

**РЕАЛИЗАЦИЯ АЛГОРИТМА ПОИСКА ПУТИ В НАВИГАЦИОННЫХ СЕТКАХ A*
С ИСПОЛЬЗОВАНИЕМ ШАБЛОНА ПРОЕКТИРОВАНИЯ ECS ДЛЯ
КРОССПЛАТФОРМЕННОЙ СРЕДЫ РАЗРАБОТКИ ИГРОВЫХ ПРИЛОЖЕНИЙ UNITY**

И. Д. ХИРЬЯНОВ

*(Полоцкий государственный университет
имени Евфросинии Полоцкой, Беларусь)*

Аннотация. В докладе рассматриваются ключевые этапы разработки алгоритма поиска пути A* с использованием технологии ECS. Сравнение данной реализации со стандартной по ключевым параметрам: скорость нахождения, правильность нахождения.

Ключевые слова: алгоритм поиска пути, ECS, A*, игровое приложение, NPC.

Введение. Алгоритм поиска пути в игровых приложениях – это алгоритм, который используется для нахождения пути между двумя точками на игровом поле. Он может быть использован для перемещения персонажей, NPC, машин и других объектов в игре. Существует множество алгоритмов поиска пути, каждый из которых имеет свои преимущества и недостатки.

A* – это алгоритм поиска кратчайшего пути, который является усовершенствованием алгоритма Дейкстры за счёт введения эвристики. А именно: переход осуществляется в ту вершину, предположительный путь от которой до конечной является кратчайшим. Чаще всего пользуются эвристикой Манхэттена, но помимо неё возможно использование эвристик Чебышева и Евклида [1].

Была поставлена цель разработать алгоритм, который мог бы максимально быстро и точно определить кратчайший путь от точки до точки. A* на фоне остальных показывает себя лучше всего используется как в игровой разработке, так и, например, в автомобильных навигаторах или решении сложных математических задач.

Анализ средств реализации. Для реализации алгоритма поиска пути была выбрана кроссплатформенная среда разработки игровых приложений Unity. Данная среда имеет ряд преимуществ на фоне других, за счет наличия таких вещей как: визуальной среды разработки; межплатформенной поддержки; модульной системы компонентов.

Языком программирования для написания стал C#, так как он является языком разработки в Unity.

Главной особенностью разрабатываемого алгоритма является использование паттерна ECS. Это шаблон проектирования «Сущность Компонент Система»

(Entity Component System). То есть «Сущности» (Entity) – объекты-контейнеры, не обладающие свойствами, но выступающие хранилищами для «Компонентов» (Component). «Компоненты» – это блоки данных, определяющие всевозможные свойства любых игровых объектов или событий. Все эти данные, сгруппированные в контейнеры, обрабатываются логикой, существующей исключительно в виде «Систем» (System) – «чистых» классов с определенными методами для выполнения. Данный паттерн является независимым от какой-либо среды разработки игровых приложений и может быть реализован множеством способов. Все «сущности», «системы» и «компоненты» должны где-то храниться и каким-то образом инициализироваться – все это является особенностями реализации каждого ECS решения для конкретной среды разработки игр [2–4].

Плюсы и минусы использования ECS:

Плюсы:

- многопоточность из коробки;
- компиляция скриптов в машинный код;
- cache-friendly компоновка данных;
- расширяемость архитектуры.

Минусы:

- сложнее для неопытных разработчиков;
- не все features были реализованы для ECS.

Реализация и тестирование. Было реализовано две версии программы для тестирования алгоритма A*, один вариант с использованием паттерна ECS, второй – без. По три теста каждой из версий представлены на рисунках 1–6.

Исходя с приведенных тестов видно, что обе версии справляются со своей задачей и находят самый короткий маршрут. Но версия с использованием паттерна ECS работает в несколько раз быстрее. Для наглядности в таблице приведено сравнение по времени нахождения оптимального пути.

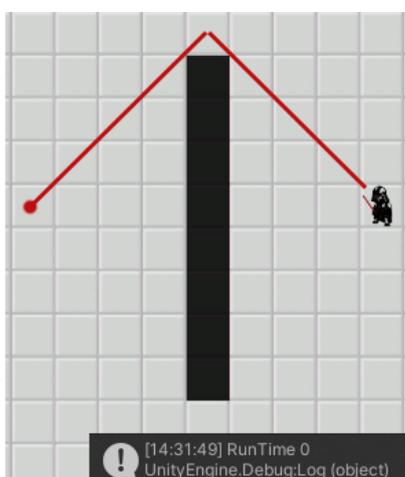


Рисунок 1. – Тест № 1 с использованием ECS

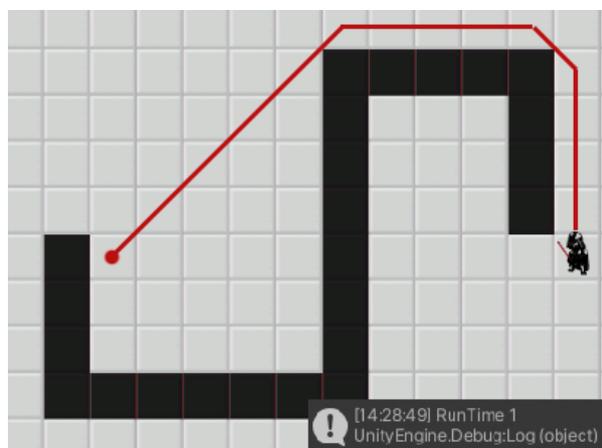


Рисунок 2. – Тест № 2 с использованием ECS

2. Entity Systems Wiki [Электронный ресурс]. – Режим доступа: <http://entity-systems.wiki-dot.com/>. – Дата доступа: 15.02.2024.
3. Неизвестный шаблон проектирования. [Электронный ресурс]. – Режим доступа: <https://dev.to/ovid/the-unknown-design-pattern-1l64>. – Дата доступа: 10.02.2024.
4. Мартин, Адам. Часть 2 – Что такое система сущностей? // Системы сущностей - будущее разработки MMOG.