



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної підготовки

Випускна кваліфікаційна робота бакалавра

Проектування гейміфікованих освітніх процесів-головоломок (на прикладі гри Granny

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проекту

№	П.І.Б.	Група
1	Буравльова Катерина	КН-П-212
2	Старченко Тимур	КН-П-212

Автор звіту _____ Буравльова Катерина Павлівна

Науковий керівник	Доцент кафедри ІТ та ФП, к.т.н., доцент Краковецький О.Ю.
-------------------	---

АНОТАЦІЯ

УДК 004.9

Б-91

Буравльова К. П. Проектування гейміфікованих освітніх процесів-головоломок (на прикладі гри *Granny*: випускна кваліфікаційна робота на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». – Одеса: ОТУШ, 2025. – 18 с.

Робота присвячена дослідженню та створенню Імплементация засобів інтелектуального аналізу ситуацій в умовах змінного середовища на прикладі комп'ютерної гри, натхненної відомим інді-проектом *Granny*. Головна мета гри – вижити в обмеженому середовищі, уникнути переслідування ворожого NPC та розгадати серію головоломок для втечі. Атмосфера страху підтримується через звуковий супровід, візуальний дизайн і динамічну поведінку ворога.

Актуальність обраної теми зумовлена популярністю жанру horror серед гравців ПК та зростанням інтересу до інді-розробок, що створюються невеликими командами або окремими авторами. Ігри такого типу не потребують великих бюджетів, але здатні забезпечити високий рівень залучення через напругу, інтригу та іммерсивність.

Об'єктом дослідження є інтелектуальні рішення взаємодії у хорор-іграх та їх вплив на гравецький досвід. У процесі роботи було поставлено наступні завдання:

- Провести аналіз найпопулярніших комп'ютерних horror-ігор для виявлення ефективних ігрових механік, інтерфейсів та поведінки противників.
- Обрати відповідні інструменти для розробки (Unity, C#, Blender, Audacity тощо) та обґрунтувати їх використання.

- Розробити базовий прототип гри для ПК з підтримкою головних механік: стелс, переховування, взаємодія з об'єктами, відкриття дверей за допомогою ключів і кодів.
- Провести тестування працездатності гри, перевірку UI/UX, а також базову оптимізацію під різні конфігурації персональних комп'ютерів.
- Реалізувати систему поведінки AI. Демон використовує систему навігації Unity NavMesh для орієнтації у просторі.
- Реалізувати поведінку поліцейських NPC. Для поліцейських створити систему станів (State Machine), що дозволяє гнучко керувати їх поведінкою та взаємодіями в залежності від подій у грі.

У процесі підготовки звіту були використані як друковані, так і електронні джерела, а також виконано порівняльний аналіз архітектури подібних інді-проектів.

ЗМІСТ

АНОТАЦІЯ.....	2
ЗМІСТ.....	4
Вступ.....	5
ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЄКТУ.....	7
РОЗДІЛ 1.....	12
ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ.....	12
РОЗДІЛ 2.....	15
РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ.....	15
РОЗДІЛ 3.....	17
РЕАЛІЗАЦІЯ АДМІНІСТРАТИВНОЇ ЧАСТИНИ.....	17
Висновок.....	21
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	23

ВСТУП

Упродовж останніх десятиліть індустрія відеоігор активно розвивається, поступово перетворюючись на одну з найвпливовіших та найприбутковіших сфер цифрових технологій. Ігри давно перестали бути лише засобом розваги — вони стали потужним інструментом соціального, мистецького та технологічного самовираження. Особливу роль у цьому процесі відіграють інді-проекти, які, попри обмежений бюджет, здатні привернути увагу мільйонів гравців завдяки цікавим ідеям та атмосферному виконанню.

На сьогодні жанр horror survival займає окрему нішу серед геймерської спільноти, оскільки пропонує унікальний досвід – напругу, страх, необхідність швидко приймати рішення та боротися за виживання у складних умовах. Яскравим представником цього жанру стала гра Granny, яка здобула значну популярність завдяки простій реалізації, напруженому геймплею та лякаючій атмосфері. Саме цей проєкт був обраний як джерело натхнення для створення власного прототипу комп'ютерної гри.

У рамках даної випускної кваліфікаційної роботи розроблявся ігровий проєкт, що є адаптацією основних ідей Granny з урахуванням авторського бачення, сучасних вимог до інтерфейсу, геймдизайну та програмної реалізації. Особливу увагу було приділено створенню логіки штучного інтелекту супротивника, поведінці об'єктів в оточенні, гравецькій взаємодії з середовищем, системі збереження прогресу та реалізації атмосфери страху за допомогою звукового й візуального супроводу.

Предметною областю проєкту стало створення desktop-гри для персональних комп'ютерів у жанрі survival horror. Функціональні компоненти гри включають переміщення гравця, механіку переховування, пошуку ключів та інструментів, відкриття дверей, взаємодію з об'єктами та загрозу у вигляді ШІ-противника, що постійно переслідує гравця.

З технічної точки зору розробка охоплювала повний цикл створення гри: від вибору інструментів (Unity, C#, Blender тощо) і проектування логіки до реалізації, тестування та налагодження роботи на різних конфігураціях комп'ютерів. У роботі також детально розглянуто дизайн ігрових сцен, інтерфейс користувача, механізми оптимізації та базові принципи UX у

хорор-іграх.

Наступні розділи містять аналіз обраного жанру, архітектуру розробленої гри, технічне завдання, приклади реалізації ключових модулів, результати тестування та обговорення труднощів, що виникали у процесі створення гри. Також описано шляхи їх подолання, використані методи й засоби, а також надано висновки щодо досягнутих результатів.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЄКТУ

THE RUNAWAY

(за зразком гри *Granny*)

Загальний опис

Програмне забезпечення (ПЗ) призначене для створення однокористувацької комп'ютерної гри в жанрі horror escape room. Суть гри полягає у втечі персонажа із замкненого середовища (будівлі), уникаючи переслідувань з боку ворожого NPC, вирішуючи головоломки та шукаючи вихід. Гра орієнтована на психологічний тиск, гравець повинен уникати гучних звуків, взаємодіяти з об'єктами довкілля, збирати ключові предмети й знаходити шляхи для втечі.

Особливістю системи є наявність штучного інтелекту супротивника, динамічного звукового супроводу, скриптованих подій, а також системи збереження прогресу. Взаємодія між компонентами побудована за клієнт-серверним принципом (для внутрішньої модульної структури гри, не мережевої гри).

Призначення

Гра створюється як ігровий інструмент дослідження механік страху та виживання, з подальшою можливістю модифікації рівнів, моделей, сценаріїв. Проєкт дозволяє вивчити архітектуру комп'ютерних ігор, а також механізми побудови геймплею, візуального та аудіо оформлення. Система

повинна бути масштабованою, з можливістю доповнення новими рівнями, ворогами, подіями та предметами.

Вимоги до функціоналу (обов'язкова частина)

Головне меню

- Стартова сцена з вибором опцій: «Почати гру», «Налаштування», «Вийти».
- Візуалізація атмосфери гри (музика, темне освітлення, анімації).
- Обмежений функціонал для ознайомлення: перегляд трейлера/інструкції.

Гра

- Гравець переміщується від першої особи у будівлі.
- Ціль гри - знайти ключі, предмети та знайти вихід
- Ворожий NPC патрулює приміщення. Він здатен інтерактувати с предметами в будинку, вимикати світло, відкривати двері тощо.
- Інвентар гравця: збирання предметів, обмежена кількість комірок.
- Схованки: можливість переховуватись у шафах.
- Головоломки: замки, коди, предмети, які потрібно знайти.

Налаштування

- Зміна гучності, керування, графіки, мова.

Архітектура

ПЗ має бути спроектоване за модульною структурою:

- Mockup – прототип інтерфейсу та ігрових екранів з переходами між сценами (створюється в Figma або Unity UI Tools).
- Core Logic – внутрішня ігрова логіка: управління гравцем, обробка подій, поведінка NPC.
- Graphics and Audio – модулі моделей, текстур, освітлення, музики та звукових ефектів.

- Persistence – система збереження прогресу, даних про проходження.
- UI – відображення інтерфейсу гри (інвентар, меню, повідомлення).

Технології

- Unity
- C#
- Mirror Networking
- Blender (готові моделі з Asset Store або самостійне моделювання)
- Audacity для обробки звуків

Звітність

Кожен учасник створює:

- Практичну частину:
Посилання на репозиторій з розподіленням ролей (коміти, гілки), робоча збірка гри (build).
- Автореферат:
Опис своєї частини роботи, вибір рішень, реалізація, труднощі, підсумки. У документі повинні бути: структура гри, розподіл задач, використані ресурси, висновки.

РОЗДІЛ 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

Згідно з технічним завданням (ТЗ), наша команда провела аналіз можливих архітектурних рішень для програмного забезпечення. Враховуючи продуктивність, зручність для користувача, а також актуальність підходів до розробки, було прийнято рішення обрати архітектурний стиль SPA (Single Page Application) як основний для реалізації сервісу.

Архітектура програмного забезпечення розділена на чотири незалежні модулі:

- Design;
- Frontend;
- Backend;
- DevOps.

Design

Дизайн у рамках нашого проєкту відіграв фундаментальну роль і виступав не просто декоративним компонентом, а повноцінним інструментом формування гравецького досвіду. Ми розглядали дизайн не лише як набір візуальних елементів, а як цілісну систему, яка повинна була забезпечити інтуїтивне сприйняття, комфортну навігацію і створення емоційного фону гри. Збалансований дизайн дозволяє гравцеві не лише легко орієнтуватися в ігровому середовищі, а й повністю занурюватися у світ гри, отримуючи задоволення від кожної взаємодії.

Під час створення візуальної частини ми орієнтувалися на цільову аудиторію, враховуючи її очікування, візуальні уподобання та ігрову поведінку. Це дозволило правильно розставити акценти у стилістиці, побудові локацій, інтерфейсі та загальній подачі гри.

Інструменти, які ми активно використовували:

- *Adobe Illustrator* – для створення 2D-графіки, статичних елементів інтерфейсу, іконок та композицій;

- *Adobe Photoshop* – для детальної обробки текстур, художніх ілюстрацій, а також глибокої кольорової корекції;
- *Blender* – для моделювання та анімації 3D-об'єктів, що були інтегровані у віртуальне середовище;
- *Audacity / Voicemod / Vegas Pro* – для створення та обробки звукових ефектів, а також роботи з голосами персонажів.

Ключові завдання:

- *Аналіз та референси*: ми опрацювали значну кількість схожих ігор, проаналізували їх стиль, геймплей та побудову сцен, що допомогло визначити вектор власного стилю.
- *Карта сцен та ігровий флоу*: було складено послідовність переходів між екранами гри (меню, лоббі, сцена гри, програш, перемога), з чіткою логікою навігації.
- *Інтеграція UI в Unity*: усі створені макети були перенесені в ігрове середовище з використанням Canvas, TextMeshPro, UI Toolkit. Ми реалізували анімації через Animator та локалізацію через Unity Localization.
- *Презентаційні матеріали*: для фінального представлення ми змонтували геймплейні трейлери, використовуючи Vegas Pro, та оформили рекламні візуали.

Frontend

Frontend у нашому проєкті - це не просто графічний інтерфейс, а динамічна та гнучка система взаємодії користувача з ігровим світом. Саме через UI гравець отримує інформацію про поточний стан гри, виконує дії та відчуває зв'язок із подіями у віртуальному середовищі. Ми приділили велику увагу тому, щоб інтерфейс був одночасно мінімалістичним, зрозумілим і функціональним.

Важливо було досягти високого рівня зручності використання (UX) — щоб жодна дія не викликала у гравця сумнівів або непорозуміння. Для цього було впроваджено низку сучасних рішень і підходів до побудови UI.

Досягнення в реалізації фронтенду:

- *Стилістична цілісність*: усі елементи інтерфейсу розроблялися в єдиній візуальній парадигмі, з чітким використанням палітри кольорів, стилів шрифтів, форм та анімацій.
- *Адаптивність*: завдяки Canvas Scaler, Layout Groups, Anchors ми забезпечили коректне відображення UI на різних розширеннях екрана, що особливо важливо для широкого спектру моніторів.
- *Ефекти взаємодії*: додано плавні переходи, інтерактивні підсвічування, звукові сигнали при натисканні, а також анімовані індикатори завантаження чи статусу.

Backend

Backend частина була реалізована за допомогою архітектури хост-клієнт, що є оптимальним підходом для інді-гри з невеликою командою та відсутністю власного виділеного сервера. Основна логіка гри оброблялася на стороні хоста, що значно зменшувало навантаження на клієнти й полегшувало налагодження ігрових подій.

Основні елементи архітектури:

- *Хост-сервер*: один із гравців створював ігрову сесію, виступав у ролі сервера, обробляв події, рух, дії інших учасників і координував їхню синхронізацію.
- *Система лоббі*: створена на базі Steamworks - дозволяла організувати сесію, шукати доступні кімнати, передавати імена гравців та параметри гри.
- *Локальне збереження*: для збереження даних користувача, таких як мова, звук, настройки - було використано PlayerPrefs, а локалізація здійснювалася через Unity Localization Tables.
- *Обмін івентами*: події гри (відкриття дверей, використання предметів тощо) передавалися у вигляді байтових повідомлень через Steam P2P, із централізованою обробкою на хості.
- *Безпека*: автентифікація через Steam ID, контроль доступу до сесії, можливість видалення гравців з лоббі, підтримка Relay-серверів у разі проблем з NAT.

DevOps

Незважаючи на невеликий склад команди, ми впровадили низку DevOps-рішень, які дозволили зробити процес розробки прозорим, зручним та безпечним. Ми поділили завдання, організували версіювання коду, автоматизували збереження білдів, тестування і публікацію.

Наші практики DevOps:

- *Unity Collaborate*: так як необхідно було зберегти великі сцени та префаби, ми використовували внутрішні сервіси Unity.
- *Тестування*: регулярне використання Unity Test Framework (Edit Mode / Play Mode), написання unit-тестів для критичних елементів логіки.
- *Збереження білдів*: кожна стабільна збірка зберігалася на Google Drive, з назвою та індексацією за версією.

РОЗДІЛ 2

РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ

Серверна частина проєкту реалізована у вигляді розподіленої peer-to-peer архітектури, що ідеально підходить для невеликої кооперативної гри, де один із клієнтів виступає хостом сесії. Для забезпечення стабільної, надійної та безпечної взаємодії між гравцями було обрано платформу Steamworks, яка надала можливість організувати сесії через Steam Lobby System та забезпечити P2P-з'єднання між учасниками гри.

Компоненти серверної частини:

- Steam Lobby System - використовується для створення, пошуку та управління ігровими сесіями. Кожна сесія ініціюється одним із гравців, який автоматично отримує роль хоста. Система підтримує передачу метаданих (ім'я лоббі, кількість гравців, статус тощо) та фільтрацію за параметрами.
- Peer-to-Peer архітектура - обмін ігровими даними реалізований без централізованого сервера. Всі повідомлення між клієнтами передаються через Steam Networking API, що забезпечує мінімальні затримки та автоматичну маршрутизацію трафіку.
- Локальне збереження даних - усі налаштування користувача (мова, гучність, інверсія управління, ім'я гравця тощо) зберігаються на боці клієнта за допомогою PlayerPrefs, що забезпечує швидкий доступ до персональних параметрів без додаткового навантаження на мережу.

Використані мови, середовища та технології

- Мова програмування: C#
- Середовища розробки: Unity, Visual Studio, Visual Studio Code
- Підключені бібліотеки та плагіни:
 - Steamworks.NET - бібліотека для інтеграції з Steam API;
 - Unity Input System - сучасна система обробки вводу з підтримкою налаштувань користувача;
- Ігрові технології Unity:
 - Unity UI Toolkit - побудова інтерфейсу з можливістю гнучкого редагування;
 - TextMeshPro - для високоякісного рендерингу шрифтів;
 - Addressable Assets - керування ресурсами без потреби перезавантаження сцени;
 - Animator Controller - для управління анімаціями персонажів та об'єктів;
 - NavMesh - навігація для NPC (антагоніста).
 - Animation Rigging - точне налаштування анімацій та інверсної кінематики для реалістичного руху персонажів і взаємодії з предметами.
 - Localization Tables — централізована система перекладів для підтримки багатомовності у грі.
 - Light System — динамічне та статичне освітлення для створення атмосфери напруги та хоррору.
 - Post Processing — покращення візуальної якості зображення через сучасні ефекти глибини різкості, шуму, bloom, віньєтування тощо.

Система локалізації

Для реалізації багатомовного інтерфейсу ми використали вбудовану в Unity Localization System, яка дозволила забезпечити повну підтримку декількох мов без дублювання сцен чи UI.

Основні характеристики реалізації:

- Усі текстові елементи інтерфейсу, включно з кнопками, повідомленнями, підказками та діалогами, були винесені в Localization Tables;
- Використано формат String Tables у поєднанні з Addressables, що дозволило змінювати мову "на льоту", без перезапуску гри чи сцени;
- Підтримувані мови:
 - Українська — основна мова проєкту;
 - Англійська — як альтернативна, для розширення доступності проєкту на міжнародну аудиторію;
- Гравець міг змінити мову в налаштуваннях, після чого всі інтерфейсні елементи оновлювалися автоматично.

Тестування серверної частини

- Функціональне тестування P2P-з'єднань та лоббі здійснювалося у локальному середовищі за допомогою транспортних засобів по типу KCP.
- Юніт-тестування виконувалося через Unity Test Framework у режимах PlayMode та EditMode;
- Особливу увагу було приділено перевірці коректності синхронізації ігрових подій між клієнтами, а також стабільності з'єднання при втраті одного з гравців або при поганому інтернет-з'єднанні.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ АДМІНІСТРАТИВНОЇ ЧАСТИНИ

Реалізація головоломок та завдань у грі

У ході розробки ігрового проєкту було створено низку інтерактивних завдань, які формують основну частину ігрового процесу. Головоломки поєднують елементи логічного мислення, дослідження простору, взаємодії з об'єктами та сюжетного напруження. Вони поступово відкривають гравцю нові ділянки будинку, нову інформацію про події, що відбулися, та створюють атмосферу постійної напруги та цікавості.

Пошук ключів

Базовим елементом прогресії в грі є система ключів. У будинку розташовано велику кількість замкнених дверей, які потребують пошуку відповідних ключів для відкриття. Деякі з ключів лежать на видних місцях, інші добре заховані — у шафах, шухлядах, вентиляційних шафах або інших важкодоступних локаціях. Гравцю необхідно уважно досліджувати кожну кімнату, помічати дрібні деталі, що можуть навести на слід схованих ключів.

Взаємодія з девайсами: телефон та пульт

Важливою частиною геймплею є інтерактивні електронні пристрої, з якими можна взаємодіяти безпосередньо:

Телефон: гравець може запускати програми, знаходити підказки, отримувати нову інформацію та, врешті-решт, скористатися додатком телефону для виклику поліції. Телефон працює у вигляді повноцінного окремого інтерфейсу з меню та додатками.

Пульт керування: дозволяє керувати окремими механізмами в будинку, наприклад — відкривати двері гаража.

Взаємодія з такими пристроями реалізована через систему наведення камери на об'єкт та подальшу інтеракцію за допомогою миші.

Система фокусної взаємодії з різними об'єктами

Окрім девайсів, гравець може безпосередньо керувати об'єктами:

комп'ютерами, де можна переглядати файли;

електричними щитами, що відповідають за живлення різних зон;

іншими механічними елементами інтер'єру, по типу генератора, усіляких замків тощо.

При фокусуванні камери активується спеціальний режим керування, що дає змогу детально працювати з об'єктами.

Головоломка з генератором

Одним із ключових завдань є запуск резервного генератора. Його пуск вимагає правильного виставлення позицій кількох важелів, кнопок і перемикачів. Послідовність правильних налаштувань щоразу формується випадковим чином, тож кожен запуск гри потребує нових рішень. Підказка до правильних налаштувань знаходиться у книзі-мануалі, яку гравцеві ще потрібно знайти.

Завдання з комп'ютером та флешкою

В окремій кімнаті розташований комп'ютерний блок, доступ до якого відкривається після знаходження флеш-накопичувача. Після підключення флешки на комп'ютері активується прихована папка, що містить важливу інформацію — зокрема, коди для доступу предметів по типу кейпаду.

Система з вентиляційною шафою

У вентиляційній шафі часто заховані ключові предмети. Для отримання доступу гравцеві спочатку потрібно відкрутити гвинти захисної решітки викруткою. Якщо позаду решітки працює вентилятор, його необхідно зупинити шляхом перерізання живлення пласкогубцями, після чого можна безпечно дістати прихований предмет.

Відкривання дверей за допомогою відмички

Для деяких дверей передбачено альтернативний спосіб відкривання — злам замків відмичкою. Цей процес реалізовано через QTE-систему: на екрані з'являється послідовність клавіш, які необхідно вчасно натискати, аби успішно зламати замок.

Відновлення мобільного зв'язку

Основна сюжетна задача полягає у відновленні мобільного зв'язку з метою виклику поліції. Щоб запустити антену, необхідно:

знайти чотири запобіжники;

вставити їх у зовнішній електрощит;

попередньо знайти код для відкриття щита, оскільки він захищений паролем.

Після відновлення зв'язку відкривається можливість зробити виклик.

Окультна головоломка з пентаграмою

У підвалі розташована кровава пентаграма. Для активації ритуалу гравець повинен розшукати спеціальний проклятий предмет та помістити його в центр пентаграми. Після ритуалу з'являється привид у вигляді дитини, що передає ключ до схованої шкатулки. У шкатулці знаходиться важлива сюжетна записка, яка розкриває фінальні деталі сюжету.

Система електропостачання та управління освітленням

У підвалі розташований головний електрощит будинку, який керує освітленням у всіх приміщеннях. Кожен з шести перемикачів відповідає за конкретні зони. У процесі гри привид може впливати на систему живлення, створюючи для гравців додаткові труднощі — періодично вимикаючи світло в окремих кімнатах, змушуючи гравців спускатися до підвалу для відновлення освітлення.

Висновки

У межах реалізації ігрового проєкту було спроектовано та впроваджено низку інтерактивних механік, що формують ядро ігрового процесу. Основну увагу зосереджено на розробці головоломок, які поєднують елементи логіки, дослідження простору, сюжетної напруги та взаємодії з об'єктами.

Здійснено проєктування системи пошуку ключів, яка забезпечує поступове відкриття нових приміщень та просування гравця за сюжетом. Реалізовано взаємодію з інтерактивними пристроями (мобільний телефон, пульт керування), що додає реалістичності та варіативності ігровим ситуаціям. Створено механізми точкової взаємодії з комп'ютерами, генератором, електрощитами, замками, вентиляційною системою тощо.

Впроваджено динамічну головоломку з генератором, де конфігурація перемикачів змінюється в кожному проходженні, що сприяє реіграбельності. Реалізовано завдання з комп'ютером та флешкою, систему злому замків за допомогою відмички (QTE), а також квест із відновлення мобільного зв'язку через пошук запобіжників.

Окрему увагу приділено містичним елементам: створено окультну головоломку з пентаграмою та викликом привида, що інтегрується у сюжетну лінію. Розроблена система енергопостачання дозволяє привиду впливати на освітлення, посилюючи напругу та потребу у стратегічному мисленні гравця.

Загалом реалізовані механіки забезпечують занурення у тривожну атмосферу гри, сприяють нарощенню напруги та активному залученню гравця в сюжет. За результатами внутрішнього тестування підтверджено ефективність ігрових рішень, логічність побудови головоломок та їх відповідність жанру психологічного горору. Водночас виявлено перспективи для подальшого вдосконалення, зокрема в аспектах UX-оптимізації та балансу складності завдань.

Перелік використаних джерел

1. Unity Manual – Animator Controller
<https://docs.unity3d.com/Manual/Animator.html>
2. Unity Manual – Addressable Asset System
<https://docs.unity3d.com/Packages/com.unity.addressables@1.21/manual/index.html>
3. Unity UI Toolkit Documentation
<https://docs.unity3d.com/Manual/UIElements.html>
4. Unity AI Navigation (NavMeshAgent, AI Agents)
<https://docs.unity3d.com/ScriptReference/NavMeshAgent.html>
5. Unity Animation Rigging Package
<https://docs.unity3d.com/Packages/com.unity.animation.rigging@1.2/manual/index.html>
6. Unity Developer Tutorials <https://www.youtube.com/@Brackeys>
7. Observing playres behaviour playing horror games <https://www.twitch.tv/>
8. Unity TextMeshPro Documentation
<https://docs.unity3d.com/Packages/com.unity.textmeshpro@3.0/manual/index.html>
9. Vegas Pro Online Tutorials
<https://www.vegascreativesoftware.com/us/online-tutorials/>
10. Learn Photoshop <https://www.adobe.com/learn/photoshop>
11. Unity Post Processing Stack Documentation
<https://docs.unity3d.com/Packages/com.unity.postprocessing@3.2/manual/index.html>
12. Unity Localization Package
<https://docs.unity3d.com/Packages/com.unity.localization@1.3/manual/index.html>
13. Steamworks API Documentation
<https://partner.steamgames.com/doc/sdk>
14. Steamworks.NET (обгортка Steamworks SDK для Unity)
<https://steamworks.github.io/>
15. Mirror Networking Documentation
<https://mirror-networking.gitbook.io/docs/>