

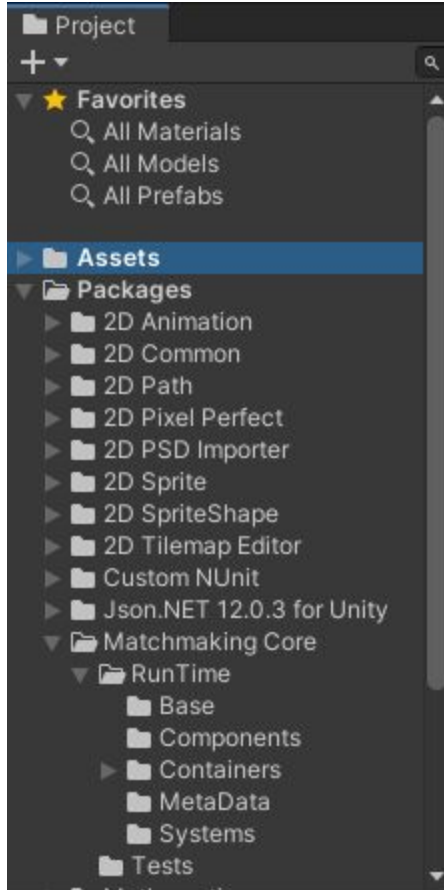
Project Structure

Project contains 2 parts. Unity and pure C#.

- Unity project can be opened by adding the project root folder to the Unity Hub.
- C# project can be found at */MatchMakingCore/MatchMakingCore.sln*

Both Unity and C# projects share the same code. Runtime code can be found inside */MatchMakingCore/Core/RunTime* folder.

Unity access Matchmaking code through Unity Package Manager.



About the Project

This project attempts to use Data Oriented Design instead of Object Oriented Design. It has 4 main parts:

- Component:
Components are the actual data, it will be processed, merged, created or removed by Systems.
- System
Systems are the actual logic, it will take in components and modify, merge or remove them.
- Entity
Entity is an abstract concept, one Entity can have multiple components attached to it. In the code, it is just an integer id.
- Container
Container is the center location for access to all the entities and components.

Code Generation:

Under */MatchMakingCore/Core/RunTime/Containers/__generatedComponent__* are all the files supposed to be generated by the code generator. (due to the time restriction, it is not generated by the code generator right now.)
Whenever a new component or component comparer is added, partial class on container will be created for caching and data access.

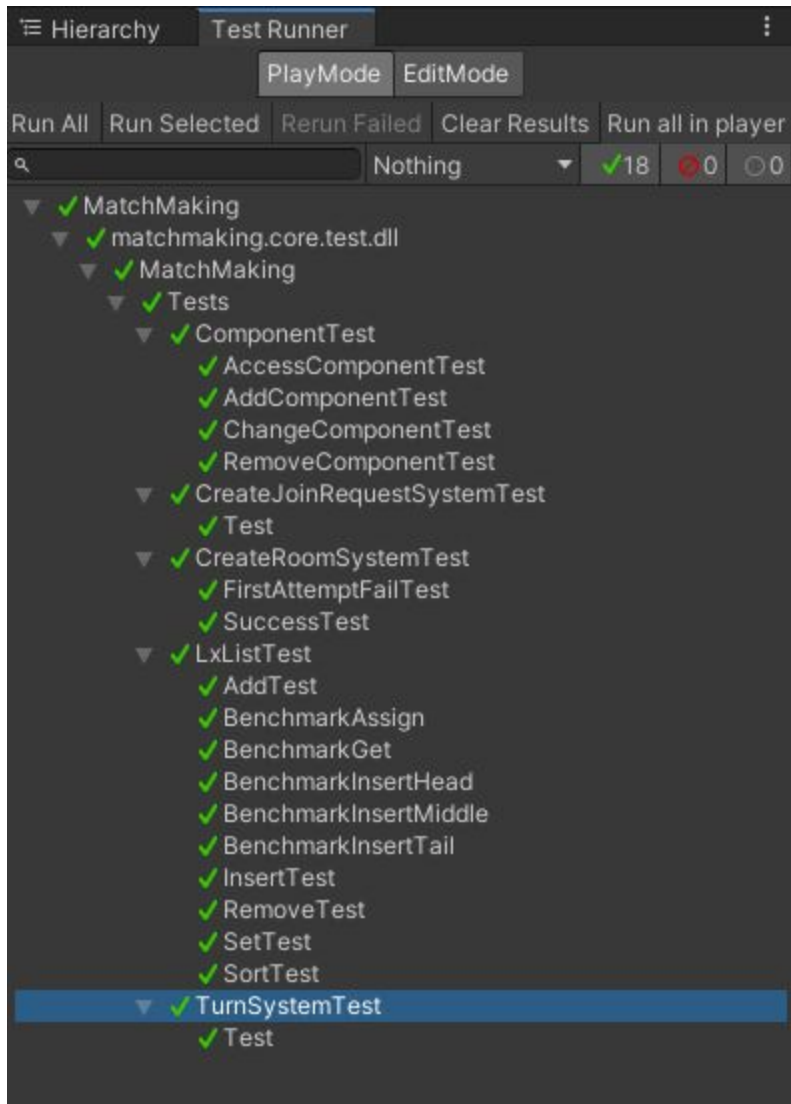
Matchmaking Implementation:

- Matchmaking configuration:
Matchmaking configuration file can be found at */Assets/Playerdata/matchmakin-config.json*.
 - PlayerPerTeam: number of players per team. E.g. 3 means 3vs3.
 - LoseWeight: the weight for the player's loses count.
 - WinWeight: the weight for the player's wins count.
 - MaxDifferenceAllowed: the max difference between players' mmr allowed to be in the same match.
 - WaitBonusWeight: the weight offset for the player who is in the queue for multiple matchmaking rounds.
- Formular:
 - Player weight: $\text{player wins} * \text{WinWeight} - \text{player loses} * \text{LoseWeight}$
 - Match check:
 $\text{player A weight} - \text{player A waiting rounds} * \text{WaitBonusWeight} - \text{player B weight} - \text{player B waiting rounds} * \text{WaitBonusWeight} < \text{MaxDifferenceAllowed}$
If true, a match is found.

Tests:

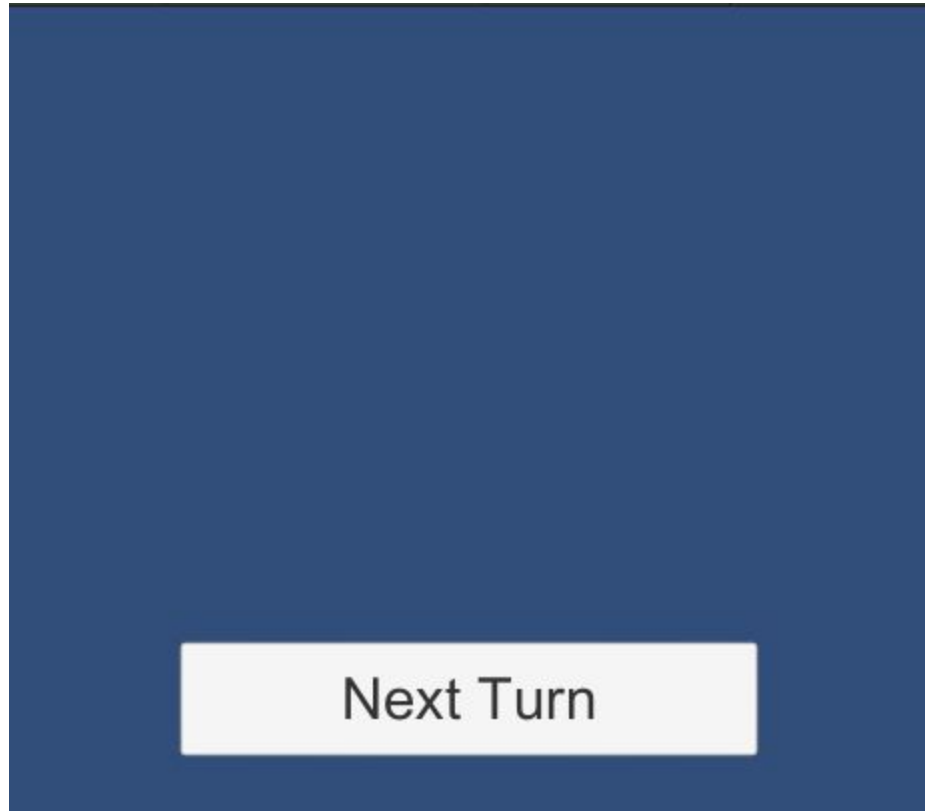
Tests are created using Unity Unit Testing.

Goto Unity Editor -> Windows-> General -> Test Runner



How to run:

- Unity Project:
 1. Goto Unity Editor
 2. Open Assets/Scenes/Matchmaking.unity
 3. Run the Scene.



4. Press “Next Turn” button to generate matches.
5. Matches will be shown in the Unity Editor Console window.



- C# project:
 1. Run */MatchMakingCore/bin/Debug/MatchMakingCore.exe*
 2. Enter "next" or "n" to search for more matches.
 3. Enter "quit" or "q" to exit.