Software Requirements Specification

for

<QCellGen>

Version 1.0 approved

Prepared by <Ковальов Микита Віталійович>

<ст. гр. ІПЗм-23-3>

<10.06.2025>

Зміст

Зміст ii

Історія змін ii

1. Вступ 1

1.1 Призначення 1

1.2 Прийняті умовності 1

1.3 Цільова аудиторія та порядок читання 1

1.4 Опис продукту 1

1.5 Посилання 1

2. Загальний опис 2

2.1 Загальний контекст 2

2.2 Основні функції продукту 2

2.3 Класи користувачів і їх характеристики 2

2.4 Робоче середовище 2

2.5 Обмеження проєктування та реалізації 2

2.6 Документація для користувачів 2

2.7 Припущення та залежності 3

3. Вимоги до зовнішніх інтерфейсів 3

3.1 Інтерфейс користувача 3

3.2 Апаратні інтерфейси 3

3.3 Програмні інтерфейси 3

3.4 Комунікаційні інтерфейси 3

4. Системні функції 4

4.1 Генерація початкового рівня 4

4.2 Навчання агента Q-Learning 4

4.3 Перевірка прохідності та валідація рівня 4

4.4 Експорт результатів 4

5. Інші нефункціональні вимоги 4

5.1 Вимоги до продуктивності 4

5.2 Вимоги до безпеки 5

5.3 Вимоги до захисту даних 5

5.4 Атрибути якості програмного забезпечення 5

5.5 Бізнес-правила 5

Історія змін

|  |  |  |  |
| --- | --- | --- | --- |
| **Ім’я** | **Дата** | **Причина змін** | **Версія** |
| Ковальов Микита | 01.06.25 | Початкова версія документа | 0.1 |
| Ковальов Микита | 05.06.25 | Додано розділи 3–5 | 0.2 |
| Ковальов Микита | 10.06.25 | Фіналізація | 1.0 |

# Вступ

## Призначення

Цей документ містить специфікацію вимог до програмного забезпечення, що реалізує систему процедурної генерації ігрових рівнів на основі алгоритму Q-Learning. Поточна версія охоплює функціональність створення рівнів, навчання агента в середовищі Unity та аналізу ефективності згенерованого контенту. Документ описує одну підсистему загального процесу створення ігрового середовища — саме модуль генерації рівнів.

## Прийняті умовності

Документ структуровано відповідно до стандарту IEEE 830. Заголовки розділів виділяються напівжирним шрифтом. Усі функціональні вимоги мають пріоритет однакової важливості, якщо не вказано окремо. Термін агент використовується для позначення об'єкта, що навчається змінювати стан ігрового рівня на основі зворотного зв’язку.

## Цільова аудиторія та порядок читання

Цей документ призначений для розробників, тестувальників, технічних менеджерів, дослідників у сфері процедурної генерації, а також для осіб, що займаються проєктуванням геймплейних механік.

Рекомендовано спочатку ознайомитися з розділами 1–2 для розуміння цілей і загальної архітектури, а далі перейти до розділів 3–4, які детально описують функціональні й нефункціональні вимоги

## Опис продукту

Програмне забезпечення призначене для автоматизованого створення рівнів у 2D-іграх за допомогою алгоритму навчання з підкріпленням (Q-Learning). Основна мета — забезпечення генерації прохідних, збалансованих та різноманітних рівнів без участі дизайнера. Система має бути інтегрована у середовище Unity як частина ігрового проєкту. Результати навчання повинні бути візуалізовані та проаналізовані для подальшого вдосконалення геймплею.

## Посилання

– Sutton R., Barto A. Reinforcement Learning: An Introduction, 2nd edition, 2018.  
– Togelius J. et al. Procedural Content Generation in Games: A Textbook, Springer, 2016.  
– Документація Unity: https://docs.unity3d.com  
– Python бібліотеки: https://pandas.pydata.org, <https://matplotlib.org>  
– Стандарт IEEE 830-1998 – IEEE Recommended Practice for Software Requirements Specifications

# Загальний опис

## Загальний контекст

Продукт є самостійним програмним модулем, призначеним для інтеграції в ігрове середовище, створене на платформі Unity. Система не є частиною існуючого продуктового сімейства, а розробляється як новий прототип для дослідницьких цілей. Вона реалізує функціональність процедурної генерації рівнів із використанням алгоритму Q-Learning, де агент навчається змінювати структуру рівня на основі зворотного зв’язку з середовищем.

Модуль взаємодіє з внутрішніми компонентами Unity: системою тайлмапів, pathfinding API, механізмом оновлення сцени та збереженням станів. Зовнішні інтерфейси представлені у вигляді експорту даних (CSV) для подальшої обробки Python-інструментами.

## Основні функції продукту

Програмне забезпечення забезпечує наступну функціональність:

– Ініціалізація базового рівня за допомогою клітинного автомата  
– Запуск агента Q-Learning з параметрами навчання  
– Модифікація рівня агентом згідно з обраною дією  
– Обчислення функції винагороди та оновлення Q-таблиці  
– Перевірка прохідності згенерованого рівня  
– Збір статистичних даних і експорт у форматі CSV  
– Візуалізація змін у реальному часі у вікні редактора Unity

## Класи користувачів і їх характеристики

**Розробники:** використовують систему для інтеграції у більші ігрові проєкти; мають досвід роботи з Unity та C#  
**Дослідники / студенти:** аналізують поведінку агента, модифікують гіперпараметри; можуть мати базові знання з reinforcement learning  
**Тестувальники:** перевіряють коректність генерації рівнів та поведінки агента; не обов’язково мають глибоку технічну підготовку

Основний акцент зроблено на підтримці зрозумілого коду, параметризованості й простоти повторного запуску навчання.

## Робоче середовище

– **Unity Editor 2022.3.30f LTS**  
– **.NET Standard 2.1**  
– **Системні вимоги:** ОС Windows 10+, мінімум 8 ГБ ОЗП, GPU з підтримкою DirectX 11  
– **Додаткове ПЗ:** Python 3.11 з бібліотеками Pandas, Matplotlib для аналізу результатів

## Обмеження проєктування та реалізації

– Алгоритм Q-Learning реалізовано без використання зовнішніх ML-бібліотек — лише на C#  
– Взаємодія з середовищем обмежується API Unity  
– Візуалізація проводиться у 2D-просторі з використанням Tilemap  
– Зберігання даних – тільки у форматі CSV  
– Без підтримки багатокористувацького режиму або мережевої взаємодії  
– Код має бути уніфікований для подальшої інтеграції у ігрові проєкти

## Документація для користувачів

– Інструкція користувача (PDF): запуск системи, зміна параметрів агента, читання результатів  
– README-файл у репозиторії (Markdown): інструкція зі збирання, налаштування й повторного запуску  
– Коментарі до коду та назви класів – англійською мовою, відповідно до стандартів Unity C# API

## Припущення та залежності

– Передбачається, що Unity Editor встановлено та налаштовано належним чином  
– Python-інструменти встановлюються користувачем самостійно  
– Ефективність роботи агента залежить від правильно налаштованих параметрів ε, α, γ  
– Робота не передбачає інтеграції з хмарними або сторонніми API  
– Проєкт залежить від стабільної роботи Tilemap та Pathfinding API всередині Unity

# Вимоги до зовнішніх інтерфейсів

## Інтерфейс користувача

Інтерфейс реалізовано у середовищі Unity Editor. Всі взаємодії здійснюються через інспектор об’єкта-агента та ігрову сцену. Основні елементи інтерфейсу:

– **Панель налаштувань агента:** поля для введення параметрів ε, α, γ, кількість епізодів  
– **Кнопки запуску / зупинки навчання**  
– **Вивід статистики у редакторі (Unity UI):** кількість дій, винагорода, стан рівня  
– **Візуалізація рівня:** сцена Unity Tilemap у 2D-режимі з кольоровим кодуванням (стіни, прохідні клітини, агент тощо)  
– **Стандарти:** інтерфейс реалізується за внутрішніми стилістичними правилами Unity UI; всі повідомлення про помилки подаються через консоль Unity або у вигляді виводу у вікні сцени.

Детальний опис UI структури оформлюється окремо у специфікації користувацького інтерфейсу.

## Апаратні інтерфейси

Продукт не взаємодіє безпосередньо з фізичним обладнанням. Єдиною апаратною вимогою є сумісність з обладнанням, яке забезпечує роботу Unity Editor:

– Платформи: ПК під управлінням Windows 10+  
– Необхідне: процесор з підтримкою x64, 8+ ГБ ОЗП, GPU з підтримкою DirectX 11  
– Додатково: доступ до файлової системи для збереження CSV-даних

Прямої взаємодії з периферійними пристроями (сенсори, контролери) не передбачено.

## Програмні інтерфейси

Програмний продукт взаємодіє з наступними компонентами:

– **Unity Tilemap API** — для створення, оновлення та візуалізації ігрових рівнів  
– Unity NavMesh / A Pathfinding API\* — для перевірки прохідності рівнів  
– **Unity Editor UI System** — для побудови інтерфейсів введення/виведення параметрів  
– **.NET API** — для внутрішньої обробки даних, таймерів, генерації випадкових значень  
– **Файлова система (CSV)** — для збереження результатів навчання та статистики  
– **Python (зовнішньо, опціонально)** — обробка даних, побудова графіків через Pandas та Matplotlib

Жодних підключень до баз даних або сторонніх API не передбачено. Обмін даними здійснюється через локальні файли.

## Комунікаційні інтерфейси

Продукт не вимагає підключення до мережі або взаємодії з веб-службами. Жодні протоколи (HTTP, FTP, SMTP) не використовуються.

Комунікація із зовнішнім ПЗ (наприклад, Python-аналітикою) виконується опосередковано — через локальні CSV-файли, які зберігаються у визначену директорію. Безпосередній обмін повідомленнями або синхронізація з іншими системами не передбачені.

# Системні функції

## Генерація початкового рівня

4.1.1 Опис та приоритет

Система повинна мати можливість автоматично створювати початковий рівень за допомогою клітинного автомата, що генерує псевдовипадкову структуру карти.  
Пріоритет: **Високий** (Benefit: 9, Penalty: 8, Cost: 3, Risk: 2)

4.1.2 Послідовність стимулів\реакцій

– Користувач натискає кнопку «Генерувати рівень»  
– Система створює масив клітин з випадковим розташуванням стін  
– Рівень відображається у вигляді Tilemap на сцені Unity

4.1.3 Функціональні вимоги

REQ-1: Система повинна генерувати рівень розміром N×M клітин, де N та M задаються в параметрах

REQ-2: Частка стін повинна бути в межах заданого діапазону (наприклад, 30–60%)

REQ-3: Всі клітини мають бути правильно візуалізовані за допомогою Unity Tilemap

REQ-4: У разі некоректних параметрів генерації виводити повідомлення про помилку

## Навчання агента Q-Learning

4.1.1 Опис та приоритет

Система повинна дозволяти запуск навчання агента з використанням алгоритму Q-Learning для оптимізації рівня.  
Пріоритет: **Високий** (Benefit: 9, Penalty: 9, Cost: 5, Risk: 4)

4.1.2 Послідовність стимулів\реакцій

– Користувач задає гіперпараметри (ε, α, γ)  
– Натискає «Почати навчання»  
– Система виконує епізоди: агент оцінює стан, вибирає дію, отримує винагороду, оновлює Q-таблицю  
– По завершенню епізодів виводиться звіт

4.1.3 Функціональні вимоги

REQ-5: Система повинна підтримувати встановлення гіперпараметрів ε, α, γ вручну  
REQ-6: Агент повинен зберігати та оновлювати Q-таблицю в оперативній памʼяті  
REQ-7: Після кожної дії має обчислюватись винагорода згідно з функцією винагороди  
REQ-8: При досягненні фінішного стану рівень вважається успішним, епізод завершується  
REQ-9: У випадку збоїв під час навчання — припинити процес і вивести повідомлення з логом

## Перевірка прохідності та валідація рівня

4.1.1 Опис та приоритет

Система повинна перевіряти, чи є згенерований рівень прохідним — від старту до цілі має існувати маршрут.  
Пріоритет: **Середній** (Benefit: 8, Penalty: 6, Cost: 2, Risk: 3)

4.1.2 Послідовність стимулів\реакцій

– Після завершення навчання або окремої дії система викликає pathfinding  
– Якщо маршрут знайдено — рівень валідний  
– Якщо ні — агент отримає штраф

4.1.3 Функціональні вимоги

REQ-10: Перевірка прохідності має виконуватись після кожної дії агента  
REQ-11: Якщо шлях не знайдено — агент отримує негативну винагороду  
REQ-12: Алгоритм перевірки може бути A\* або NavMesh (налаштовується вручну)

## Експорт результатів

4.1.1 Опис та приоритет

Система повинна експортувати дані навчання (винагорода, кількість дій, статус рівня) у CSV для подальшого аналізу.  
Пріоритет: **Середній**

4.1.2 Послідовність стимулів\реакцій

– Після завершення навчання користувач натискає «Експортувати»  
– Система формує таблицю та зберігає у вказану директорію

4.1.3 Функціональні вимоги

REQ-13: Експорт має включати стовпці: № епізоду, винагорода, кількість кроків, пройдений/непройдений  
REQ-14: Формат збереження — .csv із UTF-8 кодуванням  
REQ-15: У випадку недоступної директорії — повідомити про помилку

# Інші нефункціональні вимоги

## Вимоги до продуктивності

– Середній час генерації одного рівня не повинен перевищувати **500 мс** у режимі попереднього перегляду  
– Навчання агента повинно обробляти не менше **50 епізодів за хвилину** при стандартних параметрах  
– Візуалізація змін рівня має оновлюватись в реальному часі без помітних затримок (FPS ≥ 30)  
– Збір та експорт статистики не повинен впливати на продуктивність основного потоку Unity

Ці вимоги пов’язані з інтеграцією продукту у живе ігрове середовище та тестуванням у реальному часі.

## Вимоги до безпеки

– Система не повинна виконувати жодних дій, що змінюють середовище за межами Unity (наприклад, запис у сторонні директорії, мережеві запити)  
– Під час збереження результатів у файл має перевірятись коректність шляху та прав доступу  
– Усі дії системи виконуються локально, без залучення зовнішніх API чи серверів  
– Випадки винятків (exceptions) під час роботи повинні оброблятись, щоб уникнути аварійного завершення програми

Хоча система працює в офлайн-середовищі, безпечна обробка вводу та виходу є обов’язковою.

## Вимоги до захисту даних

– Дані, зібрані в ході навчання (Q-таблиця, статистика), зберігаються локально у відкритому форматі CSV  
– Доступ до файлів обмежується правами операційної системи користувача  
– Жодні особисті або конфіденційні дані не збираються  
– Передбачена перевірка шляху запису, щоб уникнути перезапису критичних системних файлів

Аутентифікація або контроль доступу не потрібні, оскільки продукт не має багатокористувацького або мережевого режиму.

## Атрибути якості програмного забезпечення

– **Надійність:** система повинна витримувати ≥95% безпомилкових запусків без збоїв  
– **Портативність:** система має працювати на будь-якому ПК з Unity Editor (версія 2022.3+) без модифікацій  
– **Придатність до повторного використання:** алгоритм Q-Learning реалізовано у вигляді окремого C#-компоненту  
– **Простота тестування:** кожен компонент має бути покритий модульними тестами або підтримувати ручне тестування  
– **Гнучкість:** параметри навчання змінюються без редагування коду  
– **Юзабіліті:** інтерфейс налаштувань інтуїтивно зрозумілий, передбачено стандартні підказки

Пріоритет надається: **надійності > адаптивності > простоті використання**.

## Бізнес-правила

– Зміна параметрів навчання може здійснюватись лише до запуску агента  
– Доступ до вихідного коду та результатів дослідження обмежується внутрішнім використанням у межах освітнього/наукового проєкту  
– Результати експериментів зберігаються у визначеній директорії з унікальними назвами для кожної сесії  
– У разі порушення логіки (наприклад, непридатна карта або некоректний ввід) система не повинна починати навчання