## Project 2

Team no 7

Pratik Shringarpure

Yiyan Wang

Ollyting Xin

* How to start the environment?

In the project folder, there is folder named “\_builds” which contains the .exe file. Open the .exe file named **Project 2.exe** to start the environment.

* How to run the A\* demonstration?

The A\* demonstration is set active when the game is launched. Initially, the red cube will be moving to a pre-determined position in the map. You can change the destination position of the red cube by using the mouse and clicking on the terrain in map. The a\* algorithm will use the mouse clicked position on the map, and if it is on walkable area, algorithm will calculate the path to that position and make the red cube move towards it.

* How to run the flocking demonstration?

The flocking demonstration is set active when the game is launched. The small flock of cubes that follow the red cube use the flocking algorithm.

* How to alter flocking parameters?

To alter the flocking parameters here are the instructions:

PRESS I TO DISABLE COHESION

-PRESS J TO ALLOW COHESION

-PRESS O TO DISABLE ALIGNMENT

-PRESS K TO ALLOW ALIGNMENT

-PRESS P TO DISABLE SEPARATION

-PRESS L TO ALLOW SEPARATION

-PRESS R TO RESTORE ALL THE PARAMETERS OF THE FLOCKING

* Bottleneck Issue

For the first time we implemented the flocking into the game environment, they could fly through the edge of the bridge without any collision detection. So, we tried to add a rigid body component to the flocking members/agents, but it still didn't solve the problem of the collision detection. Finally, we found out that the problem happened because that we define the movement of the flocking members by changing their position directly. Since the Unity physics engine won't work when the game object's position changed directly by adding vectors to them, we turned to use another way to define the movement of the flocking. We used the AddForce() method which is a method included in the Unity physics engine. It can change the velocity/speed of the flocking member by applying a force vector to the game object. We get the force vector by calculating the position between the flocking members, between the member and the leader game object which is the Seeker using A\* pathfinding method moving toward to a target in the game environment, and between the member and obstacles in the environment. With these changes, our flocking members can go through the bottleneck without flying away from the bridge. However, because the collision detection is discrete and called every frame when the speed of a flocking member is so fast, it sometimes can't detect the collision because it passes through the edge so fast during the frames. We tried to avoid this problem by slow down the max speed of the flocking member as well as the leader of the flocking.