

Создание “искусственного интеллекта” для врага на основе ассета A* Pathfinding

Для начала необходимо перейти по ссылке <https://arongranberg.com/astar/download> и скачать бесплатную версию “A* Pathfinding”.

A* - это метод, с помощью которого решается задача на поиск лучшего и короткого маршрута.

Далее нужно перенести скачанный файл в папку Assets и в появившемся окне нужно выбрать все галочки, после чего нажать на кнопку “import”, указанную на рисунке 74

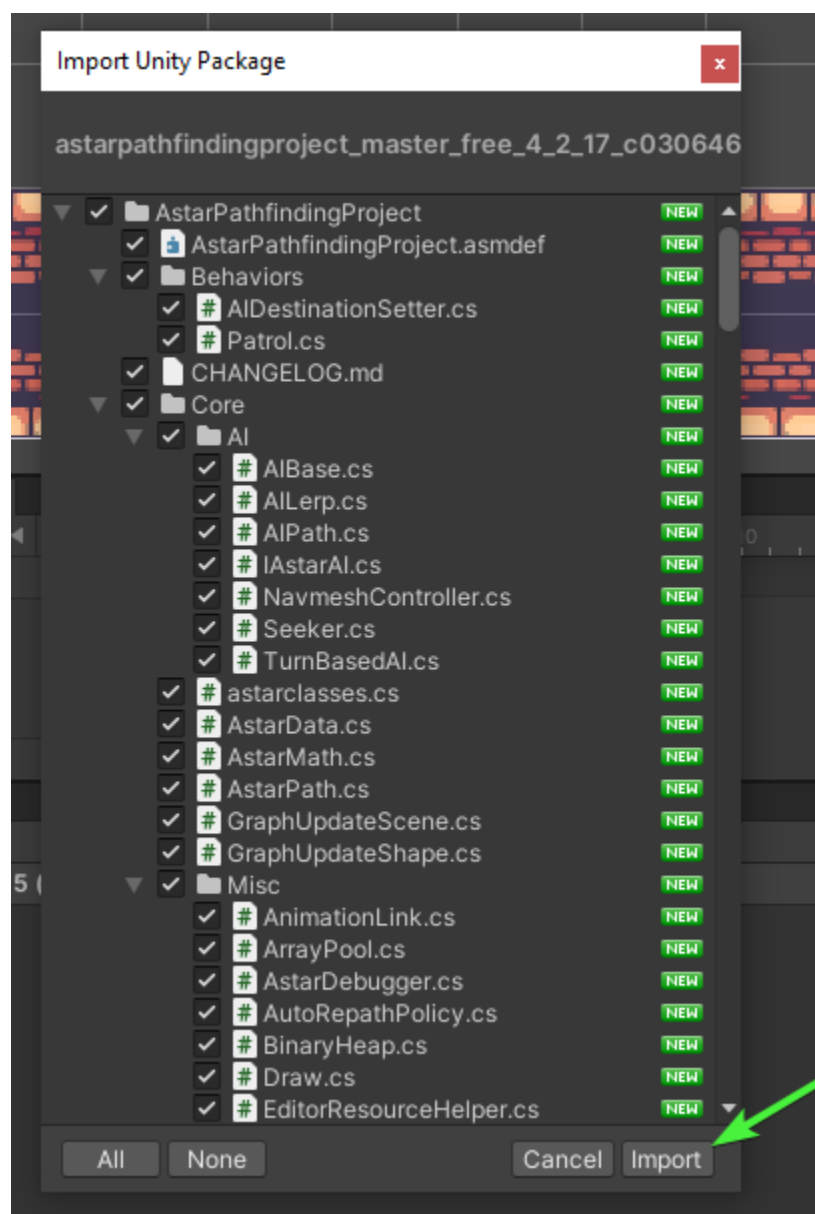


Рисунок 74. Окно со всеми компонентами ассета “A* Pathfinding”.

После того как “A* Pathfinding” был добавлен в проект нужно создать пустой объект в сцене и назвать его “A* Pathfinding”. Созданный объект будет отвечать за поле зрения врага. Далее нужно добавить ему в компоненты скрипт с названием “Pathfinder” как показано на рисунке 75.

После добавления компонента нужно нажать на вкладку “Graphs” и создать новый граф нажав на кнопку “Grid Graph”, как указано на рисунке 76. После создания нового графа его нужно настроить как показано на рисунке 77.

Граф - это модель, отражающая взаимные связи между объектами, которая состоит из множества вершин и соединяющих их ребер. При этом значение имеет только факт соединения вершин друг с другом.

Описание параметров:

1. “Node size” - отвечает за размер клеток в области поля зрения врага. Нужно уменьшить данный параметр для большей точности.
2. “Diameter” - отвечает за размер коллайдера, который используется для построения пути от врага к персонажу.
3. “Obstacle Layer Mask” - отвечает за объекты, через которые нельзя прокладывать путь. Объекты выбираются по выбранному слою в данном параметре. На рисунке 78 показано сканирование области видимости врага и выделение тех мест через которые враг не может строить свой путь.

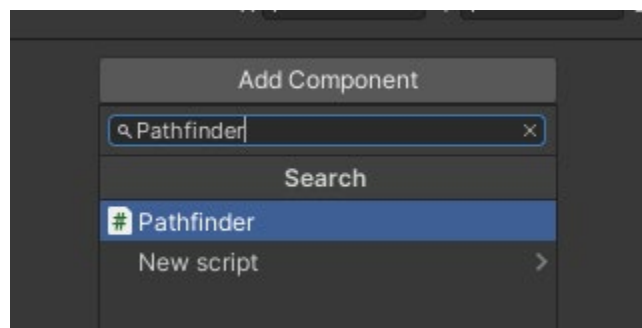


Рисунок 75. Добавления компонента Pathfinder.

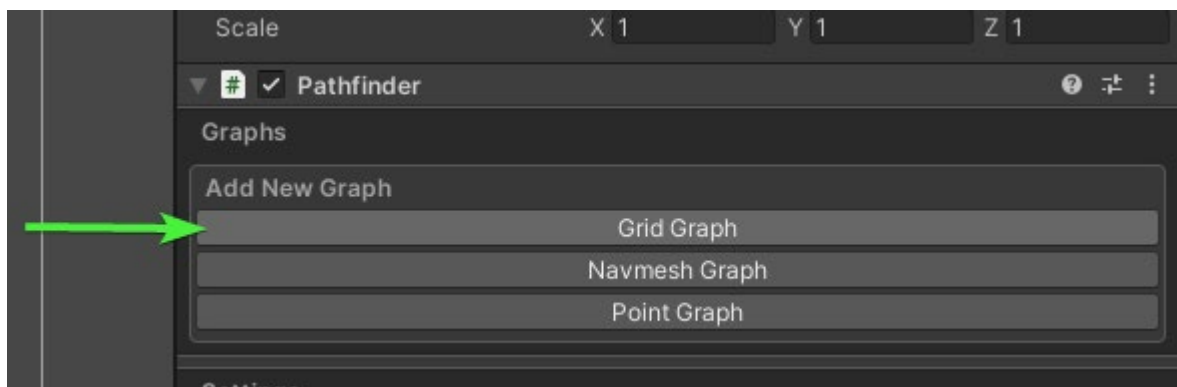


Рисунок 76. Создание нового графа.

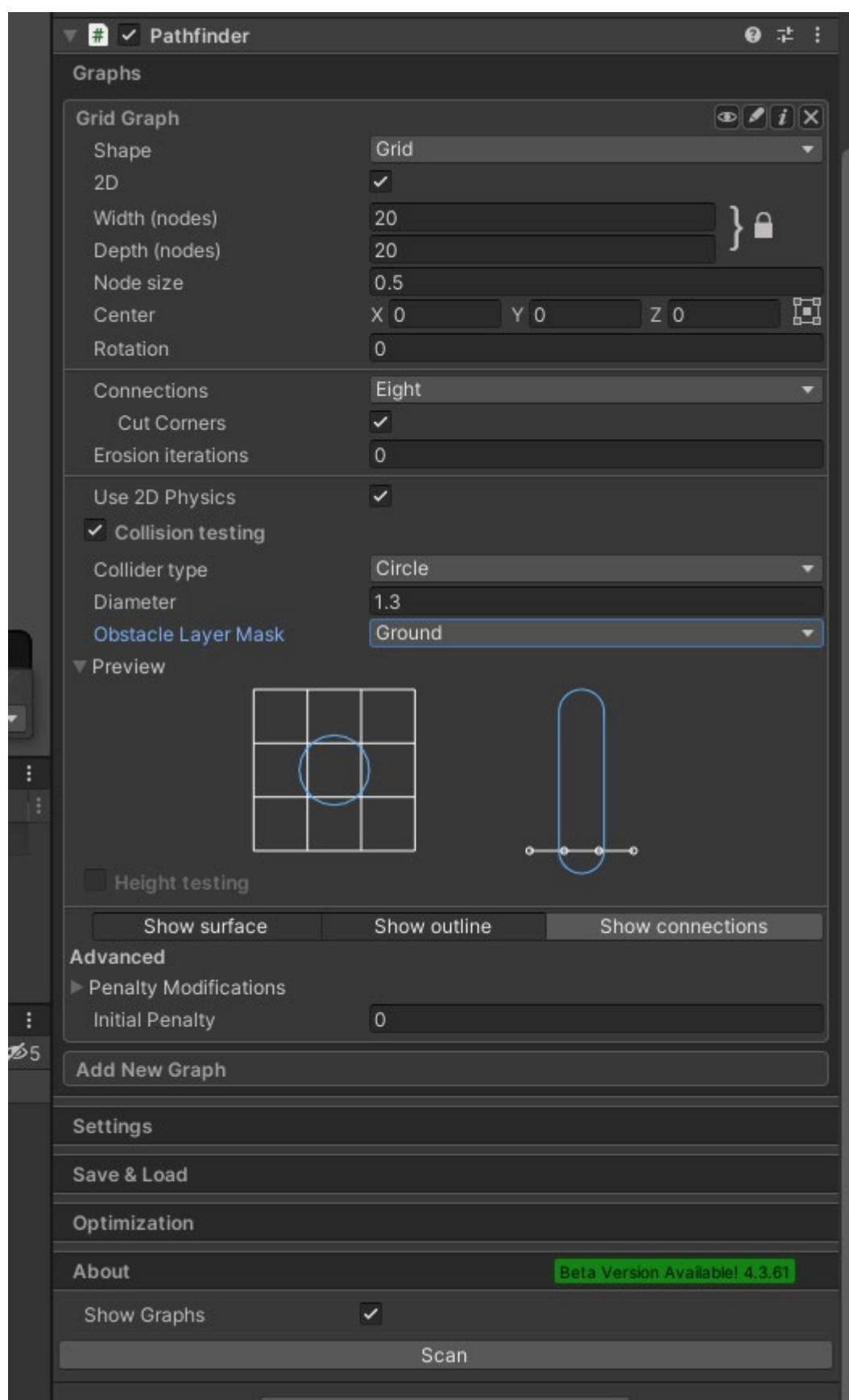


Рисунок 77. Настройка графа.

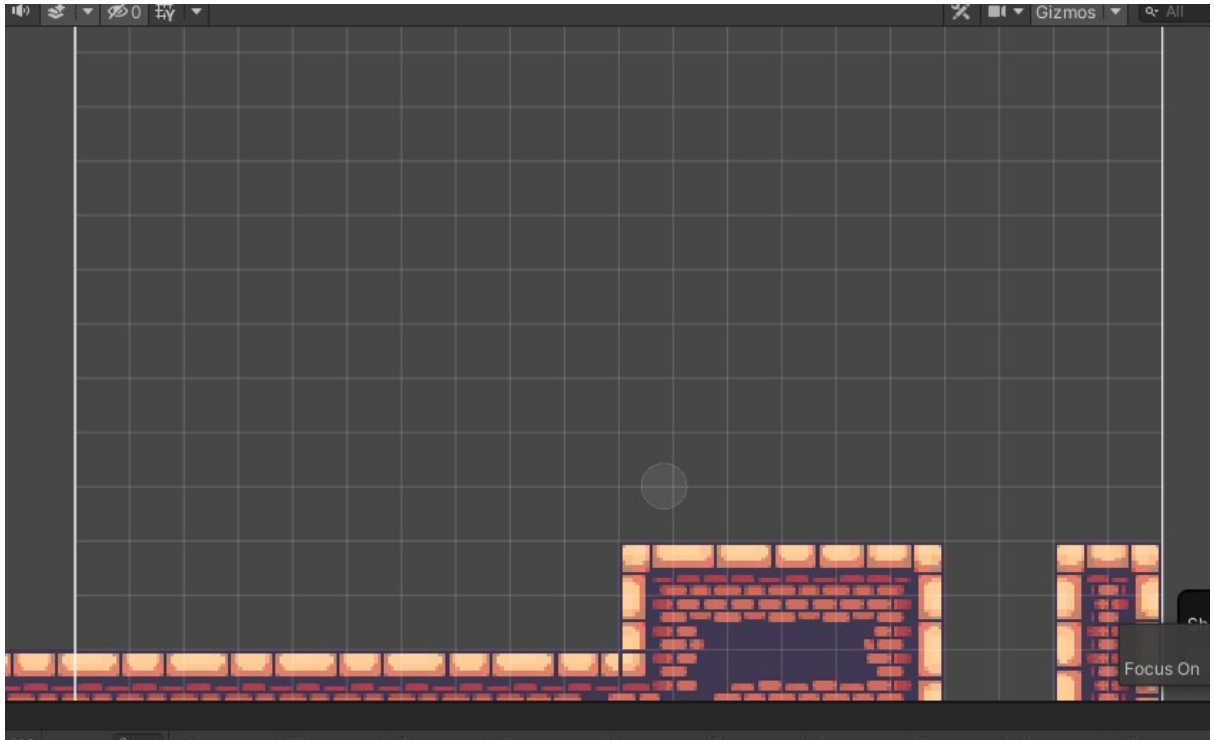


Рисунок 78. Результат сканирование области видимости врага

Теперь необходимо создать противника, для этого нужно создать пустой объект в иерархии и назвать его “Mob”, после чего добавить спрайт и настроить анимацию как было показано в части про создание персонажа.

После добавления врагу спрайта, ему нужно добавить в компоненты скрипт “AIPath” как показано на рисунке 79. Далее необходимо настроить данный компонент, как показано на рисунке 80.

Описание параметров:

1. “Gravity” - отвечает за гравитацию. Её нужно отключить так как противник летает. Для наземных врагов данный параметр нужно включить.
2. “Orientation” - выбор пространства, в котором должен ориентироваться персонаж.

3. “Enable Rotation” - отвечает за вращения врага во время движения.

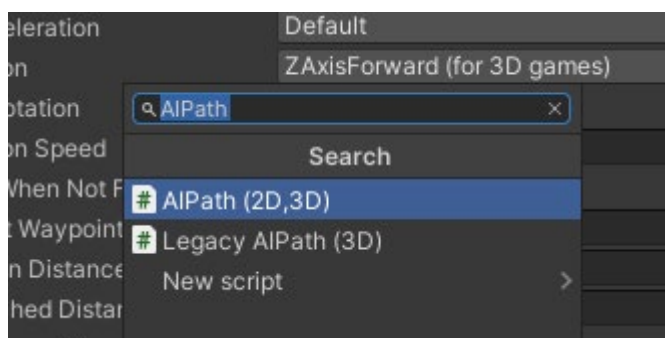


Рисунок 79. Добавление компонента “AIPath”

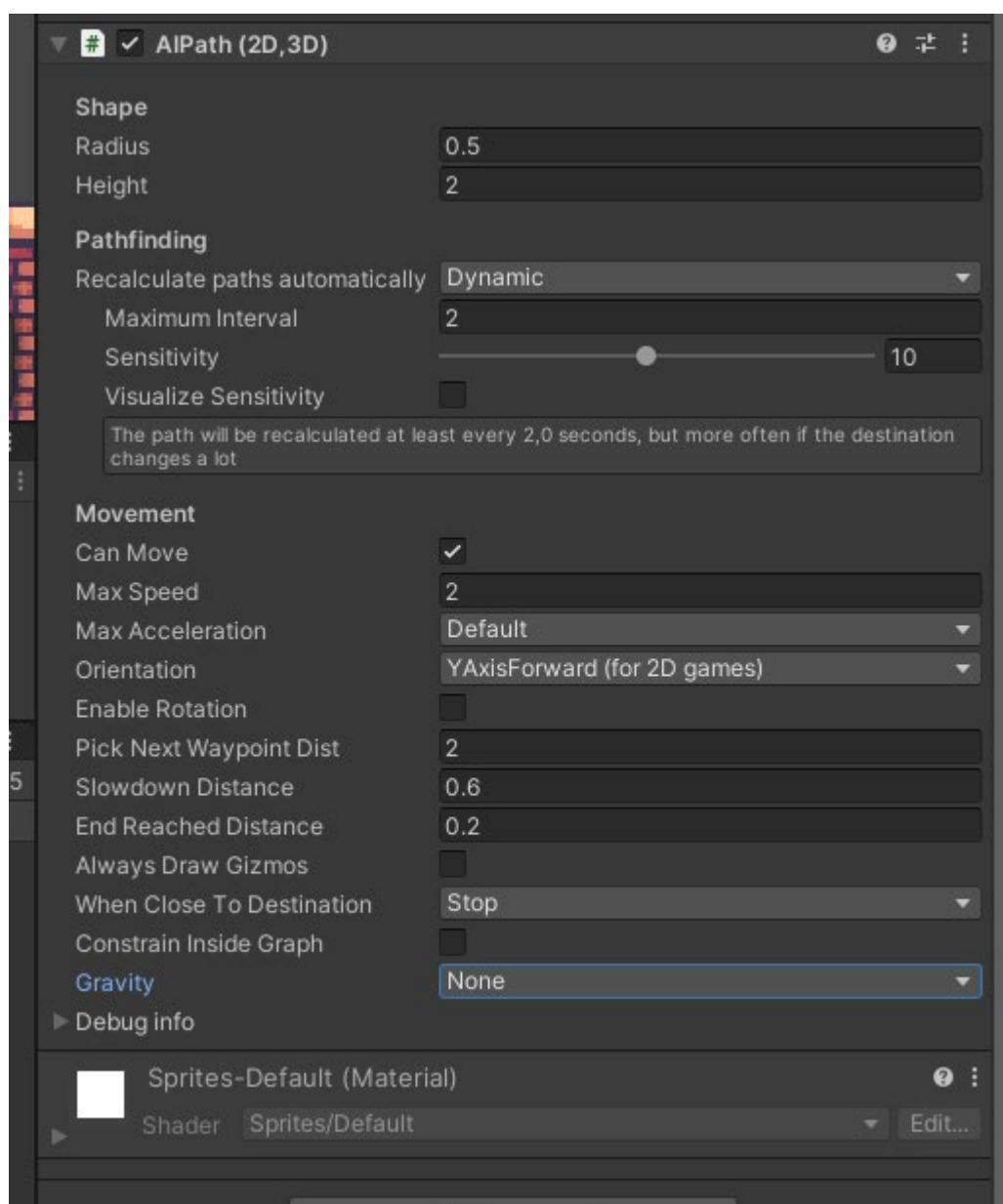


Рисунок 80. Настройки компонента AIPath.

Также после изменения параметров в скрипте “AIPath”, нужно добавить в компоненты еще один скрипт “AI Destination Setter”, который отвечает за обозначение того объекта за которым будет следовать враг. Для обозначения цели следования нужно просто перетащить объект “Player” из иерархии в параметр “Target”, который находится в скрипте “AI Destination Setter”. Ещё необходимо добавить компонент “Circle collider 2D”.

Далее нужно создать скрипт, назвать его “Mob” и добавить этот скрипт объекту врага.

Для того чтобы спрайт поворачивался в нужную сторону по направлению движения нужно написать следующий скрипт:

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using Pathfinding;

public class Mob : MonoBehaviour

{

    SpriteRenderer sprite;

    [SerializeField] private AIPath _aiPath;

    void Update()

    {

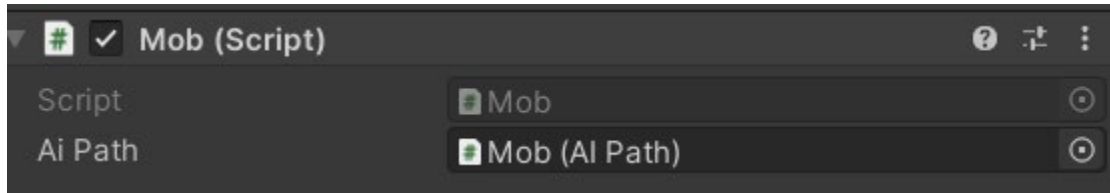
        sprite.flipX = _aiPath.desiredVelocity.x <= 0.01f;

    }

}
```

}

После написания скрипта нужно переместить объект “Mob” из сцены в параметр скрипта “AIPath” как показано на рисунке 81.



Рисунок

81. добавление объекта Mob, в параметр AI Path.

После того как все написано враг начнет следовать за персонажем и будет облетать препятствия.