Wander Behavior

Alexander Temnyakov

1 Introduction

This document presents a project on wander behavior. It is implemented using Unity. The agent is an alligator. The agent moves within a predefined circle area.

Figure 1: Preview.

2 Implementation

The agent uses the behavior tree[1] to make decisions and the NavMeshAgent component[2] for navigation in the scene. For animations, it uses the Animator component[3]. The use of the behavior tree is a big advantage, because it allows implementing complex things and creating extensions for the agent behavior. However, there is a space for working on the animations. They may look unrealistic due to different inconsistency in the animation speed and the movement speed.

2.1 Behavior Tree

The behavior tree consists of the root node, which is a selector, and 3 other nodes: for the idle state, to find a new destination if the current is reached, and to walk.

2.2 NavMeshAgent

For the NavMeshAgent component, a NavMesh on the terrain was created. To find a new destination point, the agent's AI creates a random suitable point and tries to sample it on the NavMesh. A screenshot of the NavMesh is shown in figure 2.

2.3 Animator

The Animator component has 4 states: the idle state and the slow speed, medium speed, and high speed walking states. The states for walking have the same animation, but different speeds. A screenshot of the Animator is shown in figure 3.

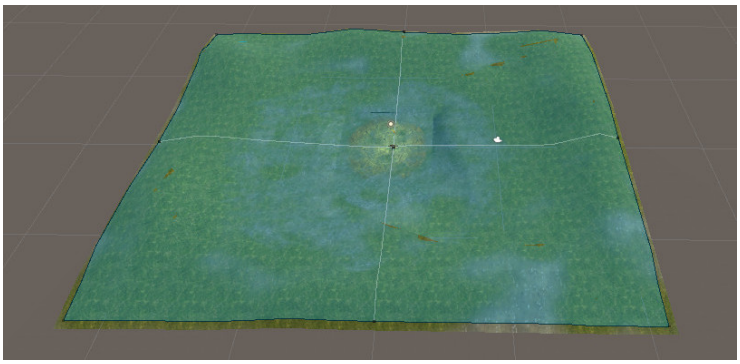

Figure 2: Screenshot of the NavMesh.

[1] https://en.wikipedia.org/wiki/Behavior_tree
[2] https://docs.unity3d.com/ScriptReference/AI.NavMeshAgent.html
[3] https://docs.unity3d.com/ScriptReference/Animator.html

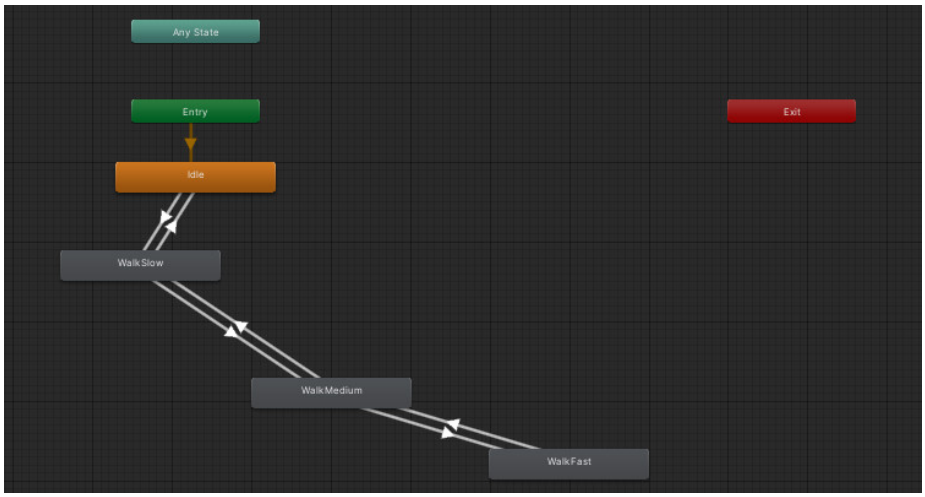

Figure 3: Screenshot of the Animator.

3 Variability

The agent is variable. It is possible to change a number of its parameters. Also, it uses randomness to choose its speed, to choose the time to be idle, and to find a new destination. A screenshot of its variable parameters is shown in figure .

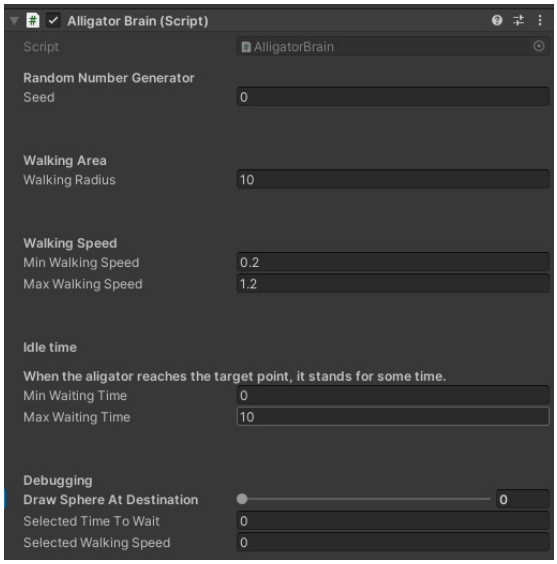

Figure 4: Agent's variability.

4 Multiple Agents

It is possible to have multiple agents in the scene. The NavMeshAgent component deals with it. The agents avoid each other.

Figure 5: Multiple agents.