CS 319 Object-Oriented Software Engineering

Spring 2019

Final Report

My Little Quadrillion - Group 1C

- Alemdar Salmoor 21500430
- Ece Çanga 21600851
- Erkin Beşer 21400961
- Gökçe Sefa 21400596
- Mustafa Azyoksul 21501426
- Talha Şen 21502663

Instructor - Eray Tüzün

1. Introduction: State of the Implementation

My Little Quadrillion game is developed in IntelliJ IDE with using JavaFX Library and Scene Builder tool for both design and connection of the core classes with methods. We used Github to share our works and the project is named as “Quadrillion”. Mustafa created the Github area for our project. Our project has lots of classes and images, so integrating became easier. Otherwise, we could not merge them into a single project. We had both face to face and online meetings. Online meetings were via Google Drive, Discord and Whatsapp.

Implementation of the work is equally distributed among the group members and each of us took part in different aspects such as core mechanics, controllers, model classes and database etc. We firstly started to implement GUI parts and game mechanics. We didn’t make many changes in game mechanics but there had been some more changes in user interface such as style changes applied to create appealing view. Even though we distributed the tasks equally, we had also dealt with each others’ works. Mainly, Mustafa was working on the core logic of the game, Ece and Gökçe were working on the interface of the game and integrating pages with the core game. Erkin was working on the new game features of counter, timer and theme, Talha was working on database system of login, register and saved created games, Alemdar was working on the settings, tutorial, color-blind mode and conflicts that caused in git while merging the game to the master branch. The communication between the members was the key in developing our game in the most efficient way.

In general, we have completed all of the promised additional and functional requirements and features.

2. Design changes

Generally, our first design of the game did not include the FXML files that were prepared using JavaFX Scene Builder. There were Scene classes in the visualization of the game. After preparing the FXML files, additional CSS were added so as to customize the themes up to the small details. There weren’t any changes in the core design implementation.

2.1. Timer & Counter Improvements

My Little Quadrillion game has timer and counter in the first iteration of the report. We finalize the timer scene and counter scene left and right part of the game to keep elapsed time, remaining time, move count and remaining move, depends on game mode, to make feel player competitiveness.

2.2. Game Level Improvements

There are 4 hardship levels in the Arcade Game scene and these levels have also 5 different levels as increasing hardship in the game. So we try to emphasise on creating various difficulty to welcome all types of players.

2.3. Creating new Game

Players can create a new game with existed pieces and grids. These grid layouts and the pieces location are designed as a unique game. This created games are saved into the level editor page with database system.

3. Lessons Learnt

All of the lessons learnt in the Iteration 1 are also applicable to Iteration 2. To reiterate:

From technical knowledge perspective, we learned git version control system and some of its commands such as init, add, commit, log, show. Since we use GitHub as a platform for our projects in CS 319, we also learned commands associated with remote version control such as push and pull. All of this opened a new perspective on software development. We became more proficient with JavaFx compared to Iteration 1.

We also gained a lot of experience in terms of working in the groups and coordinate each other to achieve a common goal. Sometimes, one person can not agree with himself but in the groups you have to make sure that 5-6 people agree on the point to proceed with the project. Therefore everyone improved in terms of compromising for the greater good.

Additionally, we became even more proficient with Github and Git compared to Iteration 1.
We learned that we should have discussed the implementation details in advance before jumping to our respective sections. This caused a lot of conflicts when the time of branch merges came.

4. User’s Guide

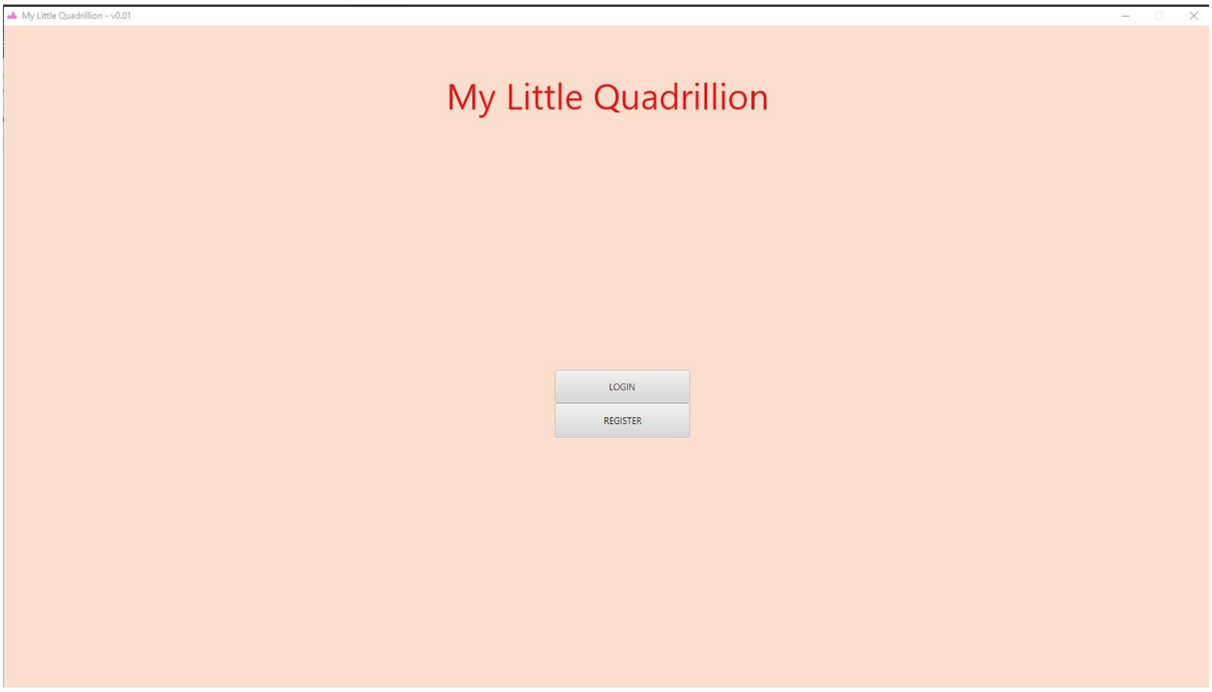
4.1 System requirements & Installation

- A device that runs Java Runtime Environment.
- ~100 MB of free storage space
- 256 MB of volatile memory

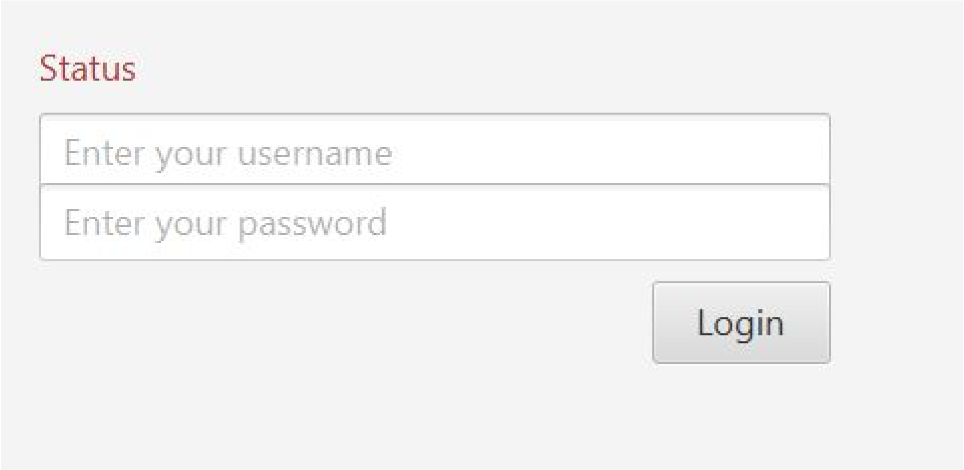
If Java Runtime Environment is not installed on the system, the user can install it from official Java download page: https://www.java.com/tr/download

The game does not require installation process. Executable file is obtained from the source code that can be run by double clicking on the file.

4.2 How to Use

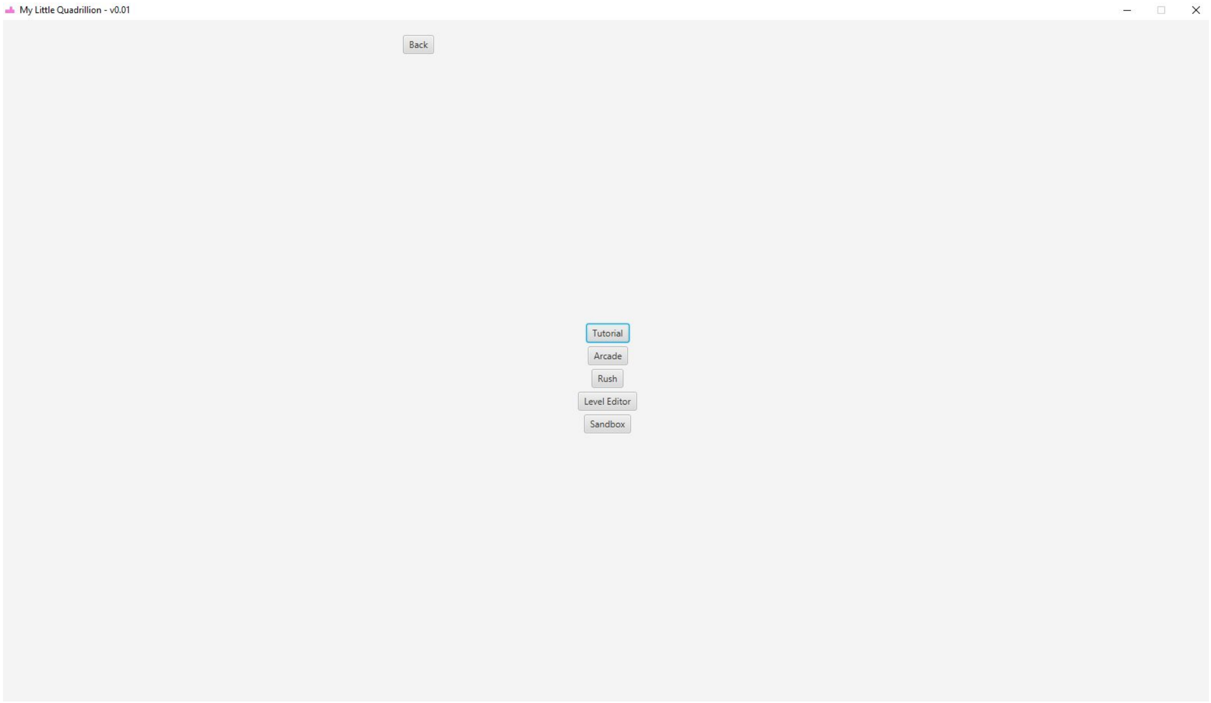


Log In



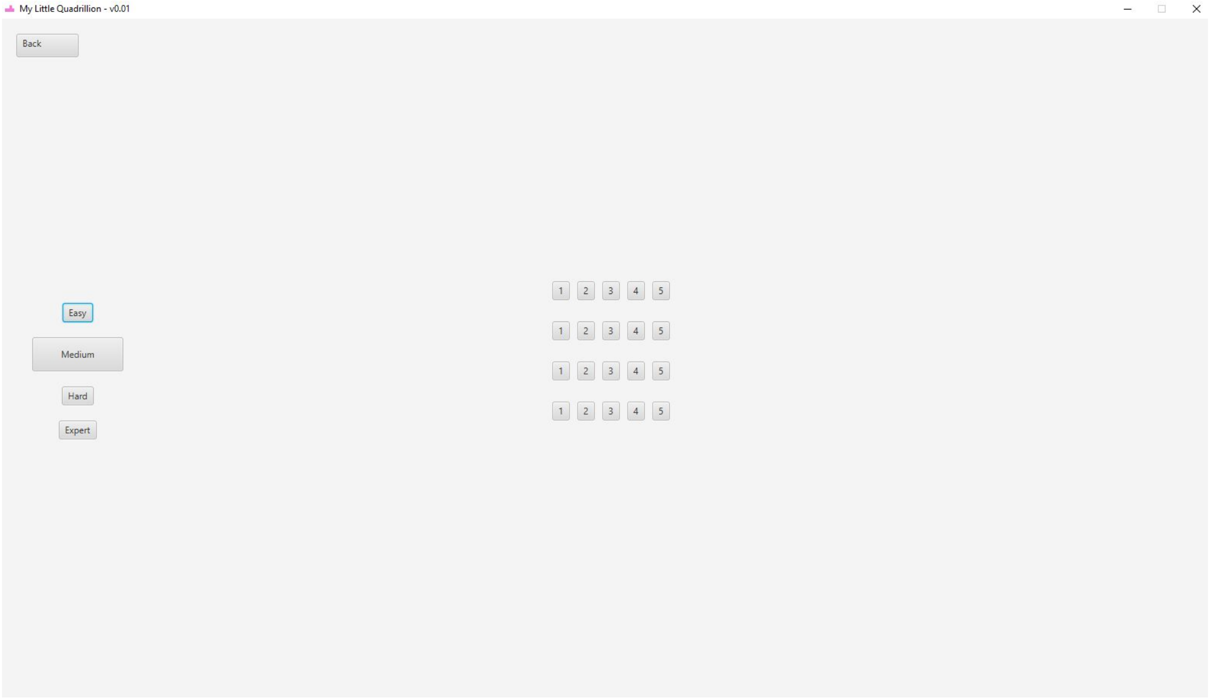
Player can log in through login page.

Main Menu

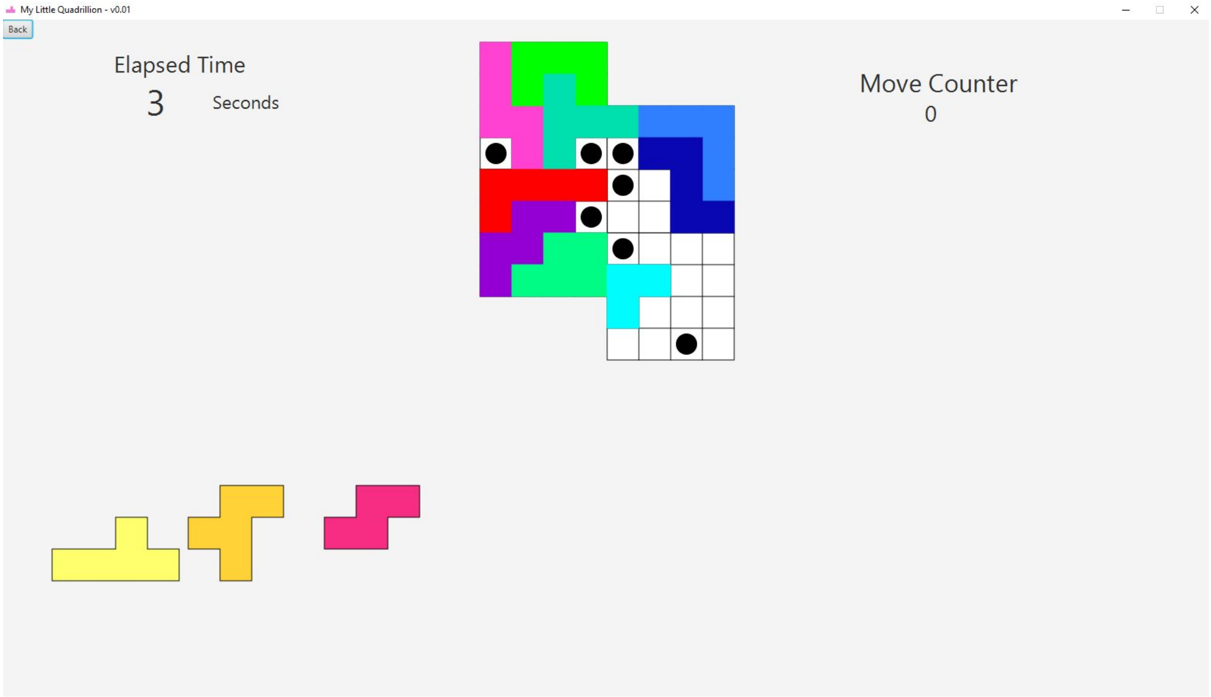


This is main menu. All buttons are self explanatory here.

Arcade Mode

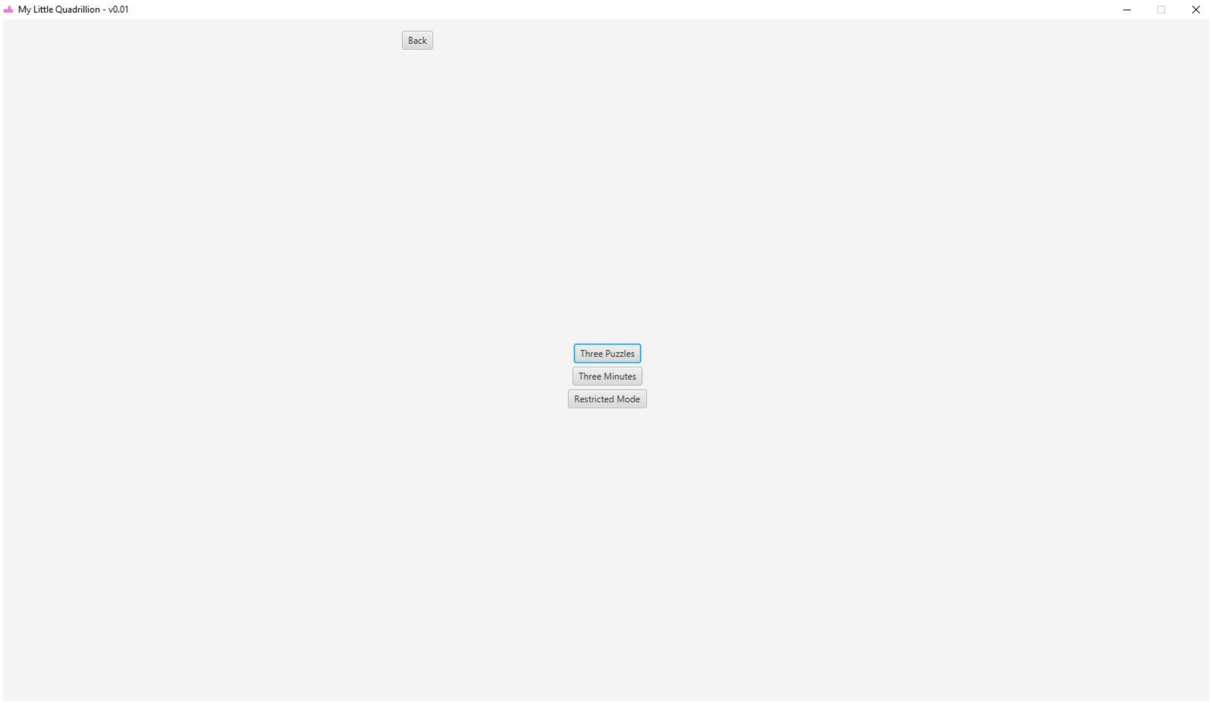


In arcade menu, there are 20 different levels in 4 different difficulty levels. Pieces can be dragged and dropped and snapped to available board places. game piece can be rotated and flipped around its vertical axis. The aim of the game is to fill all the available squares by using pieces.

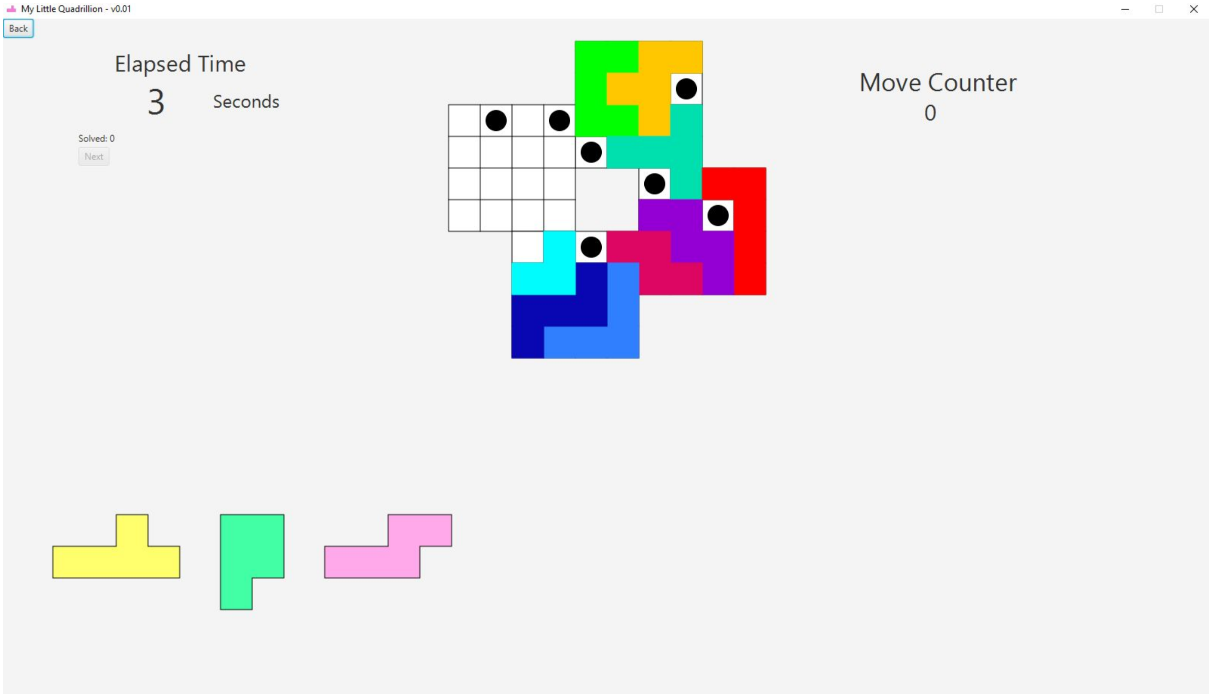


When player enters game, the puzzle is shown to the player along with other pieces to place, time counter and move counter.

Rush Mode



In rush mode menu, there are 3 different rush modes. There is also a counter for how many puzzles the player have solved.



Elapsed Time

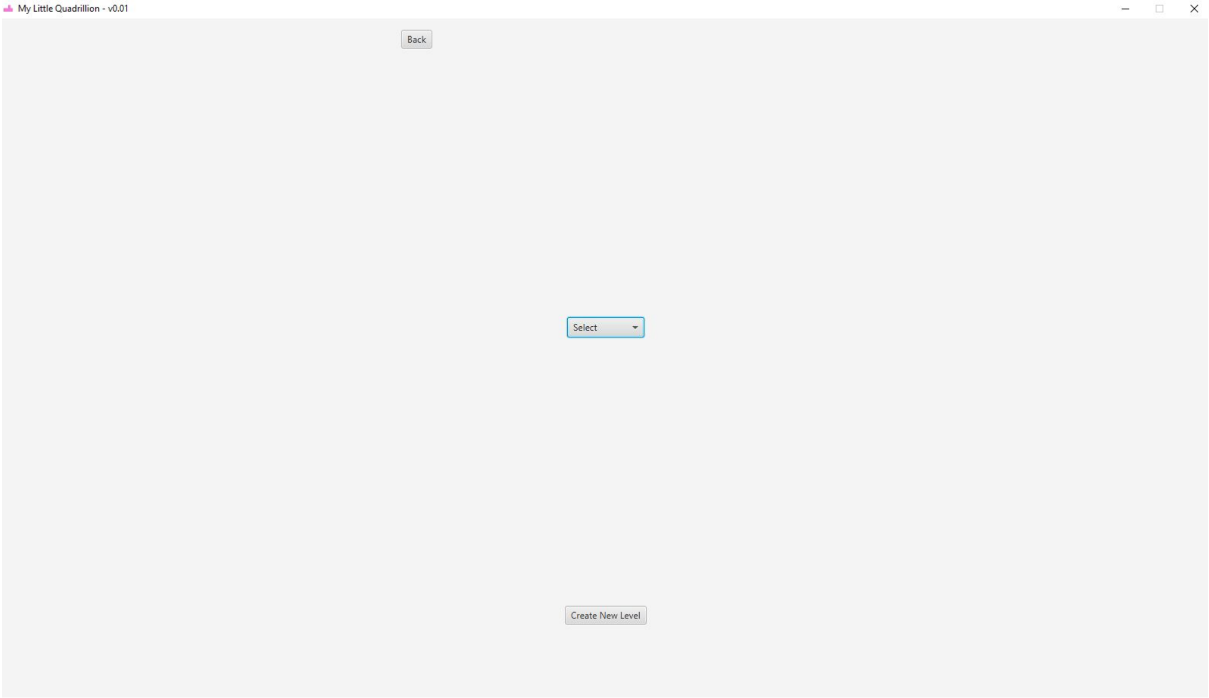
3 Seconds

Solved: 0

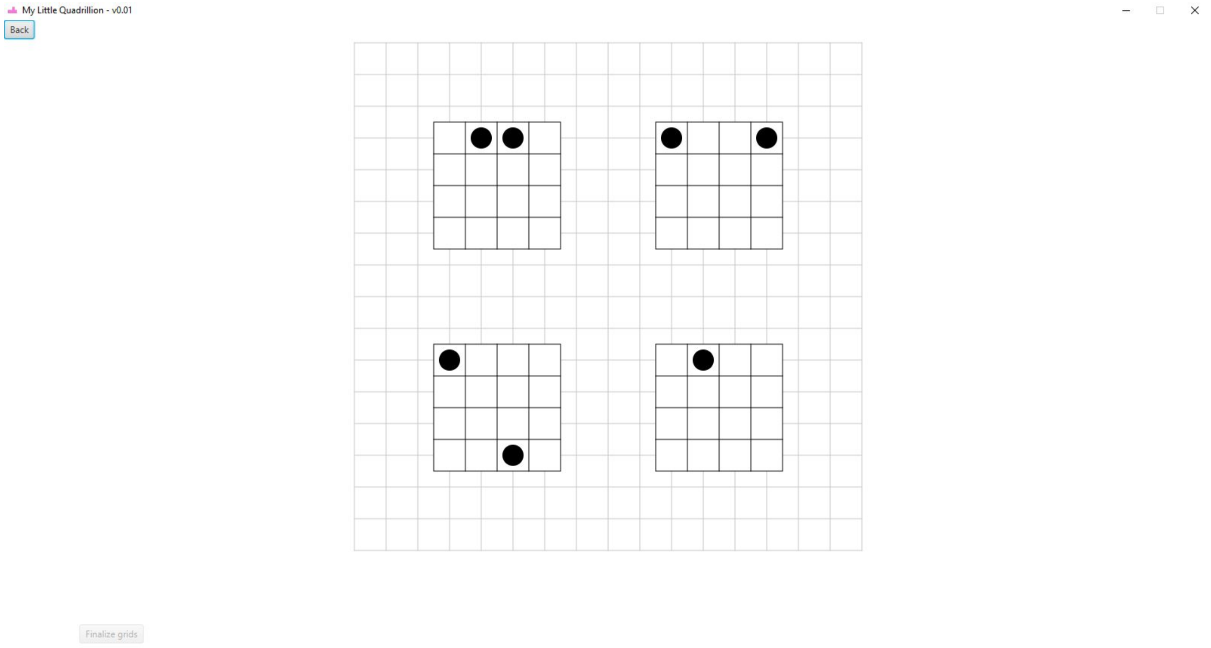
Move Counter

0

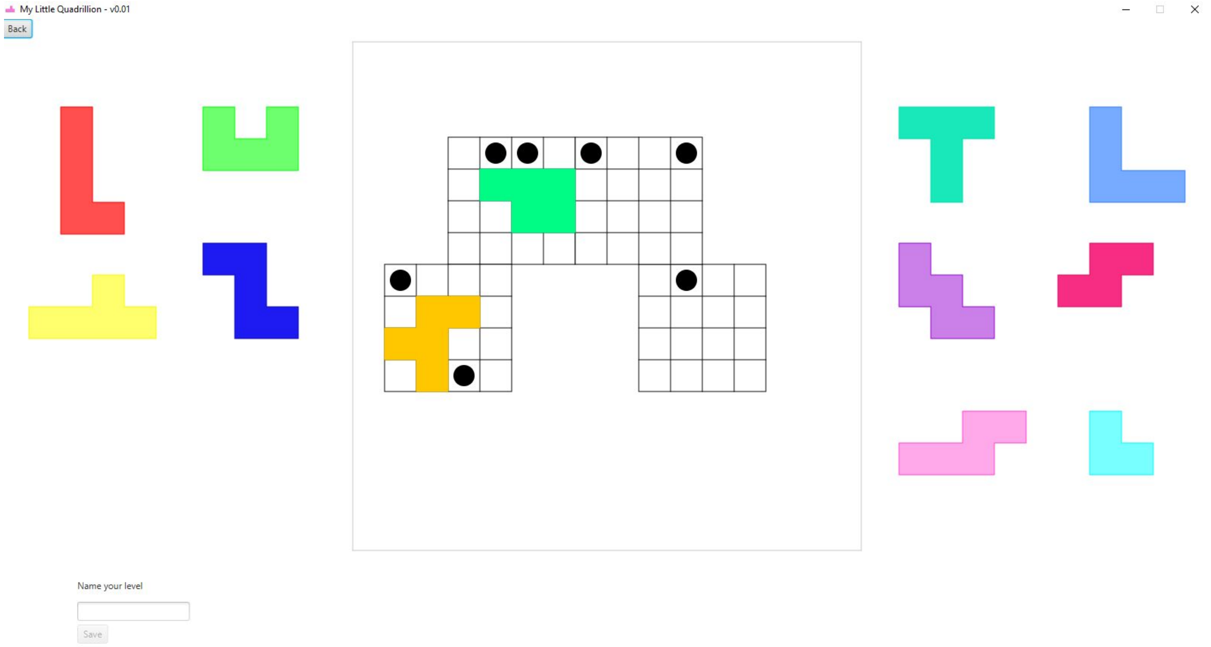
Level Editor



In level editor menu, player can choose between all the previously created levels to play from or player can choose to create a new level.

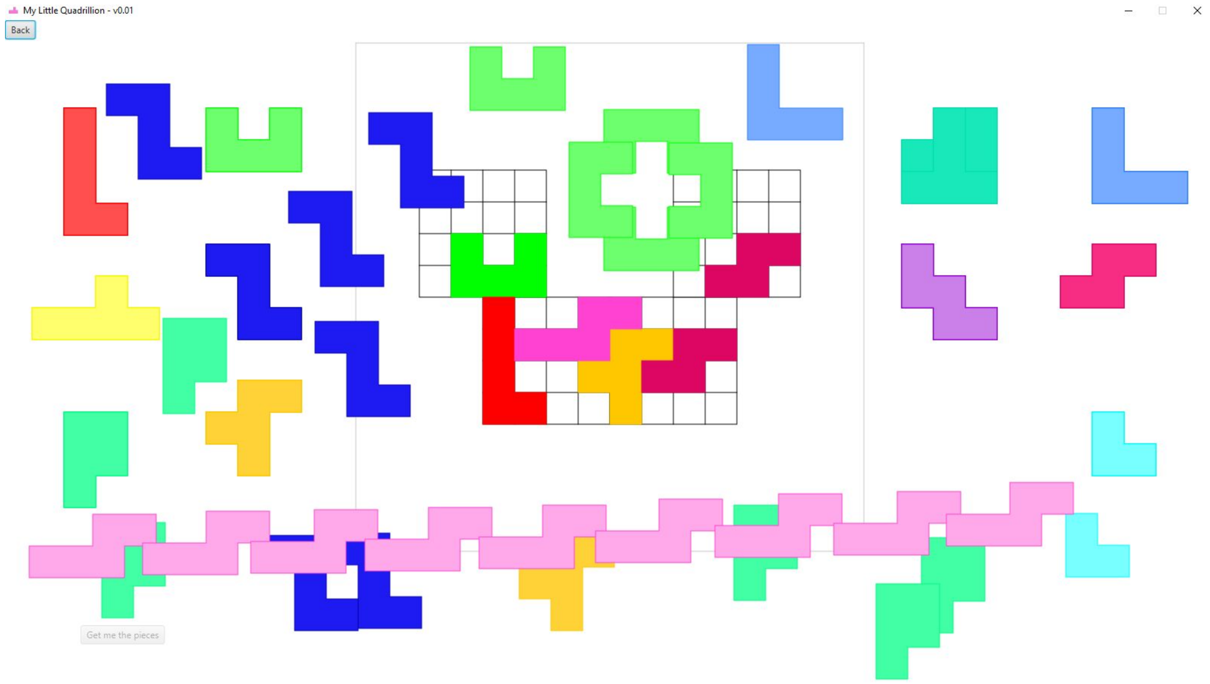


In level editor, player first places grids to the board. Then Player decides the pieces that are going to be on the grid. After placing the pieces, player gives a name to the level and save the puzzle for all players.



After placing the pieces, player gives a name to the level and save the puzzle for all players if the player solves the puzzle succesfully

Sand Box



Is sandbox, there are no rules honestly.