

АНАЛІЗ ЕФЕКТИВНОСТІ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ЛАБІРИНТІВ ДЛЯ МУЛЬТИПЛЕЄРНИХ 3D ВЕБ-ІГОР НА БАЗІ THREE.JS

*Мацій Ольга Борисівна, доцент кафедри штучного інтелекту
та програмного забезпечення*

Харківського національного університету імені В.Н. Каразіна

*Романов Микита Вячеславович, студент факультета комп'ютерних наук
Харківського національного університету імені В.Н. Каразіна*

Сучасна ігрова індустрія представлена веб-іграми, які дозволяють гравцям швидко отримати доступ до ігрового контенту без потреби встановлення додаткового програмного забезпечення. Однією з передових технологій для створення 3D веб-ігор є Three.js, яка робить можливим розробку високоякісних ефектів та складних тривимірних сцен безпосередньо у веб-браузері. Водночас, процедурна генерація контенту, а саме лабіринтів, відкриває нові можливості для збільшення варіативності ігрових світів та розширення геймплею, не потребуючи значних додаткових зусиль при розробці.

Процурна генерація контенту – це метод автоматичного створення даних за допомогою алгоритмів. У сфері веб-ігор це означає генерування унікальних ігрових елементів, як-от лабіринти, ландшапти або випадкові події, без прямої участі розробників. Процурна генерація дозволяє створювати велику кількість контенту з обмежених ресурсів, забезпечуючи високу різноманітність ігрового досвіду [1].

До переваг процедурної генерації можна віднести її здатність створювати практично необмежену кількість унікального ігрового контенту, скорочуючи таким чином час і витрати на розробку, а також підвищуючи реграбельність, пропонуючи нові ігрові сценарії з кожним проходженням. Однак він також має свої недоліки, такі як непередбачуваність якості згенерованих структур, складність у збалансуванні ігрового процесу та технічні проблеми, пов'язані з оптимізацією продуктивності.

Three.js – це високорівнева бібліотека на основі WebGL, що дозволяє створювати складні 3D-сцени у браузері. Розробка 3D веб-ігор з її допомогою має низку переваг, серед яких використання стандартних веб-технологій (HTML, CSS, JavaScript) для створення інтерактивного 3D-контенту, легка інтеграція з іншими веб-сервісами та платформами, а також доступ до великої спільноти розробників. Однак оптимізація продуктивності для забезпечення безперебійної роботи 3D-додатків у різних веб-браузерах і на різному обладнанні є складним завданням для розробників [2].

Розробка методології для оцінки якості генерації лабіринтів була спрямована на аналіз складності, різноманітності та здатності лабіринтів сприяти захопливому ігровому процесу. У цьому процесі брали участь відгуки користувачів та кількісні показники, включаючи час, потрібний для проход-

ження лабіринту, та кількість ігрових сесій. Аналіз продуктивності, особливо з використанням інструментів для відстеження ресурсів браузера та комп'ютера, виявив важливі відомості про вплив різних методів генерації на швидкість відгуку та ефективність системи.

Порівняння процедурно генерованих лабіринтів із статичними, створеними вручну, підтвердило переваги у відтворюваності та різноманітності, хоча і виявило проблеми з оптимізацією продуктивності. На основі зібраних даних було доопрацьовано методи генерації, щоб виправити ці проблеми, звернувши особливу увагу на відгуки користувачів та аналіз продуктивності. Це призвело до розробки адаптивних алгоритмів, які можуть динамічно змінювати складність лабіринту, враховуючи кількість гравців та їхні уподобання, оптимізуючи при цьому навантаження на систему [3].

Особлива увага приділялася оптимізації продуктивності, в тому числі через використання ресурсів браузера. Розробники впровадили такі техніки, як ліниве завантаження та інстанціювання об'єктів, щоб мінімізувати споживання пам'яті та час завантаження, що сприяло швидкому відображенню складних 3D-лабіринтів навіть на обмеженому обладнанні.

Покращення візуальних ефектів та анімації за допомогою Three.js дозволило створити більш реалістичні освітлення, тіні, віддзеркалення та заломлення світла, забезпечуючи захоплюючий 3D-досвід, який використовує весь потенціал веб-технологій.

На завершальному етапі тестування з використанням автоматизованих інструментів, таких як Lighthouse в Google Chrome, відіграло ключову роль у виявленні та усуненні вузьких місць в коді, гарантуючи, що кінцевий продукт відповідає високим стандартам якості та продуктивності. Цей процес підкреслив важливість гнучкого підходу до розробки, який може адаптуватися до потреб користувачів і технологічних обмежень.

Література

1. Procedural Content Generation For Games: A Survey. URL: https://www.researchgate.net/publication/262327212_Procedural_Content_Generation_for_Games_A_Survey
2. Three.js Documentation. URL: <https://threejs.org/docs/>
3. Evaluating The Impact of Procedurally Generated Content on Game Immersion. URL: https://www.researchgate.net/publication/319173732_Evaluating_the_Impact_of_Procedurally_Generated_Content_on_Game_Immersion