TÉCNICO LISBOA

Inteligência Artificial para Jogos

2020-2021

Laboratory Guide

Lab 1 – Kinematic and Dynamic Movement

1. Explore the provided scenes

a. Start by opening the provided project in Unity. To do so, start Unity 3D and select New Project, and then select the folder "IAJ Movimento1.1".

b. Explore the two scenes created inside the folder Assets/Scenes (using Unity's explorer). Observe the existing scene objects, their properties and subcomponentes.

c. The scene *KinematicMovement* contains two characters (represented by the Blue and Green cars) that use kinematic movement to move around the environment. Likewise, the *DynamicMovement* scene contains two characters that use dynamic movement. Run both scenes and analyze the behavior generated by the two types of behavior. Can you spot the differences?

d. Study the source code that implements the kinematic and dynamic movement. The source code can be found inside the Scripts folder. The scripts *DynamicManager* and *KinematicManager* are associated to the Blue and Green car, and executed by Unity to initialize the corresponding movements, to receive the user's input and to update each character for each new frame. You can use *Visual Studio* or *MonoDevelop* to visualize and edit the source code.

2. Creating the Dynamic Wander Movement

a. The class file that implements the dynamic wander movement is not complete. Implement the *getMovement()* method according to the algorithm shown in the lectures.

Tip: The Util.MathHelper and Util.RandomHelper might help you with some functions

b. The wander movement contains a set of parameters that need to be adjusted. Try out different values for them and observe the effect that they have on the resulting behavior. Determine what are the most adequate parameter values so that the resulting behavior is as realistic as possible (i.e. it cannot appear to be too erratic, but it also shouldn't move only straight ahead).

c. To better understand and visualize the resulting algorithm behavior, create a *GameObject* with a visual representation (sphere, cube) and to use such object as the movement's debug target. Justify your choices.

3. Creating and integrating the Arrive Movement

a. Create the class *DynamicArrive* that implements the dynamic arrive movement, as specified in the lectures.

b. Add the new behaviour to *DynamicManager.* The movement's target should be the opposing character.

c. As in the previous exercise, there are a set of parameters for this type of movement. Try out different values and observe their effects on the generated behavior. Choose the most appropriate parameter values for a realistic motion.