Development Roadmap

Open with Unity 2022.3.20f1. Project is created with 3d URP template.  
  
Use "Assets/Scenes/StartUp" scene to start the game. There is 5 unique level to play. After 5 the looping system is working.

# Day 1

Started with a game template that i used for game. İt contains level management, currency management, pooling and some other systems.

At first i layout the scene positions with dummy objects and create the sample level hierarchy. After this i started the create needed scripts for controlling all objets in the scene. I created an parent-child related script order for grid and match mechanics.

MatchGroup.cs -> SingleGroup.cs -> QueueObject.cs

This 3 script is handling operations for object type control, match control and empty spaces check. At this point i need some drag-drop and grid position features. I use “Flexalon: 3D Grid Layout” package for basic line grid and drag-drop movement.

Most of the match mechanic foundation is layed at this point. So i found some 3d object models and categorized them with scriptable objects so i can easly create more and set at other scripts.

Now i need some levels to test this system. So i planned the future development and created a simple semi automatic level creation tool. I used prefabs to level progression.

# Day 2

I started day with some fixes for Flexalon package. Objects are not moving the way i want so i changed some behaviors. I use flexalon package’s drag drop feature and create a system to work on drops so objects can change place at level. At this change i check front line for “all empty” and “all same” condition so i can make match pops and destroy front line place holders. Deleting the place holders automaticly trigger the Flexalon system to move objects front so there is no need for a new system here. Because of the this movement is automated i dont have a point to change material for front lines at these stage. But i figured out i dont need it and created a shader graph that change color behind Z axis. At this point i have a grid system, item matching, item sorting and cascading effects for the game. So i connected end game operations and win condition.

# Day 3

Started with game lose condition. Created the ui timer and time mechanics. Created some more objects and levels. Star collection system created with particles. Combo system created. Earned star count connected to combo system. At dragging system i found a bug and fix it. Some new sounds added. Group blocking mechanic at level 5 made. General testing and bug fixing.

# Day 4

Draggable area expanded for easy drag. All first lines can be filled with objects and game is locking in that state. Check for this problem and lose condition added. General polishing and particle changes made. After this polishing stage some of the wait systems cause a false win/lose conditions. I used async delay to fix this problems.

At this stage i have time for creating power up system but its not mentioned on the case study document so i did not create the system.

# DAY 5

I made overall testing and fixes. Changed some structural code dynamic for optimizing the game. Fixing and changing some dependecy problems. This is the first time i used the dependecy injection pattern so there can be other problems. I hope i will get better by time.