Unity Developer Test

Flower Harvest

By: Jonathan Major

# Scenes

This game is built with 3 different scenes.

* **StartScreen**

A simple scene that introduce the player to the game. One Startbutton to go to the next scene.

* **Levels**

A scene that holds every level in the game in runtime. Every level (buttonObject) generates through the script. A button leads to the gameplay. A scrollrect object helps us navigate to all levels in a easy way.

* **Gameplay**

Holds all the game mechanics which eventually leads back to the Level scene.

# Specific Objects

* **GameManager**

The GameManager is a singleton object which holds data between the scenes. Level progress is saved under runtime. It also holds the current level to be played after a button click on the chosen level. I have excluded to use PlayerPrefs saving for this project at this moment.

* **SoundManager**

The SoundManager is a singleton object which holds all the AudioSource’s needed for the game. In the Inspector, you can change volume for music and sounds. You can also add audio clips of your choice.

* **StartScreenLogic** (StartScreen)

This simple object holds a function for the Startbutton.

* **LevelMakerLogic** (Levels)

This object holds everything needed to generate level-buttons in a proper way. As well as the function for the buttons. In the Inspector, you choose the amount of levels by setting a size. Every element represents a level which has four options for now. Width, Height, Moves and Scoregoal.

* **BoardLogic** (Gameplay)

This object holds all the game mechanics. In the Inspector you add object types (Flower Object) and object background (Background to the tile), and various texts for the UI.

# Game flow

The gameplay is based on states. A state machine that make the game follow different phases.

**Begin** ->   
(Begin state run only once)

**Wait for Input** -> **Destroy Objects** -> **Move down Objects** -> **Spawn missing Object** ->   
(As long as the score is not reached and moves are left, gameplay rotates within these states)

**Reshuffle** ->  
(Reshuffle only runs from the “Wait for Input” state if needed)

**Complete** / **Failed**  
(Complete or Failed state run only once)

# Game mechanics goals

The board consists of at least 5 different colored tiles and may be of various sizes (5x5, 7x7, 6x9 etc.)  
(5 different types of flowers and a board with 5x5 – 9x9 and all in between.)  
  
Tiles are matched by linking adjacent tiles of the same color  
(Same type are matched.)  
  
Tiles can be linked horizontally, vertically or diagonally  
(Match types by moving horizontally, vertically or diagonally.)  
  
At least 3 tiles are required to make a match  
(3 flowers at least to make a match.)  
  
Matched tiles are removed from the board  
(Removing flowers that has been matched)  
  
The board collapses downward on empty tiles and new tiles fall in from the top  
(After a match, flowers drop down to spaces below and new flowers spawn from the top.)  
  
The board reshuffles the tiles if there are no valid links available to make  
(Reshuffle a board which has no valid match possibilities.)  
  
The link in progress can be undone by swiping back to the previous tile  
(Swiping back will reduce the linked flowers. But only in the same order.)

# Submission Requirements

Goal-based levels (e.g. earn a certain number of points, completing within a certain number of moves etc.)  
(Earning 10p with every flower. Moves that has not been used give 100p each. Every level has a limit set in the Inspector.)  
  
At least 2 levels to play  
(I have set 10 levels in the Inspector, but feel free to add more levels though the “Levels Scene” at the LevelMakerLogic object.)  
  
A level editor to make updates without code changes  
(LevelMakerLogic, SoundManager. BoardLogic)  
  
Unity Editor data hooks to manipulate the game and feeling (e.g. animation curves)  
(SoundManager, BoardLogic and the LevelMakerLogic)  
  
A main menu and level selection screen with navigation between them  
(StartScreen, Levels, Gameplay with navigation)

# Notable

Camera settings are only based on my Windows and Android Phone tests.  
Camera and Zoom level for each board size is manually tested and set through the script and not customizable any other way for now.  
The Canvas is now based on scaling with the width of the screen.  
The Windows and Android Platform have been tested. Support for Mouse and Touch are implemented.  
Graphics are designed in Inkscape by me.  
Music is from OpenGameArt.org (Credits to: "World Map" by Scott Elliott).  
Hopefully you will like the Notes written in the code.