



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

КУРСОВОЙ ПРОЕКТ КОМПЬЮТЕРНАЯ 2D ИГРА «НЕЙРОНЫ И ГЕРОИ»

Выполнил студент группы БПИ-201
Красковский Дмитрий Дмитриевич

Научный руководитель:
Старший преподаватель
Департамента программной
инженерии факультета компьютерных наук
Пантюхин Дмитрий Валерьевич

Москва, 2022



ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметная область:

- **Аркадная игра с простым, но интенсивным процессом**
- **Изучение искусственного интеллекта противника, используя средства геймификации**

“Neurons and Heroes” - игра, в которой можно узнать много нового в области искусственного интеллекта, но при этом отдохнуть и насладиться интересным игровым процессом. В перед началом игры пользователю предлагается изменить параметры искусственного интеллекта противника.

Игра завершается, когда герой “освободил принцессу” или был настигнут врагами



АКТУАЛЬНОСТЬ РАБОТЫ

С каждым годом игровое сообщество развивается всё быстрее, и быстрее. Игры, выпускаемые большими компаниями, всё меньше похожи на игры, которые выпускались несколько лет назад. В связи с этим, у многих людей возникает чувство ностальгии, когда они притрагиваются к играм, которые не похожи на современные большие проекты, а напоминают аркады из игровых клубов. Данное приложение позволяет испытать это чувство.

Ещё с каждым ходом всё чаще поднимается вопрос искусственного интеллекта – всё больше людей стремятся изучить современные технологии в этом направлении. В связи с этим данная игра будет полезна с обучающей точки зрения – искусственный интеллект противников, построенный на алгоритме поиска пути в графе A* (“A-star”) можно будет настраивать, изменяя некоторые параметры в меню настроек



ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы:

Создание однопользовательской игры, напоминающей ряд старых аркадных игр, с возможностью настраивать искусственный интеллект противника

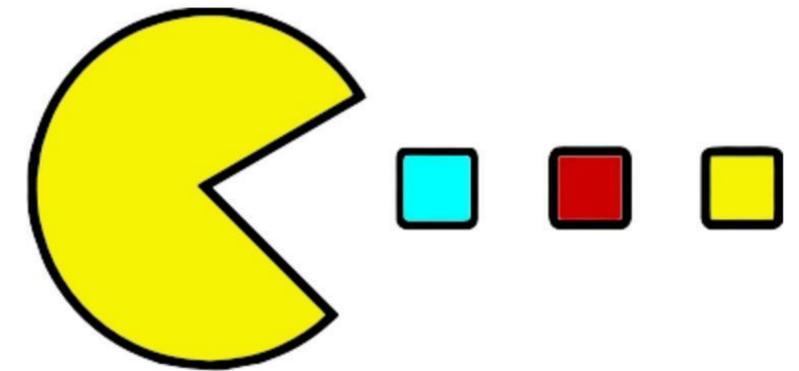
Задачи работы:

1. Анализ существующих решений;
2. Извлечение функциональных требований;
3. Выбор технологий для реализации;
4. Изучение выбранных технологий;
5. Создание уровней (карт) для игры;
6. Реализация функционала героя;
7. Реализация искусственного интеллекта противников;
8. Разработка GUI главного меню и меню настроек;
9. Связывание параметров искусственного интеллекта противников и параметров в меню настроек;
10. Разработка технической документации.

АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

На момент создания приложения существует множество как отечественных, так и зарубежных аналогов за счёт простоты создания. Самая известная и самая первая – это Pac-Man, популярность которого никто не смог переплюнуть. Все остальные игры, находятся в тени этой игры, но бывают исключения: например Tomb of the Mask. Это происходит из-за того, что механика игры у всех однообразная. Данная игра будет отличаться своей информативностью и полезностью, так как в процессе игры пользователь ознакоми́вается с искусственным интеллектом, меняя параметры и осознавая, какие из них меняют сложность игры.

Другой пример популярного, но не похожего на Pac-Man аналога – это Super Mario. Главная цель этой игры – тоже спасти принцессу, избежав на своём пути противников. Было выпущено много пародий на эту игру, но игра “Neurons and Heroes”, как и в прошлом примере, нечто большее – за счёт своей обучающей составляющей.





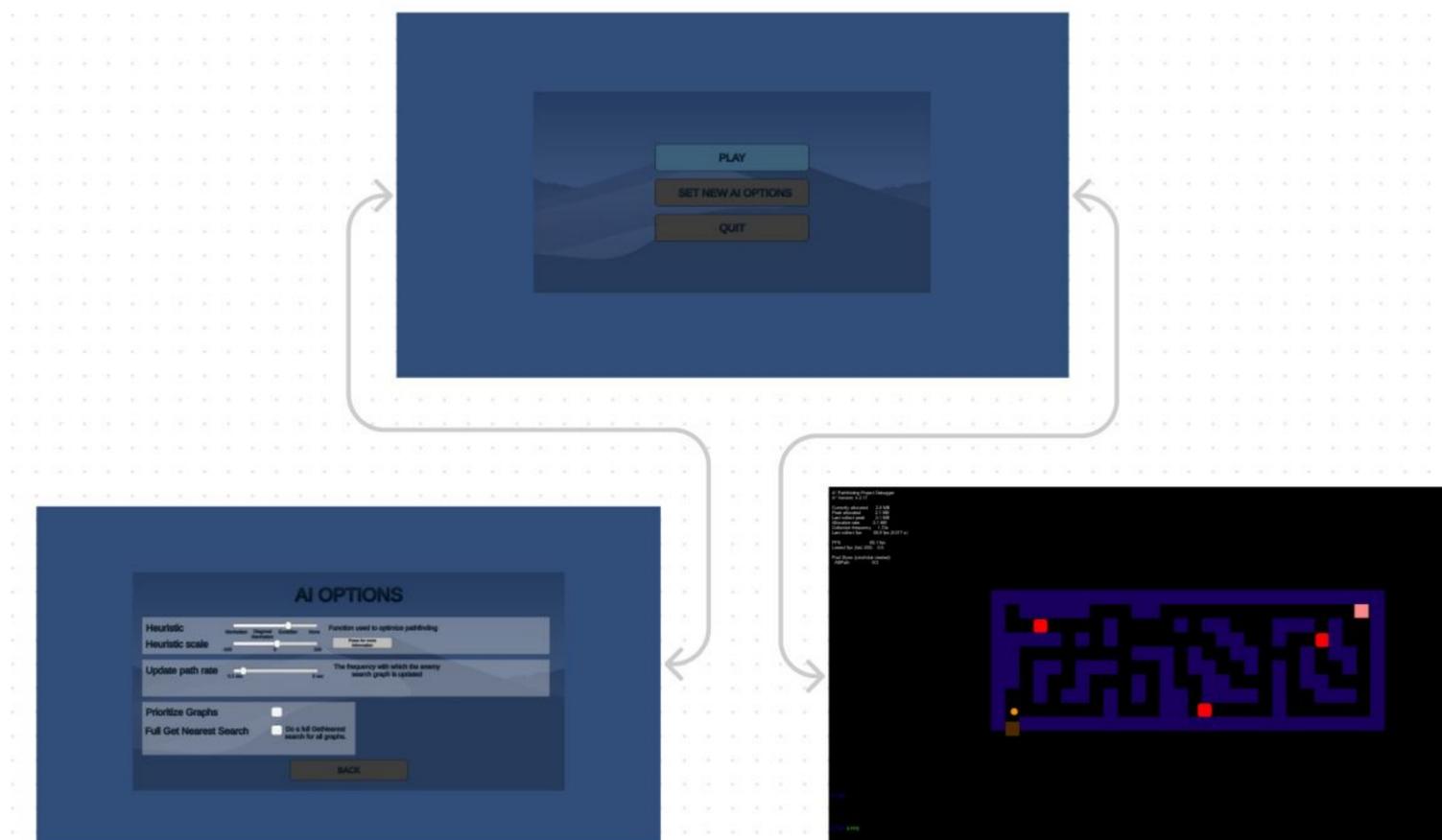
ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Список требований

- Управление персонажем, а также отображение его сверху;
- Инициализация карты, случайным образом выбранной из списка карт;
- Инициализация героя, противника и всех сопутствующих объектов игры;
- Инициализация объекта «принцесса» при приближении героя к заданной точке;
- Изменение настроек искусственного интеллекта противника с последующим его анализом;
- Переход из игры в меню и наоборот.

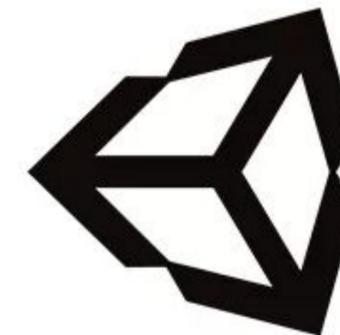
ОПИСАНИЕ РЕАЛИЗАЦИИ

Общая схема работы приложения



ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РЕАЛИЗАЦИИ

- Unity 2020.3.32f1
- JetBrains Rider 2022.1.1
- Figma
- C# v.9.0





ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

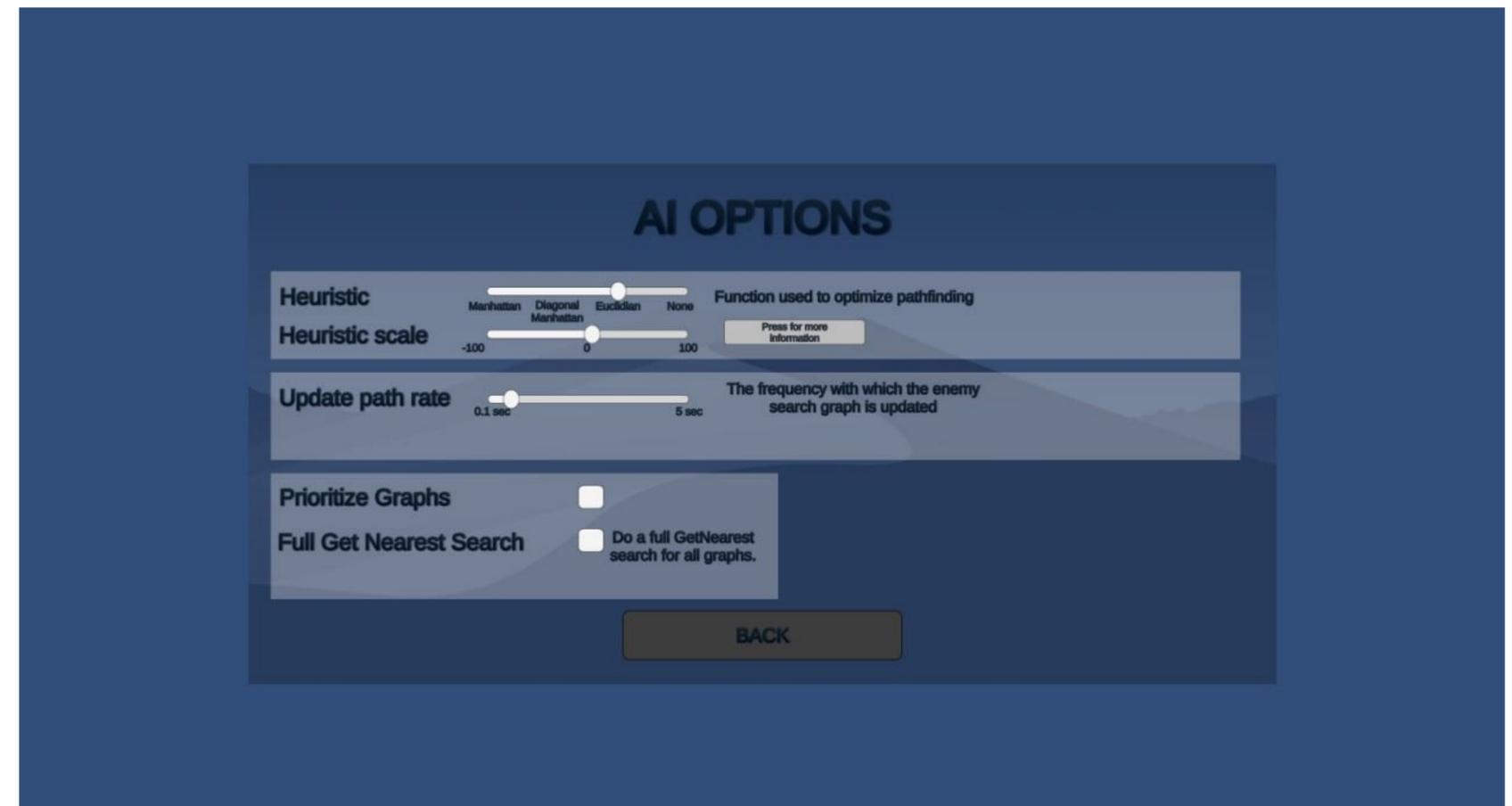
Главное меню



ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

Меню настроек

Здесь находятся все параметры, характеризующие искусственный интеллект, которые можно изменить и опробовать в игре. К этим параметрам даются пояснения. В процессе разработки было решено выбрать английский язык, для более широкой аудитории, а также из соображений, что в игру будут играть люди, давно занимающиеся обучением и знающие английский язык.





ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

Меню настроек

- Heuristic – выбор функции, используемой для оптимизации поиска пути в графе
- Heuristic scale - Если используется значение меньше 1, навигатор будет искать больше узлов (медленнее). Если используется значение 0, алгоритм поиска пути будет сведен к алгоритму Дейкстры. Это эквивалентно установке для эвристики значения None. Если используется значение больше 1, поиск пути (обычно) будет быстрее, потому что он расширяет меньшее количество узлов, но пути могут больше не быть оптимальными (т.е. кратчайшими возможными путями).



ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

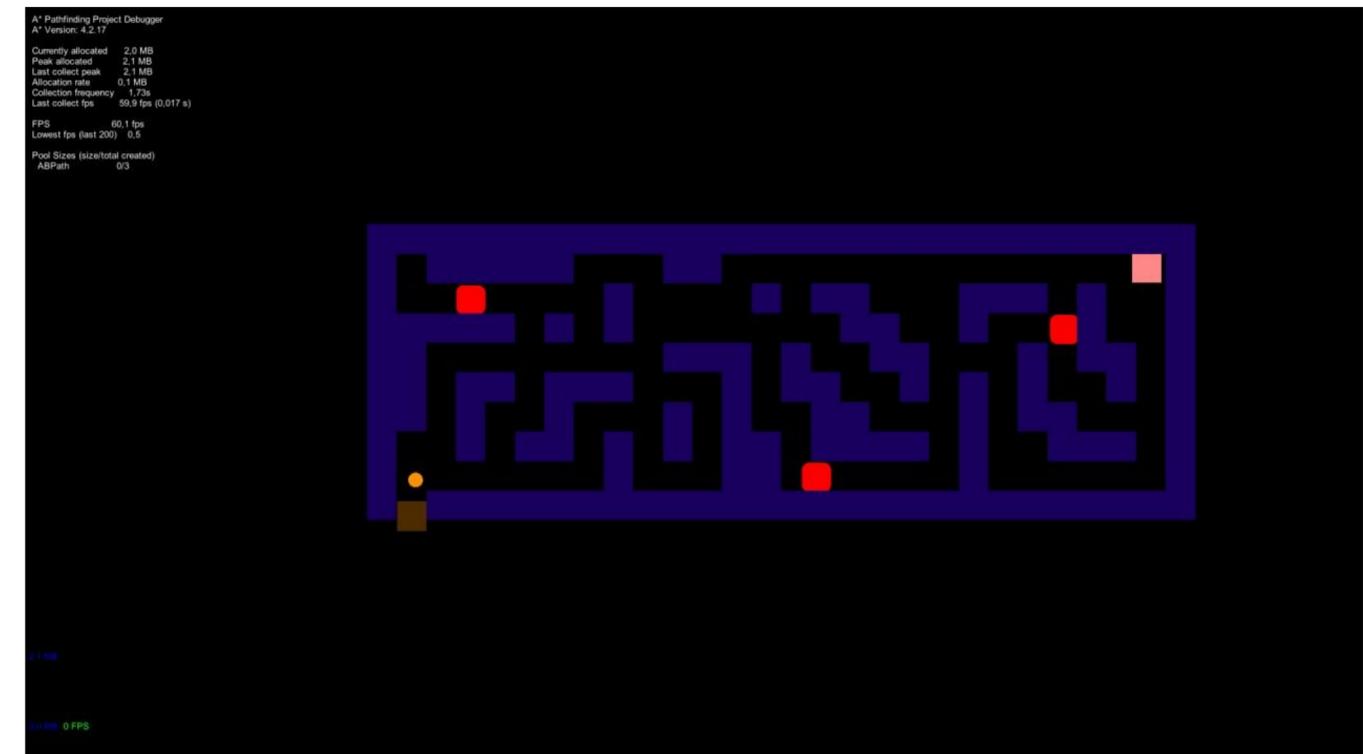
Меню настроек

- Update path rate – частота, с которой обновляется граф пути
- Full get nearest search – если стоит значение true, то будет осуществляться полный поиск в графе с перебором всех возможных путей. Это замедлит работу ИИ, но может немного оптимизировать путь.

ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

Основная игра

При запуске игры случайным образом выбирается карты. Игроку надо добраться до противоположного края карты, спасти принцессу и вернуться обратно, избежав столкновения с врагами, которые будут либо преследовать его, либо отслеживать периметр (в зависимости от выбранных игроком параметров их искусственного интеллекта)



ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

Основная игра

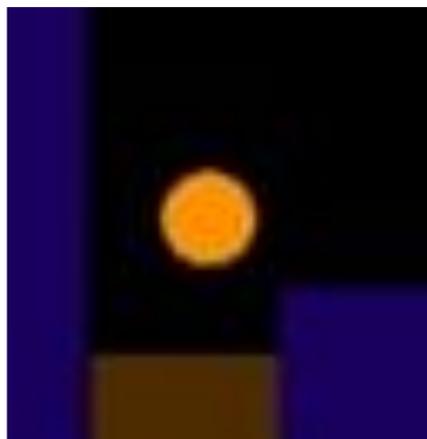


Рис. 1
Главный игрок

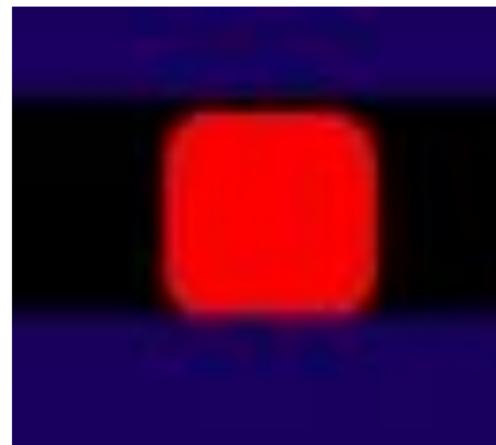


Рис. 2
Один из врагов

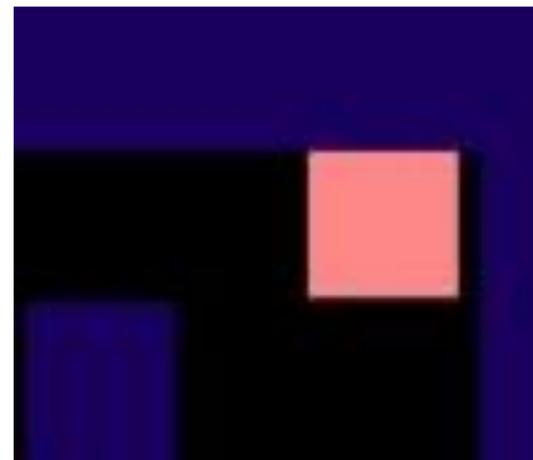


Рис. 3
Темница, в которой
заточена принцесса

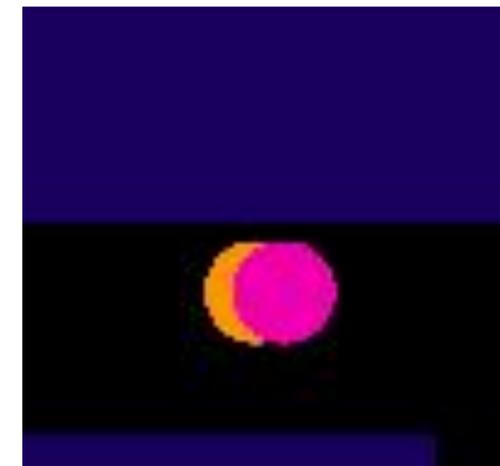


Рис. 4
Принцесса, которая движется
за героем после спасения



ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

Код

Перемещение главного игрока

```
private void FixedUpdate()
{
    var horizontal:float = Input.GetAxis("Horizontal");
    var vertical:float = Input.GetAxis("Vertical");
    rb.velocity = new Vector2(x:horizontal * speed, y:vertical * speed);
}
```

Спасение принцессы

```
public void HeroTakePrincess()
{
    hero.speed /= 2;
    princess.gameObject.SetActive(true);
    Debug.Log(message: "You take princess!!");
    SetWithPrincess(result: true);
}
```



ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

Код

Игрок покидает пещеру

```
1 usage
public void HeroLeaveDungeon()
{
    if (WithPrincess)
    {
        GameEnd(message: "You win");
    }
    else
    {
        hero.ResetPosition();
    }
}
```

Игра завершается

```
2 usages
private void GameEnd(string message)
{
    Debug.Log(message);
    fields[currentField].gameObject.SetActive(false);

    for (int i = 0; i < this.enemies.Length; i++)
    {
        enemies[i].gameObject.SetActive(false);
    }
    hero.gameObject.SetActive(false);

    princess.gameObject.SetActive(false);

    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex - 1);
}
```



ДЕМОНСТРАЦИЯ ПРИЛОЖЕНИЯ

Код

Применение введённых пользователем настроек

```
Event function
void Start()
{
    updateRate = MenuOptions.UpdateRate;
    heuristicScale = MenuOptions.HeuristicScale;
    heuristic = MenuOptions.Heuristic;

    prioritizeGraphs = MenuOptions.PrioritizeGraphs;
    prioritize = MenuOptions.Prioritize;
    fullGetNearestSearch = MenuOptions.FullGetNearestSearch;
    _active = AstarPath.active;
    AllUpdate();
    InvokeRepeating(methodName: "UpdatePath", time: updateRate, updateRate);
}
```

Обновление графа, по которому перемещаются враги

```
Frequently called 1 usage
void UpdatePath()
{
    if (_active != null)
    {
        _active.Scan();
    }
}
```



ПУТИ ДАЛЬНЕЙШЕЙ РАБОТЫ

1. Добавление новых параметров, характеризующих работу ИИ противника
2. Сделать больше карт, которые будут выбираться случайным образом
3. Заменить спрайты, из которых состоит карта, на спрайты стен, используя Tilemaps – инструмент Unity
4. Добавить более подробное и понятное описание к каждому параметру ИИ противника
5. Добавить звуки происходящих событий
6. Сделать процесс игры более разнообразным, добавляя новые сложности (например ключ, без которого не спасти принцессу)
7. Сделать возможность выбора русского языка



СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- Unity Documentation [Электронный ресурс] / Документация Unity, используемая для конструирования приложения. URL: <https://docs.unity.com/> (дата обращения: 01.05.2022) Режим доступа: свободный
- A* Project [Электронный ресурс] / Проект, используемый для подключения искусственного интеллекта. URL: <https://arongranberg.com/> (дата обращения: 01.05.2022) Режим доступа: свободный
- Документация по C# [Электронный ресурс] / Используется для написания скриптов в конструкторе Unity. URL: <https://docs.microsoft.com/ruru/dotnet/csharp/> . (дата обращения: 1.05.2022) Режим доступа: свободный
- Обучающие статьи по Unity3D [Электронный ресурс] URL: <https://unity3d.com/learn/tutorials> (дата обращения: 1.05.2022) Режим доступа: свободный.
- Список относительно старых игр [Электронный ресурс] / Использовал для нахождения косвенных и прямых конкурентов. URL: <https://goldies.ru/games/?genre=ARCADE> (дата обращения: 14.05.2022) Режим доступа: свободный



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Спасибо за внимание!

Красковский Дмитрий
Дмитриевич
ddkraskovskiy@edu.hse.ru
Москва - 2022