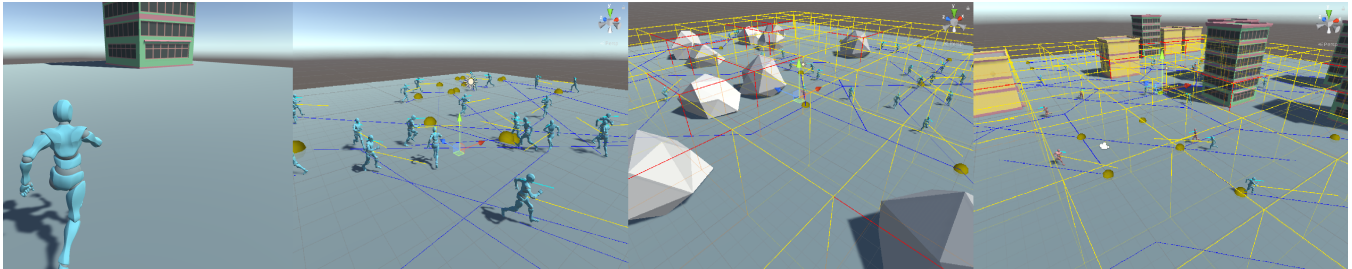


# Character Animation Report

Luis Carranza  
luis.carranza@estudiantat.upc.edu  
Universitat Politècnica de Catalunya  
Barcelona, Spain



**Figure 1.** First: Exercise 1 game view. Second: Exercise 2 scene view. Third: Exercise 3 scene view. Fourth: Exercise 4 scene view.

## 1 Introduction

The project challenges students to learn to animate characters movements and interactions with other characters and objects in a crowd simulation system. In this project, the main tasks have been split in 4 exercises.

- Exercise 1: Locomotion
- Exercise 2: Collision Detection and Avoidance between agents
- Exercise 3: Pathfinding
- Exercise 4: Steering

## 2 Setup and folder structure

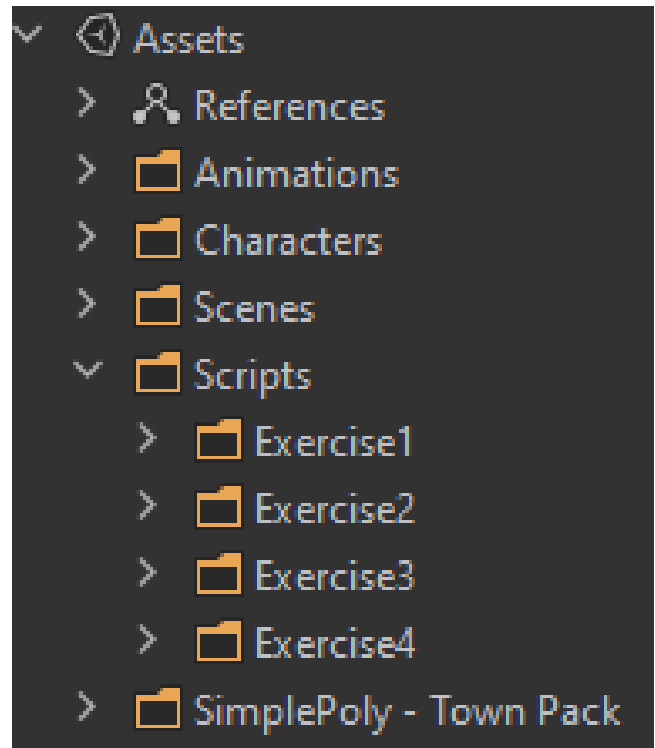
The project has been developed in Windows OS and the software pre-installed to run it is the following:

- Unity Hub
- Unity Editor Version 2020.3.28f1

As for the project structure, the deliverable consists of the folders *Assets*, *textitProjectSettings*, and *Packages*. The folder *Assets* contains the *Animation* and *Characters* folders implemented for Exercise1, the *Scripts* folder containing the script files divided in folders per exercise implemented, and the *Scenes* folder where there is one scene per exercise. There is an additional folder called *SimplePoly - Town Pack*, where some objects are used to give some references in a scene. Figure 2 shows the structure of the *Assets* folder explained before.

## 3 Exercise 1: Locomotion

To test this exercise there are two views that can be used, the game view to see the animation as a third person, or the scene view to play with the parameters of the agent from the inspector tab. As for the game view mode, Figure 3



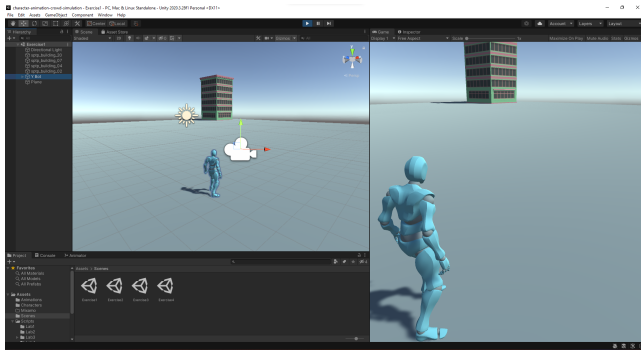
**Figure 2.** Folder Structure in the Assets folder.

shows a way to visualize the exercise. The options to test the animation in this mode are the following:

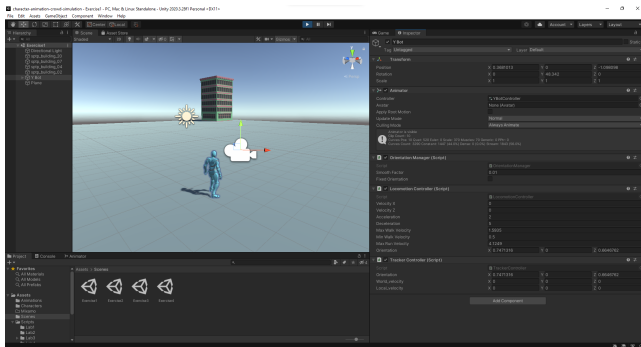
- WASD keys: For character movement among the scene.
- Shift-Left: To enable character running while pressed.
- Ctrl-Left key: To enable FixedOrientation.

- Alt-Left key: To disable FixedOrientation. It's implemented in this way so it is possible to edit this parameter in the scene mode.

As for the scene view mode, Figure 4 shows a way to visualize the exercise. The inspector tab is showing the properties of *Y Bot*, which in combination with 3D movement tool of the character in the scene allows to see how the animation behaves under more abrupt changes.



**Figure 3.** Unity Editor recommended window structure for game view mode in Exercise 1.



**Figure 4.** Unity Editor recommended window structure for scene view mode in Exercise 1.

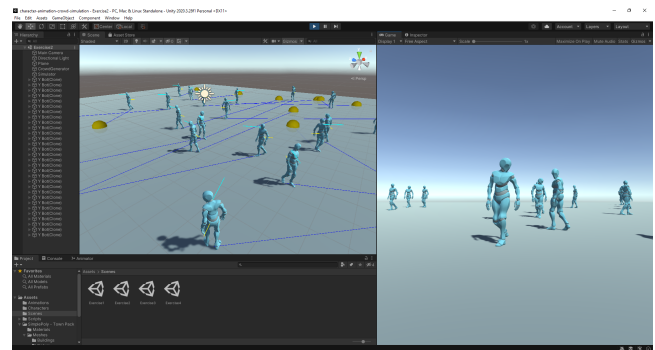
## 4 Exercise 2: Collision Detection and Avoidance between agents

Figure 5 shows the expected results when running the scene. Similarly as in Exercise 1, it is possible to enable/disable fixedOrientation either with Ctrl-Left or Alt-Left keys in game view mode and in inspector properties of the character before running the scene, or after running by enabling each of the characters in the inspector tab. It is also possible to use the WASD and Shift-Left keys, but the rotation is not properly handled for every animation frame position of the character, so I recommend to enable fixedRotation before doing so (otherwise some characters could start walking in the air or below the plane). Figure 6 shows an example of how

the WASD+Ctrl-Left keys can also perform an action in the characters, for testing purposes. The parameters expected to be modified for this exercise lies within the CrowdGenerator GameObject, in the Crowd Generator Script Component.



**Figure 5.** Expected result for Exercise 2.



**Figure 6.** Ctrl-Left + S keys used in Exercise 2.

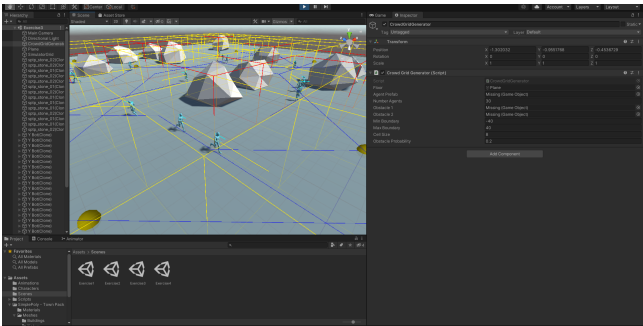
## 5 Exercise 3: Pathfinding

The pathfinding algorithm implemented is A\*. In this exercise, the parameters expected to be modified lies within the CrowdGridGenerator GameObject, in the Crowd Grid Generator Script Component. It is expected to place only one agent per GridCell at the start of the program, so it is advised to always set a number of agents below the number of GridCells (Maxboundary-Minboundary/Cell Size). When instantiating the obstacles, I implemented a condition of not creating them in the boundaries in order to try to have always a possible path available for the agents. Figure 7 shows the expected results when running the scene.

## 6 Exercise 4: Steering

The following steering forces were implemented and used for this exercise:

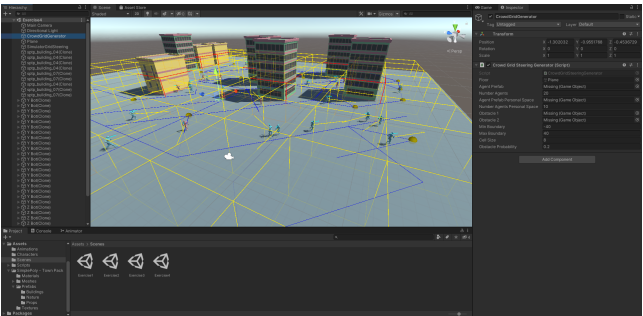
- Seek Force
- Arrive Force
- Collision Avoidance Force



**Figure 7.** Expected result for Exercise 3.

Figure 8 shows the expected results when running the scene. In this exercise, the parameters expected to be modified lies within the CrowdGridGenerator GameObject, in

the Crowd Grid Generator Script Component, and in the Y Bot and Z Bot Agent Component.



**Figure 8.** Expected result for Exercise 4.