, так як ціна пропозиції формується окремо для кожного випадку. У нашому дослідженні ми не порівнюємо інформаційну технологію з іншими продуктами, так як прямих конкурентів нема, а ITS не є прямими конкурентами й ціна їх невідома.

4.4. Економічна ефективність

Як було зазначено вище, у ході дослідження була створена навчальна адаптивна технологія як додаток для особистого використання, тож дане програмне забезпечення не було призначено для впровадження на підприємстві. Отже, ми не можемо обчислити економічну ефективність цього продукту, але ми можемо визначити його соціальний ефект.

Соціальним ефектом від нашого продукту є покращення та збалансування процесу навчання (зокрема самонавчання) людини правилам дорожнього руху за рахунок того, що технологія сама вирішує які завдання та за який час потрібно пройти користувачу згідно його потреб та пріоритетів.

Окрім цього, напрацювання та технології, використані у цьому продукті, можуть бути повторно застосовані в інших адаптивних освітніх системах.

ВИСНОВКИ

У процесі дослідження були отримані наступні результати:

При вивченні принципів побудови адаптивних систем було розглянуто архітектуру, підходи та форми адаптивних систем.

Загалом адаптивні системи складаються з:

1. Медіа модуль;
2. Експертний модуль;
3. Адаптаційний модуль;
4. Модель користувача;

Існує 3 головних підходи до побудови адаптивних систем:

1. Макроадаптивний;
2. Підхід АТІ;
3. Мікроадаптивний.

З форм адаптивних систем виділяють:

1. Адаптивна презентація;
2. Адаптивна навігація.

Для визначення оптимального інструменту візуалізації завдань було розглянуто 3 основні безкоштовні ігрові рушії та обрано Unity.

В роботі було запропоновано впровадити в систему вивчення правил дорожнього руху елементи ігрофікації. Це було зроблено для підвищення мотивації користувачів.

Серед елементів ігрофікації ми визначили головних 6:

1. Бали;
2. Значки досягнень;
3. Таблиці лідерів;
4. Графіки ефективності;
5. Зворотній відлік;
6. Соціалізація.

Для даного ПЗ ми вирішили використовувати бали досвіду, значки досягнень та зворотній відлік, адже саме ці елементи можуть не лише мотивувати користувача, а й показувати прогрес не тільки для користувача, а й для адаптаційного модулю.

Програма складається з п'яти компонентів:

1. Візуалізатор;
2. Адаптивний модуль;
3. Контролюючий модуль;
4. Ігрофікаційний модуль;
5. Модуль зацікавлення.

Було проведено два дослідження в яких перевірялась ефективність даного ПЗ в аспектах мотивації та якості знань.

Ми вважаємо, що узагальнення отриманого практичного досвіду буде корисно для більш глибокого розуміння суті ігрофікації та побудови адаптивних систем, їх необхідності і можливостей, а також полегшить перехід до вирішення складніших завдань. На наш погляд, проведене дослідження буде корисно для науковців, розробників, аспірантів і студентів, що спеціалізуються або цікавляться проблематикою ігрофікації та адаптивних навчальних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Heinze N., Nejd W., A Logical Characterization of Adaptive Educational Hypermedia. *New Review of Hypermedia and Multimedia (NRHM)*. – 2004. - № 10(1). - P. 77-113.
2. Brusilovsky, P. Adaptive and intelligent Web-based educational systems / P. Brusilovsky, C. Peylo // *International Journal of Artificial Intelligence in Education. Special Issue on Adaptive and Intelligent Web-based Educational Systems* – 2003. – № 13 (2-4). – P. 159-172.
3. Brusilovsky, P. Methods and techniques of adaptive hypermedia / P. Brusilovsky // *User Modeling and User-Adapted Interaction*. – 1996. – № 6 (2-3). – P. 87-129.
4. Fisher G., *User Modeling in Human-Computer Interaction. Contribution to the 10th Anniversary Issue of the Journal. User Modeling and User-Adapted Interaction (UMUAI)*. - 2000
5. Keller F. S., *Good-Bye, Teacher....* № 1(1). - 1968. - P. 79–89.
6. Beldagli B. Illustrating an ideal adaptive e-learning: A conceptual framework/ B. Beldagli, T. Adiguzel//. - №2(2). - 2010. - P. 5755-5761.
7. Cronbach L. J. The two disciplines of scientific psychology. – 1957. - №12 (11). – P. 671-684.
8. Lockee B. B. *Handbook of reseach on educational communications and technology/ B. B. Lockee, M. B. Larson, J.K. Burton, D. M. Moore//*. – 2008. – P. 187-197.
9. Choi S.Y. An adaptive system supporting collaborative learning based on a location-based social network and semantic user modeling/S. Y. Choi, J. M. Kang//. – 2012.
10. Park O. *Handbook of reseach on educational communications and technology/ O. Park, J. Lee//*. – 2007. – P.634-664.

11. Kickmeier-Rust M. D. Micro-adaptivity: Protecting immersion in didactically adaptive digital educational games/ M. D. Kickmeier-Rust, D. Albert//. – 2010. – P. 95-104.
12. Huotari K. Defining Gamification – A Service Marketing Perspective/ K. Huotari, J. Hamari. – 2012. – P. 17-20.
13. Hamari J. Does Gamification Work? – A Literature Review of Empirical Studies on Gamification/J. Hamari, J. Koivisto, H. Sarsa//. – 2014. – P. 3025-3034.
14. Deterding S. Just Add Points? What UX Can (and Cannot) Learn From Games. – 2010. – P. 13-84.
15. Nelson M. Soviet and American precursors to the gamification of work. – 2012. – P. 22-36.
16. Fuchs M. Ludic interfaces. Driver and product of gamification. – 2012. – P. 19-25.
17. Herger M. Gamification and Law or How to stay out of Prison despite Gamification. – 2012. – P. 1-4.
18. European Central Bank Virtual Currency Schemes. – 2012. – P. 14-42.
19. Sailer M. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction/M. Sailer, J. U. Hense//. – 2017. – P. 371-378.
20. Zichermann G. Gamification by Design: Implementing game mechanics in web and mobile apps/ G. Zichermann, C. Cunningham//. – 2011. – P. 14-52.
21. Werbach K. The gamification toolkit – dynamics, mechanics, and components for the win/K. Werbach, D. Hunter//. – 2015. – P. 1-47.
22. Costa J. P. Time's up: Studying Leadboards For Engaging Punctual Behaviour/J.P. Costa, R. R. Wehbe, J. Robb, L. E. Nacke//. – 2013. – P. 26-33.
23. Landers R. N. An Empirical Test of the Theory of Gamified Learning: The Effect of Leaderboards on Time-on-Task and Academic Performance/R. N. Landers, A. K. Landers//.– 2014. – P. 770-782.
24. Kapp K. M. The gamification of learning and instruction: Game-based methods and strategies for training and education. – 2012. – P. 25-51.

25. Reiners T. Gamification in education and business/T. Reiners, L. C. Wood//. – 2014. – P. 220-446.
26. Van Grove J. Gamification: How Competition Is Reinventing Business, Marketing & Everydy Life. – 2011. – P. 1-3.
27. O'Brien C. Get ready for the decade of gamification. – 2010. – P.13-25.
28. Toubia O. Idea Generation, Creativity, and Inceptives. – 2006. – 12-36.
29. Lister C. Just a Fad? Gamification in Health and Fitness Aps/C. Lister, J.H. West, B. Cannon, T. Sax//. – 2014. – P. 2-12
30. Schwartz T. Pokemon Go is Secretly the Best Exercise App out there. – 2016. – P. 1-5.
31. Belluz J. Pokemon Go may be the greatest unintentional health fad ever. – 2016. – P. 1-7.
32. McFarland M. Pokemon Go could add 2.83 million years to uses' lives. – 2016. – P. 1-2.
33. Ruhi U. Level Up Your Strategy: Towards a Descriptive Framework for Meaningful Enterprise Gamification. – 2015. – P. 5-15.
34. Osatuyi B. Systematic Review of Gamification in IS Education: A Multi-Method Approach. Communications of the Association for Information Systems/B. Osatuyi, T. Osatuyi, Ramiro de la Rosa//. – 2018. – P. 14-32.
35. Helms R. W. A Method for the Design of Gamified Trainings/R.W. Helms, R. Barneveld, F. Dalpiaz. – 2015. – P. 33-44.
36. Hulling R. Gamification: Turning Work Into Play. – 2010. – P.5-14.
37. Gooch D. Using Gamification to Motivate Students with Dyslexia/D. Gooch, A. Vasalou, L. Benton, R. Khaled//. – 2016. – P. 2-13.
38. Souza M. Gamification in Softwaare Engineering Education: An Empirical Study/ M. Souza, K. Constantino, L. Veado, E. Figueiredo//. – 2017. – P.276-284.
39. Penenberg A. The Gamification of Corporate Training. – 2017. – P.1-3.

40. Lowry B. Taking fun and games seriously: Proposing the hedonic-motivation system adoption model/B.Lowly, J. Gaskin, N. Twyman, B. Hammer, T. Roberts//. – 2013. – P. 617-671.
41. Kappen D. The Kaleidoscope of Effective Gamification: Deconstructing Gamification in Business Applications/ D. Kappen, E. Nacke//. – 2013. – P. 119-122.
42. Van Mierlo T. Behavioral Economics, Wearable Devices, and Cooperative Games: Results from a Population-Based Intervention to Increase Physical Activity/T. Van Mierlo, D. Hyatt, A. T. Ching, R. Fournier, R. S. Dembo//. – 2016. – P. 8-12
43. Encarnacao L. M. Adaptivity in graphical user interfaces: An experimental framework. - Computers & Graphics 19 (6). - 1995. - P. 873-884.
44. Alrifai M. Learner Profile Management for Collaborating Adaptive eLearning Applications./Dolog P., Nejd W./APS'2006: Joint International Workshop on Adaptivity, Personalization and the Semantic Web at the 17th ACM Hypertext'06 conference, August 2006, Odense, Denmark/ACM Press.
45. Allen R. User Models: Theory, Method, and Practice. International Journal of Man-Machine Studies. - №32. – 1990. – P. 511-543.
46. Beaumont I. H. User modeling in the interactive anatomy tutoring system ANATOM-TUTOR. In User Models and User Adapted Interaction. – 1994. – №4(1). – P. 21–45
47. Gaudioso E. Towards web-based adaptive learning communities/ E. Gaudioso, J. G. Boticario//. – 2006. – P. 2-10.
48. Yee N. Motivations for play in online games. – 2006. – P. 772–775.
49. Deterding S. From game design elements to gamefulness: defining gamification/ S. Deterding, D. Dixon, R. Khaled, L. Nacke// Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. – 2011. – P. 9-15
50. Maciuszek D. Component-based development of educational games: The case of the user interface/ D. Maciuczek, G. Ruddeck, A. Martens // Proceedings of

the 4th European Conference on Games-based Learning (ECGBL 2010), 2010, Reading, UK. – P. 208-217.

51. Burke B. Gamify: How Gamification Motivates People to Do Extraordinary Things/ Routledge – New York, 2014 – 192 с.

52. Hocking J. Unity in action: Multiplatform game development in C#/ Manning – New York, 2015 – 334 с.

53. Nystrom R. Game Programming Patterns/ Genever Benning. – Columbia, SC, 2018. – 332с.

54. Lavieri E. A Study of Adaptive Learning for Educational Game Design. – 2014. – 166 с.

55. Kapp K. The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education/Pfeiffer. – Washington, 2012. – 336с.

56. Behr A.L. Exploring the lecture method: An empirical study. Studies in Higher Education. – 1988. – 189-200 с.

57. Lukosek G. Learning C# by Developing Games with Unity 5.x/ G. Lukosek, T. Norton//. – 2016. – 230с.

58. Martin R. C. Agile Principles, Patterns, and Practices in C#/ R. C. Martin, M. Martin//. – 2006. – 732с.

59. Albahari J. C# 7.0 in a Nutshell/J.Albahari, B. Albahari//. – 2017. – 1070с.

60. Бойко В.В. Экономика предприятий Украины. Основной курс: Учебник для вузов. – Д.: Пороги, 1997. – 312 с.

61. Андрусенко Г.О. Основы маркетинга. – Е.: НМК ВО, 1992. – 143 с.

62. Баркан Д.И. Маркетинг для всех. – Л.: «Культинформпрес», 1991. – 256 с.

63. Бусыгин А.В. Предпринимательство. Основной курс: Учебник для вузов. – М.: ИНФРА-М, 1997. – 608 с.

64. Завялов П.С., Демидов В.Е. Формула успеха – маркетинг. – М.: Междунар. Отношения, 1991.

65. Котлер Ф. Основы маркетинга: Пер. с англ. / Общ. Ред. и вступ. ст. Е.М. Пеньковой – М.: Прогресс, 1990. – 636с.
66. Скворцов Н.Н. Бизнес-план предприятия. – К.: Вища школа, 1995. – 187 с.
67. Мескон М.Х., Альберт М., Хедоури Ф. Основы менеджмента. – М.: Дело, 1992. - 702 с.
68. Мете А.Ф., Штец К.А., Бельгольский Б.П. и др. Организация и планирование предприятий. – М.: Металлургия, 1986. – 560 с.
69. Основы инновационного менеджмента: Теория и практика: Учеб. пособие / Под ред. П.Н. Завлина и др. – М: ОАО Издательство «Экономика». 2000. – 475 с.
70. Савчук В.П., Прилипко С.И., Величко Е.Г. Анализ и разработка инвестиционных проектов. - Учебное пособие. - Киев: Абсолют-В. Эльга. 1999. - 304 с.

ЛІСТИНГ ПРОГРАМИ**Ігровіфікаційний модуль: GamificationManager.cs**

```
using UnityEngine;

public class GamificationManager : Singleton<GamificationManager>
{
    [SerializeField] private int experienceAmount;
    public int ExperienceAmount => experienceAmount;

    private float _maxCountdown;

    private float _countDownCurrent = 0f;

    public float MaxCountdown => _maxCountdown;

    public float CountdownCurrent => _countDownCurrent;

    private void Update()
    {
        if (GameManager.Instance.CurrentGameState == GameState.play)
        {
            if (_countDownCurrent > 0f)
            {
                _countDownCurrent -= Time.deltaTime;
            }
            else
            {
                AdaptiveManager.Instance.TimeExpired();
            }
        }
    }

    public void AddAmount(int amount)
    {
        experienceAmount += amount;
    }

    public void ChargeAmount(int amount)
    {
        if (experienceAmount - amount >= 0)
        {
            experienceAmount -= amount;
        }
        else
        {
            experienceAmount = 0;
        }
    }

    public bool IsAffordable(int amount) => experienceAmount - amount >= 0;

    public void GetCountdownSettings(float time)
    {
        _maxCountdown = time;
    }
}
```

```

        _countDownCurrent = time;
    }
}

```

Адаптивный модуль: AdaptiveManager.cs

```

using System.Collections.Generic;
using Random = UnityEngine.Random;

public class AdaptiveManager : Singleton<AdaptiveManager>
{
    public Dictionary<AvailableTasks, PDDTask[]> availableTasks;

    private Dictionary<AvailableTasks, bool[]> doneTasks;

    private bool _isTimeExpired;

    public void DonePDDTask(PDDTask task, bool isFail)
    {
        AddTask(task.currentKind, isFail, _isTimeExpired);
    }

    private void Start()
    {
        _isTimeExpired = false;
    }

    public void TimeExpired()
    {
        _isTimeExpired = true;
    }

    public PDDTask GetNextTask()
    {
        AvailableTasks next = RecommendedKind();

        return availableTasks[next][Random.Range(0,
availableTasks[next].Length)];
    }

    public void AddTask(AvailableTasks kind, bool isFail, bool
isTimeExpired)
    {
        doneTasks.Add(kind, new []{isFail, isTimeExpired});
    }

    public AvailableTasks RecommendedKind()
    {
        if (!doneTasks.ContainsKey(AvailableTasks.crossroads))
        {
            return AvailableTasks.crossroads;
        }
        if (!doneTasks.ContainsKey(AvailableTasks.turn))
        {
            return AvailableTasks.turn;
        }
        if (!doneTasks.ContainsKey(AvailableTasks.rush))
        {
            return AvailableTasks.rush;
        }
        if (!doneTasks.ContainsKey(AvailableTasks.theoretical))

```

```

    {
        return AvailableTasks.theoretical;
    }
    if (!doneTasks.ContainsKey(AvailableTasks.sings))
    {
        return AvailableTasks.sings;
    }

    AvailableTasks currentWorstTask = AvailableTasks.theoretical;

    foreach (var currentKind in doneTasks.Keys)
    {
        int WorstResults = 0;
        int taskFailed = 0;
        int taskExpired = 0;
        foreach (var values in doneTasks.Values)
        {
            if (values[0]) taskFailed++;
            if (values[1]) taskExpired++;
        }
        if (taskExpired + taskFailed > WorstResults)
        {
            WorstResults = taskExpired + taskFailed;
            currentWorstTask = currentKind;
        }
    }

    return currentWorstTask;
}
}

```

Контролирующий модуль: GameManager.cs

```

using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public enum GameState
{
    play,
    pause
}

public class GameManager : Singleton<GameManager>
{
    private GameState _currentGameState;

    private bool isTimeExpired = false;

    private PDDTask _currentTask;

    private bool _isFail;

    [SerializeField] private HUDController hud;

    [SerializeField] private GameObject TryAgainWindow;

    [SerializeField] private GameObject ExplanationWindow;

```

```

public GameState CurrentGameState
{
    get => _currentGameState;
    set => _currentGameState = value;
}

private int numberCarToSpawn;

// Start is called before the first frame update
void Start()
{
    _currentGameState = GameState.pause;
}

public void CarChosen(Car chosen, CarSpawning parent)
{
    parent.CarChosen(chosen);
    if (chosen.CanMove)
    {
        DonePDDTask();
        _currentGameState = GameState.pause;
        _currentTask = null;
        StartLevels();
    }
    else
    {
        _currentGameState = GameState.pause;
        hud.ShowTryAgainWindow(TryAgainWindow);
    }
}

public void GameStart(GameObject menuToClose)
{
    menuToClose.SetActive(false);
    _currentGameState = GameState.play;
}

public void StartLevels()
{
    _currentTask = AdaptiveManager.Instance.GetNextTask();
    SceneManager.LoadScene((int) _currentTask.currentKind);

    if (_currentTask.currentKind == AvailableTasks.sings)
    {
        var sign = new SignTask();
        sign.GetQuestion(_currentTask.question.GetComponent<Sprite>());
        sign.GetRightAnswer(_currentTask.rightAnswer);
        HashSet<int> answerNumbers = new HashSet<int>();
        answerNumbers.Add(Random.Range(0,
            _currentTask.wrongAnswers.Length));
        answerNumbers.Add(Random.Range(0,
            _currentTask.wrongAnswers.Length));
        answerNumbers.Add(Random.Range(0,
            _currentTask.wrongAnswers.Length));
        var arrayNumbers = answerNumbers.ToArray();

        sign.FillNextAnswer(_currentTask.wrongAnswers[arrayNumbers[0]]);
        sign.FillNextAnswer(_currentTask.wrongAnswers[arrayNumbers[1]]);
        sign.FillNextAnswer(_currentTask.wrongAnswers[arrayNumbers[2]]);
    }
}

```



```

        if (_currentTask.currentKind == AvailableTasks.theoretical)
        {
            var theoretical = new TheoreticalTask();

            theoretical.GetQuestion(_currentTask.question.GetComponent<Text>());
            theoretical.GetRightAnswer(_currentTask.rightAnswer);
            HashSet<int> answerNumbers = new HashSet<int>();
            answerNumbers.Add(Random.Range(0,
            _currentTask.wrongAnswers.Length));
            answerNumbers.Add(Random.Range(0,
            _currentTask.wrongAnswers.Length));
            answerNumbers.Add(Random.Range(0,
            _currentTask.wrongAnswers.Length));
            var arrayNumbers = answerNumbers.ToArray();

            theoretical.FillNextAnswer(_currentTask.wrongAnswers[arrayNumbers[0]]);
            theoretical.FillNextAnswer(_currentTask.wrongAnswers[arrayNumbers[1]]);
            theoretical.FillNextAnswer(_currentTask.wrongAnswers[arrayNumbers[2]]);
        }

        if (_currentTask.currentKind == AvailableTasks.turn)
        {
            var turn = new TurnTask();
            turn.CarToSpawn = _currentTask.CarToSpawn;
        }

        if (_currentTask.currentKind == AvailableTasks.turn)
        {
            var crossroads = new TurnTask();
            crossroads.CarToSpawn = _currentTask.CarToSpawn;
        }

        if (_currentTask.currentKind == AvailableTasks.rush)
        {
            var rush = new RushTask();
            rush.CarToSpawn = _currentTask.CarToSpawn;
        }
    }

    public void DonePDDTask()
    {
        AdaptiveManager.Instance.DonePDDTask(_currentTask, _isFail);
    }

    public void DoneCountdown()
    {
        isTimeExpired = true;
    }

    public void TryAgain()
    {
        if(GamificationManager.Instance.IsAffordable(1))
        {
            GamificationManager.Instance.ChargeAmount(1);
            hud.HideCurrentWindow();
            CurrentGameState = GameState.play;
        }
    }

    public void NextTask()
    {

```

```

        _currentTask = null;
        StartLevels();
    }

    public void ToMainMenu()
    {
        SceneManager.LoadScene("Menu");
    }
}

```

Car.cs:

```

using System;
using UnityEngine;
using UnityEngine.AI;

public class Car : MonoBehaviour
{
    [SerializeField] private NavMeshAgent _navMeshAgent;

    private CarSpawning parent;

    [SerializeField] private bool _canMove;

    [SerializeField] private GameObject TurningRight;

    [SerializeField] private GameObject TurningLeft;

    public bool CanMove
    {
        get => _canMove;
        set => _canMove = value;
    }

    // Start is called before the first frame update
    void Start()
    {
        _canMove = false;
        _navMeshAgent = GetComponent<NavMeshAgent>();
        TurningRight.SetActive(false);
        TurningLeft.SetActive(false);
    }

    private void OnMouseDown()
    {
        if (_canMove)
        {
            GameManager.Instance.CarChosen(this, parent);
        }
    }

    public void IsTurning(bool isTurningLeft, bool isTurningRight)
    {
        if (isTurningLeft)
            TurningLeft.SetActive(true);
        if (isTurningRight)
            TurningRight.SetActive(true);
    }

    public void GetParentObject(CarSpawning spawningParent)
    {
        parent = spawningParent;
    }
}

```

```
public void MoveTo(Vector3 destination)
{
    _navMeshAgent.SetDestination(destination);
}

private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Finish"))
    {
        Destroy(gameObject);
    }
}
}
```

PDDTask.cs

```
public class PDDTask : MonoBehaviour
{
    public int difficulty;
    public int minTime;
    public int maxTime;

    public AvailableTasks currentKind;

    public GameObject ExplanationWindow;

    public GameObject question;

    public string rightAnswer;

    public string[] wrongAnswers;

    public int CarToSpawn;
}
```

ВІДГУК
керівника економічного розділу
на кваліфікаційну роботу магістра
на тему: «Методи, алгоритми та інформаційна технологія для
вивчення правил дорожнього руху за допомогою впровадження елементів
ігрофікації»
студента групи 122м-19-2 Сердюка Владислава Юрійовича

Керівник економічного розділу
доцент каф. ПЕП та ПУ, к.е.н.

Л. В. Касьяненко

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файла	Опис
Пояснювальні документи	
Диплом_Сердюк.doc	Пояснювальна записка до магістерської роботи. Документ Word.
Диплом_Сердюк.pdf	Пояснювальна записка до магістерської роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація_Сердюк.ppt	Презентація до магістерської роботи