**Grid Game Instructions**

The primary goal of this assignment is to understand the Command pattern and how it can be useful.

The following three steps apply to the command pattern.

1. In the Moves.cs file, under Scripts/Commands, create the following classes that inherit from Command (see Scripts/Commands/Command.cs. These are short classes so you can put them all in the same file (Moves.cs). In each of these classes you need to override the CanExecute and Execute methods. See the notes at the top of Moves.cs.
   1. MoveLeft
   2. MoveRight
   3. MoveUp
   4. MoveDown
2. After Step 1 is completed uncomment the commented-out section in CubeMover.cs in the Move\_Performed method (Lines 81 – 94). If you created Step 1 correctly then you should be able to build the project. When you run the game the keystrokes will move the player across the board subject to the rules(ASDW).
3. After Step 2 is completed then uncomment the commented-out section in MenuSystem.cs in the UpdateNavButtons method. (Lines 222-238). If you created Step 1 correctly then you should be able to build the project. We you run the game you will see that the Up/Down/Left/Right buttons with enable/disable as expected.

Additional steps apply to the User Interface

1. With each move increment the value for the numMoves field in the UpdateFields method.
   1. Use the value property of the \_\_numMovesTextField class variable.
   2. There is a class variable, numMoves, which is an int, that holds the current move.
2. Complete the Destination class.
   1. Use the LandingAreaDetector class in the TimeToImpact-WithCustomPlanets project as an example.
   2. The OnTriggerEnter in this class needs to Publish the event that will trigger GameOver in the MenuSystem class. Look at the MenuSystem class to find which event is subscribed to fire GameOver.
3. In the GameOver method let the player know that they have won the Game.
   1. Use the \_gameStatus class variable.

The rules of the game: The Player Character can move in any direction, once, except where there is a barrier. Once the player visits a square, they cannot revisit the same square again in the same game round. The goal is to reach the destination with the fewest number of moves.

## END OF INSTRUCTIONS