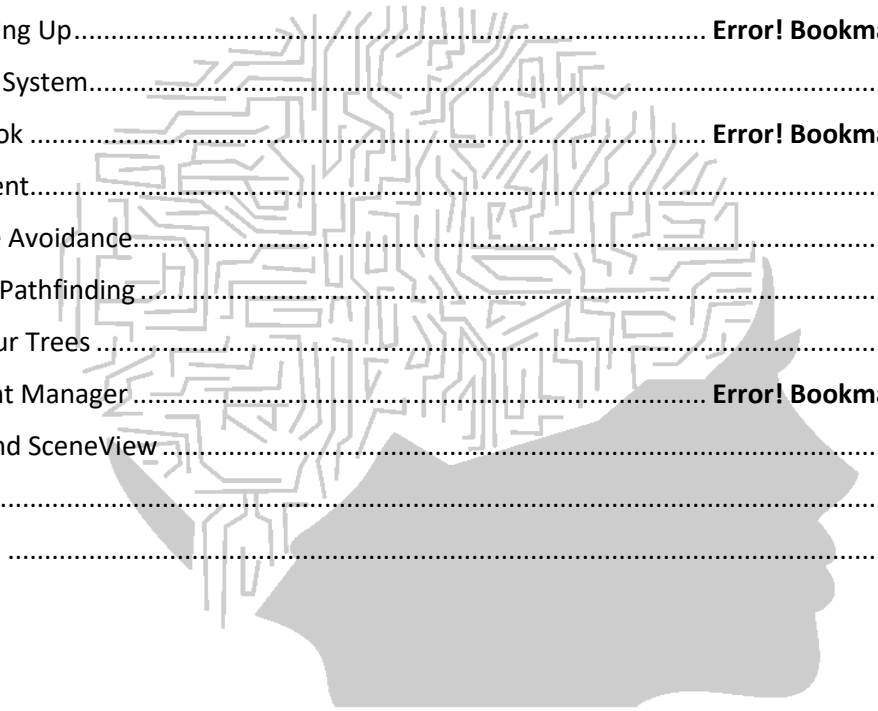
CANDICE AI

for

Games

User Manual

Contents



CANDICE

AI

Introduction

Thank you for downloading this plugin for Unity3D.

Candice AI for Games is a development tool that controls all aspects of your video game AI. You can create intelligent AI that are suitable for role playing games, racing games, tactics style games and more.

The asset also supports AI for both 2D and 3D.

Getting Started

In this system, there are three main parts.

- Candice AI Manager
    - This houses all the higher level logic, such as calculating shortest paths using Pathfinding algorithms.
    - It also handles the registration of AI agents and issues each agent a unique Agent ID. It maintains a list of all registered agents that can be accessed from other classes.
    - An instance of Candice AI Manager is required for **all agents** to function normally.
    - Only one instance of Candice AI Manager is allowed per scene.
- Candice AI Controller
    - This script controls the actual AI agents. It is equipped with **Object Detection**, **Obstacle Avoidance**, **Behavior Trees** and more.

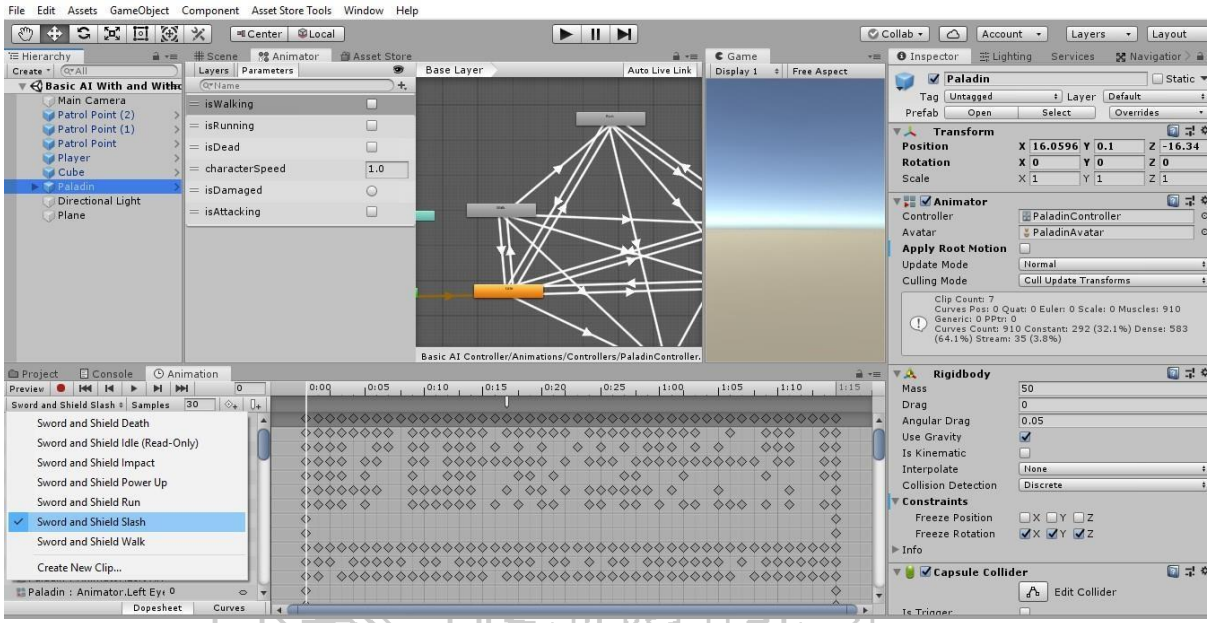
***Note: The AI Controller uses Behaviour Tress by default. So you will have to set up one of you own and attach it to the Game Object. A more detailed explanation can be found under the features section. However, a default Behavior Tree is provided that should cover most basic AI needs.**

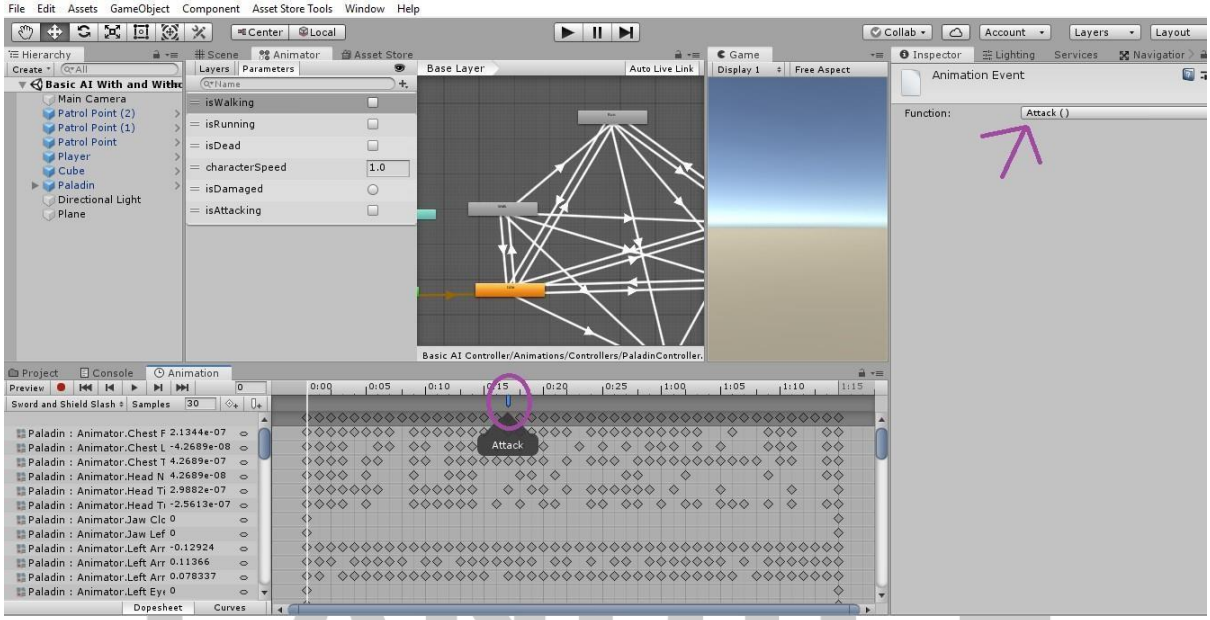
Features

Combat System

The AI unit can also attack and deal damage to objects around. You will need to use an animation Event to call the Attack() method in the script. To do so, follow these steps:

1. Select the 3D Model, in this case it is **Paladin**.
2. Select the **Animation Window**.
3. Select the Attack animation.
4. Add an animation Event at the point where you feel the damage should be dealt. Usually where the sword comes down and makes contact with the player.
5. Choose Attack() as the function to call.





Damage System

The Damage System works by sending a Raycast as a sphere around the character. If it collides with an object marked with any of the Enemy Tags, a message will be sent to the enemy object in order to receive damage. Ensure that the receiving object has a method with the name ReceiveDamage(). It should have the following method signature

Public void ReceiveDamage(float damage) { }

Movement

The AI controller supports two types of movement:

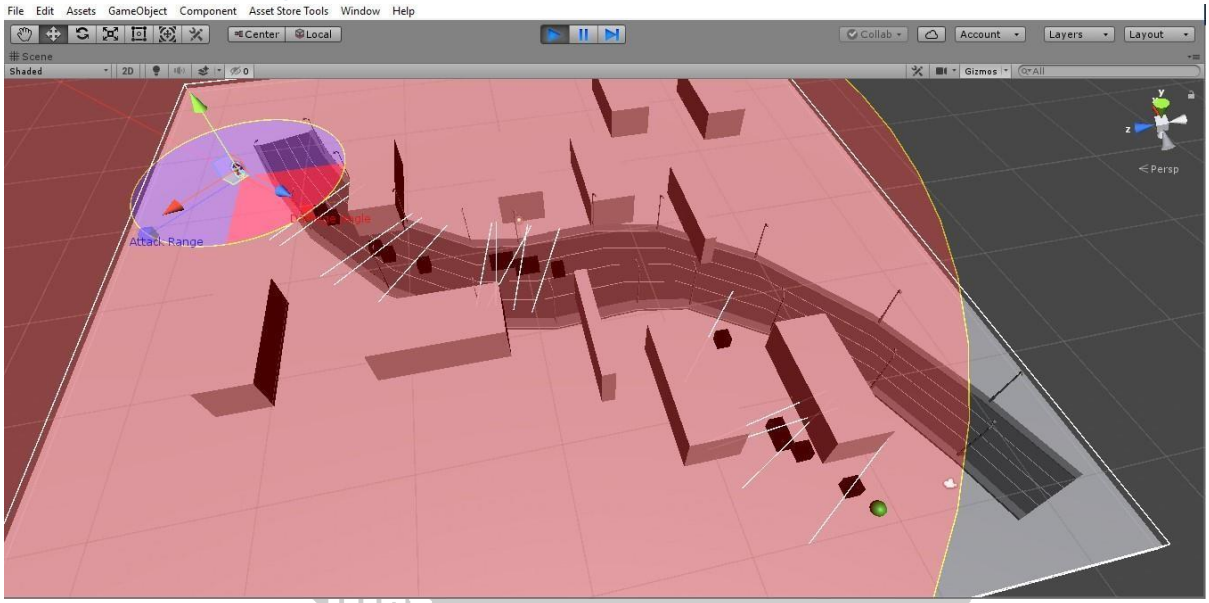
- Static
    - With this, the agent will blindly follow its target in a straight line.
- Dynamic

- With this, the agent will utilise **obstacle avoidance** and/or **Candice Pathfinding** to evade immediate obstacles while advancing to the target.

Obstacle Avoidance

This allows the AI to evade immediate obstacles, within a certain range. This algorithm is most efficient in straight line paths with a few random obstacles. However, it can also be used in more complex scenes, but efficiency is not guaranteed.

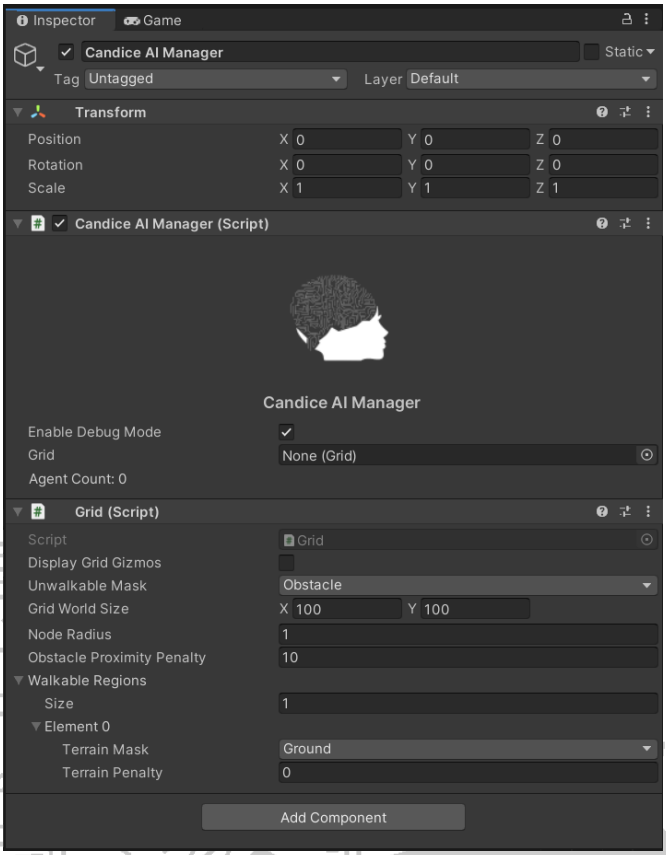
Candice Pathfinding



This system allows the agent to find the shortest path to the target.

To make it work in your own scenarios, follow these steps:

1. Ensure you have a **Candice AI Manage**r script attached to a GameObject
    a. Click on **Window -> Candice AI for Games -> AI -> Candice Setup Assistant and it will automatically create one for you, with a Grid component as well**.
2. Tweak the Grid settings to fit your terrain and define an unwalkable mask. Unwalkable mask is the Layer which the algorithm will define as unwalkable, and thus the AI will never walk on terrain/gameobjects with that Layer. (e.g a layer with the name “Obstacles”)
3. Define Walkable Regions.
    a. For each walkable region, you must specify a Terrain Mask (basically a Layer that will be assigned to a terrain/gameobject) and a movement penalty for that terrain. This allows the agent to “prefer” certain terrain types over others. E.g, it might prefer moving on the road layer as opposed to the sand layer.



Behaviour Trees

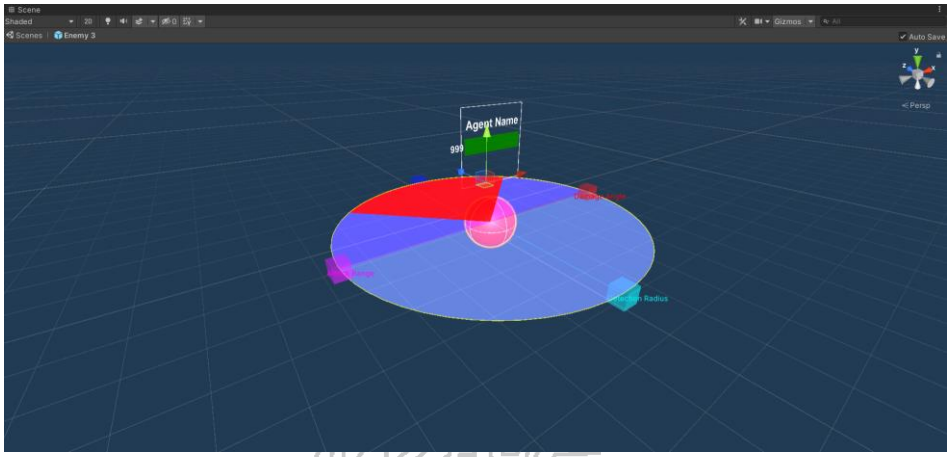
The AI Controller also supports behaviour trees. They are the default implementation if the agent is controlled by Candice.

You will have to create a script that inherits from MonoBehaviour. In the script, you will have to create your various nodes, starting from the root node. **This will require a bit of knowledge on how behaviour trees work**. A link is provided at the end explaining Behaviour Trees.
However, a default behavior tree script (**CandiceBehaviorTreeMono.cs**)is provided that covers most basic functionality. **Candice Behavior Designer™** Is a visual editor specifically designed to create Candice Behavior Trees, making the process effortless and easy.
There are three types of nodes in a behavior tree.

- Composites
- Decorators
- Tasks/Actions

Editor and SceneView

Candice AI also has powerful editor and SceneView controls. You can adjust the properties including attack range and detection radius right from the SceneView. You can also adjust all other properties, like enemy tags, move speed and more from the inspector. There's no need to touch the script if you don't want to.



Support

We at Candice AI for Games strive to provide the best AAA quality video game assets to the world. If you have any issues with this asset, suggestions, or just want to send some feedback, then feel free to contact us on one of our channels.

email: support@candiceai.co.za natubeast@gmail.com

Discord Server: https://discord.gg/GUtK6EH

Facebook: https://www.facebook.com/kandakeai/

References

Item	Author	Website
Behavior Trees		https://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php