This is the Readme File for the project

Author: Sandeep Saha
Purpose: Code Test for Sprung Studios
Application: Unity 5.6.1

Project Name: New Unity Project 4
Application: Unity
Type: Interface Design
Description: A dynamic grid system where user can move items from one grid position to other. The movement has two different modes: Swap and ReOrder. The Swap feature will exchange the position between the two different elements of the grid. The ReOrder feature will shift the places of other elements of the grid upon the switch of the target and final elemental positions. The complete grid system can be also dynamically changed according to its width and height sizes i.e. dynamic row and column numbers.

Folders: Folder named "Assets" holds all the subfolders for the required project. The subfolders are:

1. Prefabs: This folder contains all the prefab files. These prefab files are the game elements that majorly contains all the interactive game elements excluding the UI items.

2. Scenes: This folder holds all the scenes of the project. For this particular project, there is only one scene file called: Main.unity.

3. Sprites: This folder contains all the graphical and image elements that has been used in the entire project.

4. Scripts: This folder contains all the required scripts files written in C# language. More information about each scripts are:

i. Board.cs: This script controls the major elements of this project. From setting of grids and different elements on the screen to all the interactions.

ii. BoardShuffler.cs: This script file is responsible for shuffling the board when the user hits the "Random" UI button on the screen.

iii. GamePiece.cs: This script describes all the grids that the player can actually see visibly and interact in the runtime. This is responsible for movement of the visual grids on the screen according to the given conditions.

iv. Restarting.cs: This is a small script only functions to restart the board completely from the beginning of the start.

v. Tile.cs: This script controls the placements and changing of invisible grids and positions for the GamePiece.cs scripted elements and also control the mouse interactions on the grid.

HOW I APPROACHED THE PROBLEM (BEHIND THE SCENES):
Broke down the different elements that were required, and worked on them modularly:

i. Making of grid systems

ii. Placing game objects that will be controlled by the users i.e. the visual tiles.

iii. Created a simple swapping mechanism with static number of grid dimension like the match3 games.

iv. Created the reorder mechanism with static grid dimensions like the reorder of app icons on iPhone.

v. Created a random generator where the random generator will fill up the tiles randomly with existing elements on the board that are currently present.

vi. Created a mechanism to change the dimensions of grid in real time i.e. at runtime without restarting the project.

vii. Attached slider to control the dynamic grid dimension.

viii. Created a system to make the swapping and reordering mechanism work with the changing grid dimension.

ix. Created a pattern to add and remove game objects according to the changing grid dimensions i.e. With the increase in dimension, the elements will be added and decrease in dimension will remove it.

x. Camera system also setup to keep the overall grid/tile dimension to fit the screen.

NOTES:

1. The code is commented in many regions for better understanding. If there is any confusion and or need any clarifications, contact me via email: sandyloisa@gmail.com specifying the method name or line number(s) of any particular script.
2. Assumption has been made to have minimum and maximum grid because of slider interactions which limits it other way.
3. The restart button is provided for internal check only to reduce the effort of restarting the project from the IDE while testing and producing.
4. The UI elements and all other visual elements in the project has been made in order to replicate the given pitch video as much as possible but due to time constraints and other priorities, it might not be up to exact replica.
5. The time of production of this project approximately: 3 working days.
6. There are some unused methods, functions and variables in the code/script files which is required to be omitted.