# AI Escape

## Project Summary

### *Objectives*

The objectives of this game are for the player to avoid the AI NPC that is chasing them, for as long as possible. When the AI gets close enough to the player, the game is over.

### *Supplied Resources*

You have been provided with the Unity project files for the game which includes the scene file for the game as well as some of the code for the core gameplay functionality. You will need to read through the provided scripts to identify the missing functionality, as well as the appropriate places to implement the missing functionality.

## Gameplay

### *Core Mechanics*

#### ***Player***

The player needs to be able to move their character between nodes by using the appropriate input controls. While the player's character is moving from one node to another, it should not respond to any inputs.

#### ***Artificial intelligence***

The AI NPC will start by moving to the first node in the static list of nodes, before using the depth-first search algorithm to determine which node the player has moved to. Whenever the AI reaches its current target node, it will use the DFS algorithm to find a new target node (the current node of the player). When the AI collides with the player, the end of the round is triggered and the game is reset.

## Specifications

### *Platform*

A gold-master build that is compatible with windows will need to be built. A web-compatible build will also need to be constructed and tested across the Google Chrome and Mozilla Firefox browsers.

### *User-Interface*

#### ***Control Scheme UI Widget***

The UI for the game will need to be setup to use a resolution of 1920x1080, and will need to scale with both the screen width and height.

A UI widget that displays the movement directions will need to be integrated.

Whenever an appropriate input is pressed, the corresponding direction on the UI widget will need to briefly flash green to indicate that input was selected. If the direction corresponding with the selected input is not available, the corresponding direction of the UI widget should flash red instead.

Each movement direction on the UI widget will also need to be capable of being clicked with the left mouse button in order to move the player's character in the corresponding direction.

***Control Scheme***

The player will be able to select the direction that their avatar moves in using the keyboard, mouse, or a controller input. Each arrow key on the keyboard will need to correspond with the appropriate movement direction (up arrow – north, down arrow – south, left arrow – west, right arrow – east). You are able to determine an appropriate control scheme that is suitable to the type of gamepad controller you have selected. The player should also be able to mouse click directly on each direction displayed in the widget to move their avatar.