

200 РОКІВ
ОСВІТНІХ ТРАДИЦІЙ



Том 1

**ТЕЗИ
71-ої наукової конференції
професорів, викладачів, наукових
працівників, аспірантів та студентів університету**

**ПОЛТАВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ЮРІЯ КОНДРАТЮКА**

Міністерство освіти і науки України
Північно-Східний науковий центр НАН України та МОН України
Полтавський національний технічний університет
імені Юрія Кондратюка

Тези

71-ої наукової конференції професорів,
викладачів, наукових працівників, аспірантів
та студентів університету

Том 1

22 квітня – 17 травня 2019 р.

Полтава 2019

ПЕРЕВАГИ ТА НЕДОЛІКИ ВИКОРИСТАННЯ GODOT ENGINE ПРИ РОЗРОБЦІ КООПЕРАТИВНОЇ КОМП'ЮТЕРНОЇ ГРИ

Комп'ютерні ігри є значною частиною індустрії інформаційних технологій, цим обумовлено поширення технологій з відкритим доступом для створення комп'ютерних ігор різноманітного жанру та типу. Кооперативні комп'ютерні ігри є одним з найпопулярніших типів, участь в яких приймають від двох та більше гравців, усі завдання вирішуються спільно. Попит на кооперативні комп'ютерні ігри позначився і на ринку ігрових рушіїв, наразі практично кожен рушій має мережевий функціонал та готові рішення для створення ігор з доступом до мережі з мінімальними витратами зі сторони розробника. Низка популярних на сьогодні рушіїв таких як Unity, Cry Engine, Unreal Engine або Gamemaker наділені мережевими елементами, деякі містять базові елементи (чати або низькорівнева передача даних).

З метою розробки власної кооперативної комп'ютерної гри використано Godot Engine – безкоштовний рушій з відкритим кодом, що дозволяє створювати ігри для більшості сучасних популярних платформ з використанням спільної кодової бази та має широкий спектр ефективних інструментів для роботи в трьох (3D) та двовимірних (2D) просторах. Головною перевагою Godot Engine є його повна безкоштовність.

Рушій Godot Engine для розробки мережевої частини гри має невелику кількість варіантів. Наразі існують два доступні варіанти: використання вбудованого функціоналу – Godot Engine має вбудоване API для роботи з мережевими функціями через HTTP, UDP, TCP та SSL протоколи; створення власної мережевої системи на мові C#. Недоліком використання Godot Engine при розробці саме мережевих рішень є відсутність готових рішень від сторонніх розробників. Оскільки рушій є відносно новим і немає уваги зі сторони великих компаній, таких як Photon, DarkLib або PlayFab, які декілька років постачають стабільні рішення для інших ігрових рушіїв.

При розробці кооперативної комп'ютерної гри основна її частина виконується у 3D, для роботи з яким Godot Engine має широкий спектр вбудованих інструментів. Головними компонентами, які необхідні для роботи є графіка та сам код гри, який здійснює рух супротивників по карті. Графіка є індивідуальним компонентом для будь-якого ігрового рушія, оскільки потреби розробників відмінні, існують потреби реалістичного

зображення, цим обумовлено використання всіх можливостей цільової платформи, разом з тим іншим розробникам достатнє використання простих двовимірних зображень на однокольоровому фоні. Розробка Godot Engine розпочалася як рушій для 2D ігор, на сьогодні його інструменти для роботи в 3D відповідають стандартам індустрії та напряду працюють з більшістю форматів для 3D-моделей. Існує пряма підтримка імпортів з пакету Blender з перенесенням усіх компонентів ігрових сцен. Одним з недоліків роботи з 3D в Godot Engine є відсутність підтримки формату FBX, що є стандартом комп'ютерної індустрії впродовж багатьох років та відсутність документації щодо роботи у вбудованому редакторі шейдерів, це не дозволяє з легкістю розпочати роботу.

Логіка кооперативної комп'ютерної гри обумовлена власною мовою програмування Godot Engine – GDScript, що є гібридом Python та Lua популярність яких постійно зростає. Мова є динамічною, тобто відсутня потреба в компіляції коду до початку роботи, код компілюється при виконанні програми, це дозволяє швидко вносити зміни, уникаючи очікування кожного разу при перезапуску проекту.

Найбільш трудомісткою частиною для реалізації коду є створення системи навігації для супротивників, що представляють загрозу для гравців. Рушій Godot Engine має два варіанти для подальшої роботи: Navigation – вбудована система навігації, що працює як у 2D, так і в 3D. Ця система дозволяє повністю автоматизувати отримання шляху з точки А в точку Б простими функціями, але використання супротивників різного розміру через один і той самий вузол Navigation не є можливим; AStar – вбудований навігаційний метод, що використовує алгоритм пошуку A* («А зірочка» або «А star») для прокладання маршрутів через встановлені користувачем точки у дво- або трьохвимірному просторі. Недоліком є необхідність ручного розставлення та з'єднання точок поверхонь для сітки навігації через код. При розробці кооперативної мережевої комп'ютерної гри використано навігаційний метод AStar.

Таким чином, Godot Engine є багатофункціональним ігровим рушієм, який має необхідний для зручної роботи інструментарій та дозволяє швидко та якісно виконати проекти різноманітного розміру. Основною причиною більшості існуючих недоліків є його недостатня популярність на ринку ігрових рушіїв, це обмежило кількість сторонніх розробників, які мали можливість покращити рушій за рахунок додаткових модулів.

Література

- 1. Кріс Бредфілд. Розробка ігрових проектів на Godot Engine – Packet Publishing, 2018. – 298 с.*
- 2. Роберт Цієсла. Godot 2.1 Джуліана Лінецькі, Арієля Манзура// Розробка майже без коду: Новітні рушії для ігор – Apress, 2017. – 233 с.*