Bilkent University

Department of Computer Engineering

CS319
Object Oriented Software Engineering
Project Final Report

Katamino v2.0

Group 2-F

Sena Er - 21502112

Zeynep Sonkaya - 21501981

Hüseyin Orkun Elmas - 21501364

Utku Görkem Ertürk - 21502497

Bartu Atabek - 21602229

1. Introduction **3**

2. Design Changes **4**

3. Lessons Learnt **4**

4. User's Guide **5**

4.1 System Requirements & Installation 5

4.2 How to Use 5

4.2.1 Game Overview 5

4.2.2 Menu Operations 5

4.2.2.1 Starting a New Classic Single Player Game 6

4.2.2.2 Displaying Highscore Board 6

4.2.2.3 Viewing Credits 7

4.2.2.4 Settings 8

4.2.3 Single Player Arcade Gameplay Operations 8

4.2.3.1 Game Mode Selection Menu 8

4.2.3.2 Player Selection Menu 9

4.2.3.3 Single Player Classic Game Tutorial 11

4.2.3.4 Single Player Classic Game 12

4.2.3.5 Pause Menu 12

4.2.3.6 Level Selection Menu 13

4.2.3.7 Hint Getting 14

4.2.4 Multiplayer GamePlay Operations 14

4.2.4.1 Multiplayer Board Selection 14

4.2.4.2 Multiplayer Game 15

4.2.4.3 Multiplayer Pause Menu 16

4.2.5 Single Player Custom Gameplay Operations 17

4.2.5.1 Custom Game Menu 17

4.2.5.2 Custom Board Maker 18

4.2.5.3 Custom Gameplay 18

5. Work Allocation **19**

Sena Er - 21502112 19

Hüseyin Orkun Elmas - 21501364 20

Utku Görkem Ertürk - 21502497 21

Bartu Atabek – 21602229 22

Zeynep Sonkaya – 21501981 23

1. Introduction

In this report, we aim to discuss the implementation process, the design decisions, the changes during implementation and provide a user's guide for Katamino v2.0. In the remaining of this section, we will discuss the implementation process and current status of the implementation.

After reconsidering our new game design and newly added features, we made some updates in the game that we had before. In the design report, we decomposed our system into subsystems.The subsystems were useful for distributing the work.

We made our meetings face to face when possible, in other cases we used Skype. In these meetings, we took design choices about implementation and structure of our game. We had initial meetings to create tasks and after that most meetings were about specific tasks, therefore the meetings were between assignees of the tasks. This resulted in more efficient meetings.

After all of the game implementation of Katamino v2.0, we have implemented the single player mode of the game with save feature and reload feature,multiplayer game with different boards that has special looks other than real board game Katamino and custom board game with different looks of game boards and an option to design board to be played. In all our game modes we have a time limit. Also single player mode has an hint feature and score feature where all highscores are saved and can be displayed on scoreboard.

Single Player Game implementation includes, movement of pentominoes such as drag and drop, rotations and flips. The algorithms for detecting clashing pentominoes and detection of a full board is implemented. Also, a stopwatch to measure game time, and to calculate score is implemented. Also player's score and board is saved to a file for later play after closing the game and restarting it with the saved player name. Also any game level can be choose if player has already played that level. We have implementations for it too. And for single player version we support giving hints and solutions during game.

In GUI subsystem; main menu, level selection,settings, player selection,board selection, high scores,single player game mode,multiplayer game mode,creation of game board and credits screens are implemented. Furthermore, settings menu and mechanism to enable sound adjustment is implemented.

The multiplayer game mode is implemented in this version where it supports two player play in the same screen by changing turns in every 15 second. In this mode players should be the last player who put an pentomino on the board. For that there exists an algorithm for checking any pentomino that not is on the game board could be fit on the board or not. If a player do its move it changes turn.

Custom board modes are implemented also in this version where there are many different boards than original board game Katamino. In addition an feature is added to add a custom board according to the desire of the player. This mode is very similar to single player game only uses different boards..

In terms of user interface,special components of the game (such as buttons) could be integrated into fxml files. However, for the sake of the simplicity and understandability we separated our components from the main project and built a component library. In the contemporary design of the project imports the external component library which is internal work product.

2. Design Changes

After our first iteration demo, we have decided we should connect same models to game controllers for having 3-layer closed architecture as our design. For that we have created connections between classes.

First we have only single game model and controller classes.When we start implementing multiplayer game we have realized we should separated game into two as Single Player Game and Multiplayer Game. According to what we have learned we have decided to change our design in second iteration design phase and we have changed our implementations for it. Our change decision was right because otherwise we would had hard time to manage single player game and multiplayer game mode and about task assignment.

Secondly we have thought we need a Game Manager to control game controllers where game controllers are all belong to different levels of single player. However during implementation we see that Game Manager is not necessary. Changes could be done in the Single Player Game Controller by changing its model, Single Player Game.

Thirdly we have thought we need enum class for Cell Type but we have solved this problem only using cell id numbers during implementation.

So to sum up, for the second iteration,we have completed all of ours functional features and additionally we have added support for creation of customized board by player’s choose without making big design change.

3. Lessons Learnt

During our implementation we have seen that design and analyses stages are really important and useful for implementation stage. When we start implementing Katamino after first iteration, we have used many information that has been already discussed in the previous stages and we had an outline to how to go with next stages. We have thought many thing,however there were some thing we have not disgusted,implementers took initiative in those cases. Thus those did not cause problem in the implementation part. We know that change can occur anytime of the programing and why we implement it we have

witnessed it from first-hand experience. We have reconsidered our design and learned that not everything could be determined in the earlier stages however many things could be determined in the earlier stages and used in later stages.

Another lesson we learnt throughout the implementation phase is group work is essential and also division of tasks could be hard while tasks are sometimes vague. We had an system architecture that separates our tasks into layers. However in every layer some classes are really connected to each other that they had hard time to coordinate assignment division and used different approaches for task assignment. We have seen being a little bit flexible is useful in a group work while keeping the balance of equal division of labor. However for this stage of implementation our division of labor was easier while we know what is exactly left and what should be reconsidered. We have learned that after a milestone, things go faster like we mentioned in cs319 class.

4. User's Guide

4.1 System Requirements & Installation

Katamino can be runned in all platforms and does not need any installation. It is a single executable JAR. However, in order to be able to run the JAR, computer should have Java Runtime Environment installed.

Code Run Instructions:
Clone the project using IntelliJ IDEA: https://github.com/utgoer/Katamino.git
Import as maven project and follow next buttons.
In the source file run the Main.java

4.2 How to Use

4.2.1 Game Overview

Katamino 2.0 is a tile based 2D puzzle game which is styled with a retro modern user interface, created for the desktop platform. After the initialization of the application the user is presented with a welcome screen which prompts them with options such as choosing a gameplay mode i.e. single-player, or multiplayer, displaying game score board and changing the settings of the game. Players will be able to save their progress and compare their scores with other players.

4.2.2 Menu Operations

The main menu screen of the Katamino v2.0 is given in the next figure. From the main menu, the player can navigate to single player gameplay, multiplayer gameplay, display the scoreboard or change the settings. The details of these operations are given below.

4.2.2.1 Starting a New Classic Single Player Game

Player can start a new game by simply clicking on the single player game button in main menu screen. Afterwards the player chooses between the different game modes and after selecting “Classic (Arcade) Mode, users will be asked to choose their player from the menu and if not present they can create a new player by entering a username.

After name choose in arcade mode,the player will start directly with its last game levels.

4.2.2.2 Displaying Highscore Board

Users can view the current high scores of the players sorted by their scores by clicking on the High Score button in the main menu. After clicking the button, high scores will be displayed in the screen.

4.2.2.3 Viewing Credits

Player can view credits by clicking on the label text at the bottom of the window. After clicking the button, creators of the game will be displayed in the screen.

4.2.2.4 Settings

In settings player can adjust volume of the game music.

4.2.3 Single Player Arcade Gameplay Operations

4.2.3.1 Game Mode Selection Menu

The players presented with two gameplay options; Classic (Arcade) mode and Custom Shapes mode. When clicking either one of them, the player will be presented with the game levels according to their selection.

4.2.3.2 Player Selection Menu

After selecting the single player option the user will be presented with the saved players of the game.

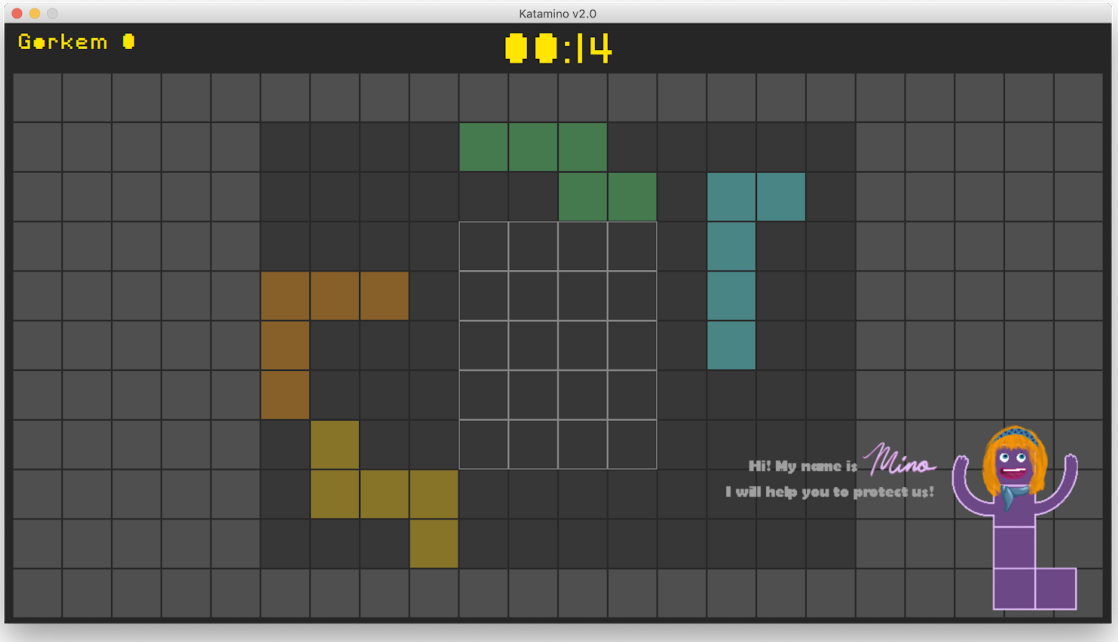
Users can slide through the players and choose theirs by clicking on the player name or click the '+' button to create a new player and entering their name and clicking Enter to finish writing their name.

If they try to write a name already used by other player they will see an error message.

After a player is selected by clicking on the player name they will be lead to their previous game to continue playing their old game.

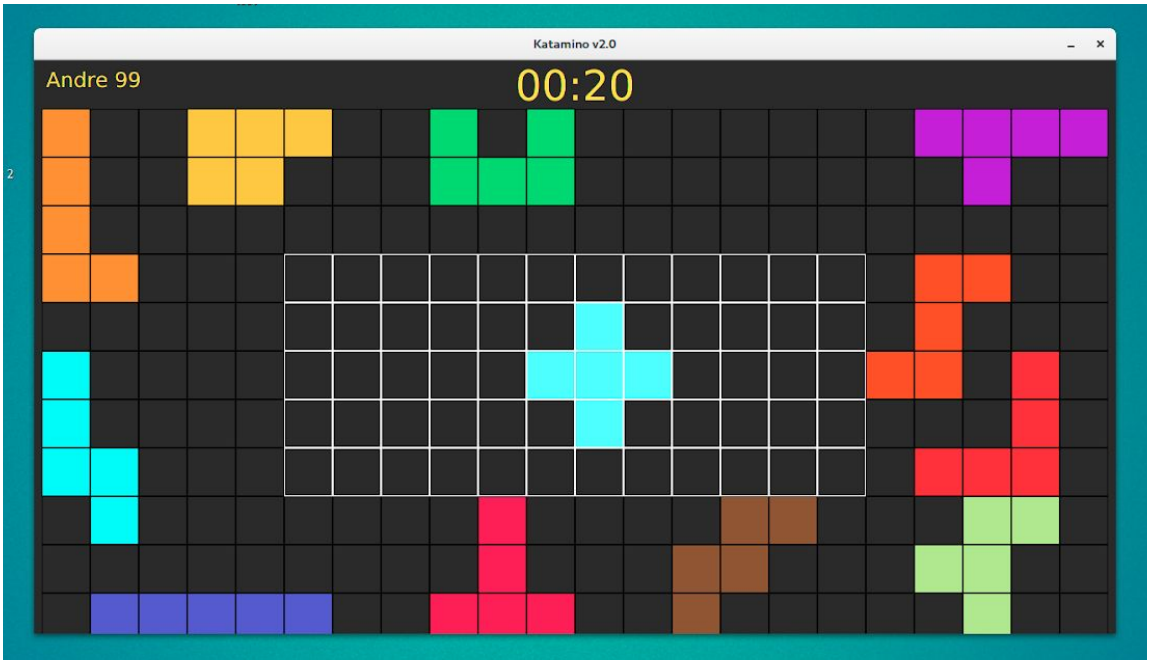
4.2.3.3 Single Player Classic Game Tutorial

If player play for the first time, there is an tutorial to guide him through first level.



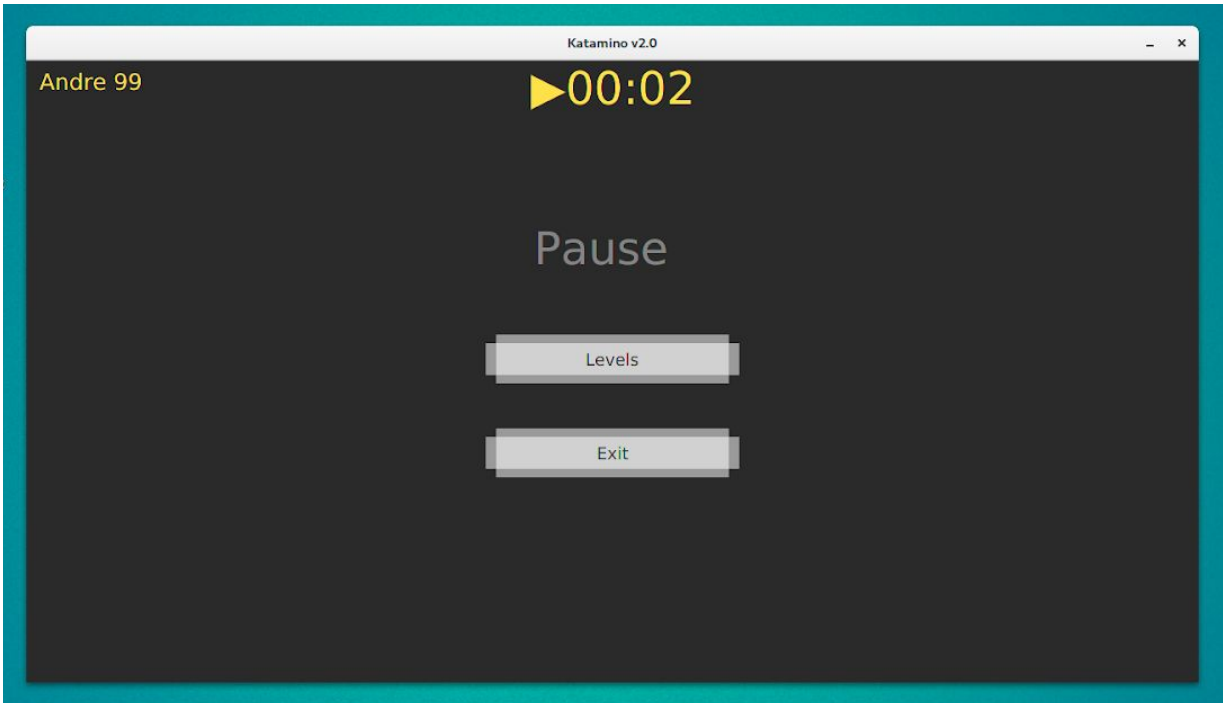
4.2.3.4 Single Player Classic Game

A regular classic game can be played as below.
The player uses drag and drop method and the player should fill the board. Also, there is a stopwatch for time tracking. The main goal is filling the board with pentominoes. If player click on the time, game is paused and player directed to pause menu. If player presses H in keyboard during dragging the pentomino, it shows a hint. Player uses W-S for flipping and A-D for rotating the pentomino during the game.



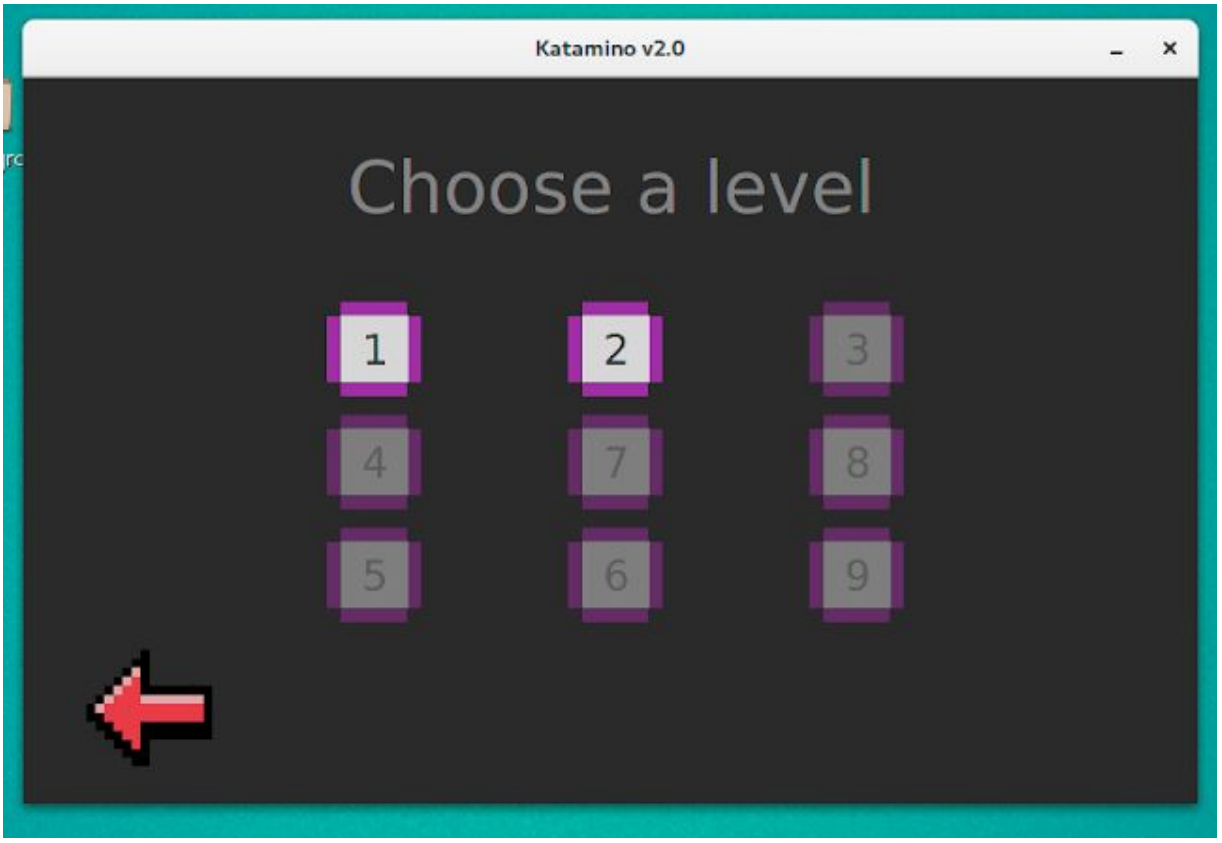
4.2.3.5 Pause Menu

In pause menu player can click the timer and continuo playing its game or can choose levels to see accessible levels for him and there can choose to play an already played game level. Else if player click exit in pause menu it directs the player to main menu.



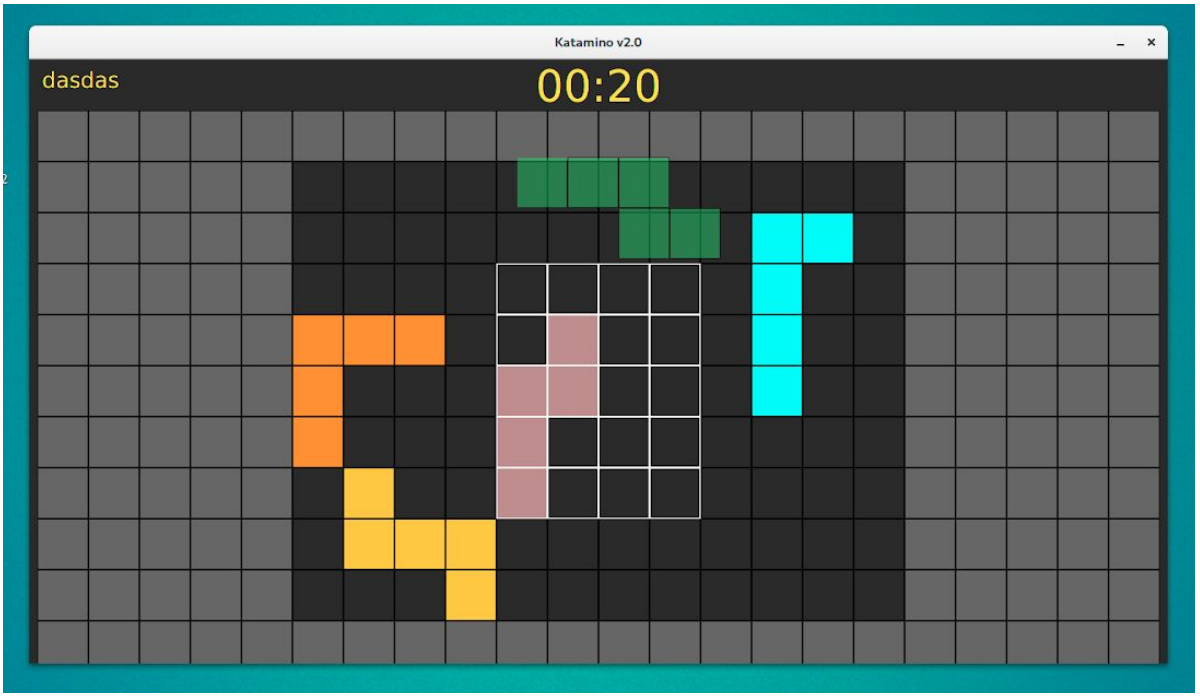
4.2.3.6 Level Selection Menu

The players can choose from the available game levels presented. Some of the levels will be locked and can be seen as greyed out from the interface. As they progress throughout the game more levels will become available and unlocked. When a level selected by clicking their icon the selected level will be loaded to the game board and the game will start.



4.2.3.7 Hint Getting

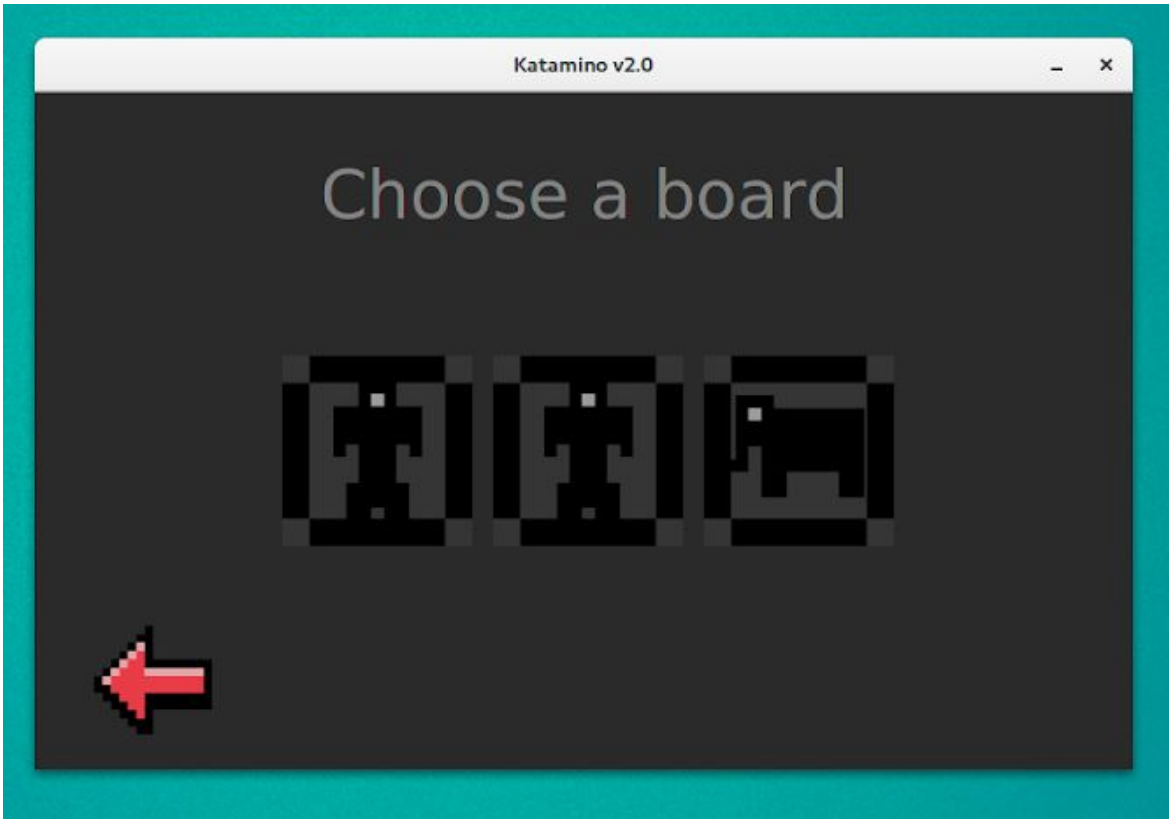
If a hint asked during the game with pressing H while holding the pentomino, hint is shown as below .



4.2.4 Multiplayer GamePlay Operations

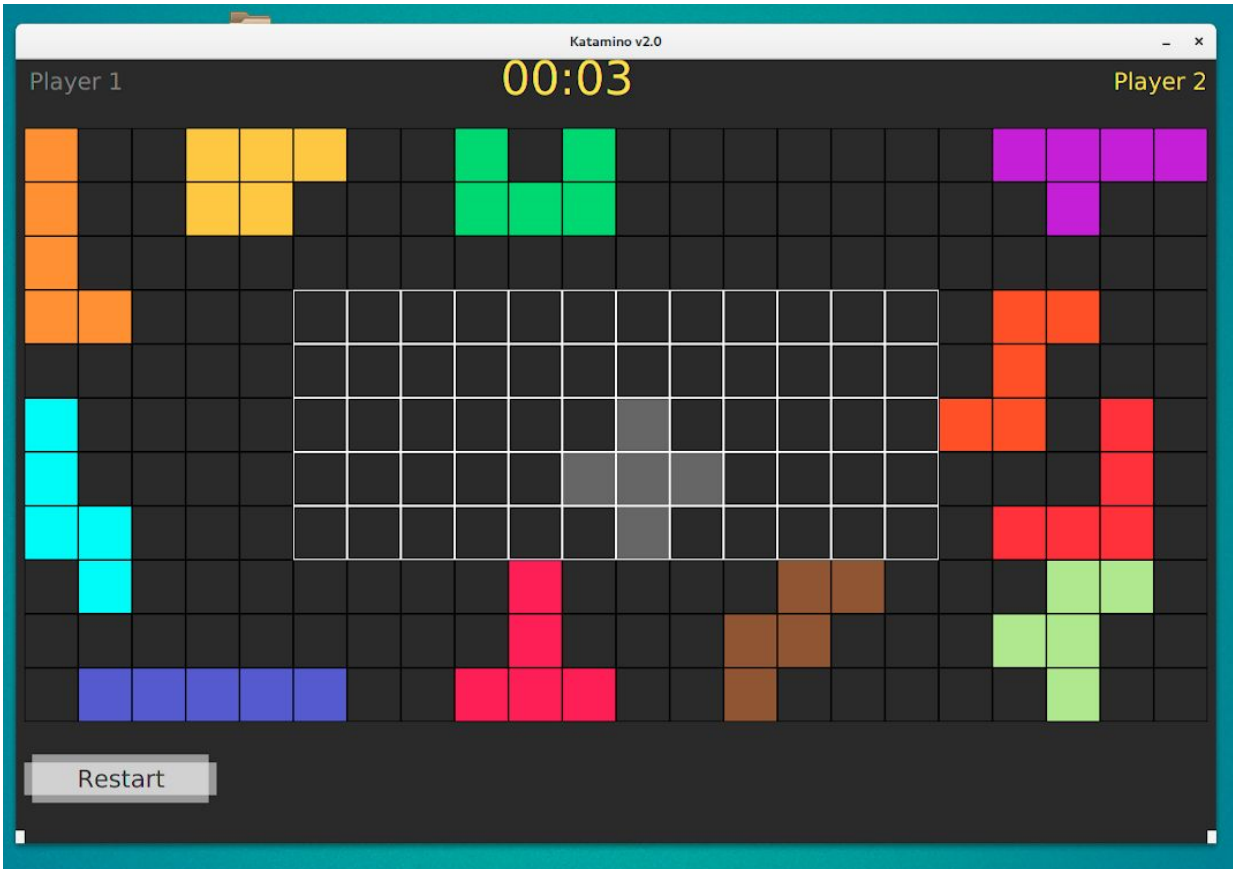
4.2.4.1 Multiplayer Board Selection

Player can choose different shaped board when a board symbol clicked.



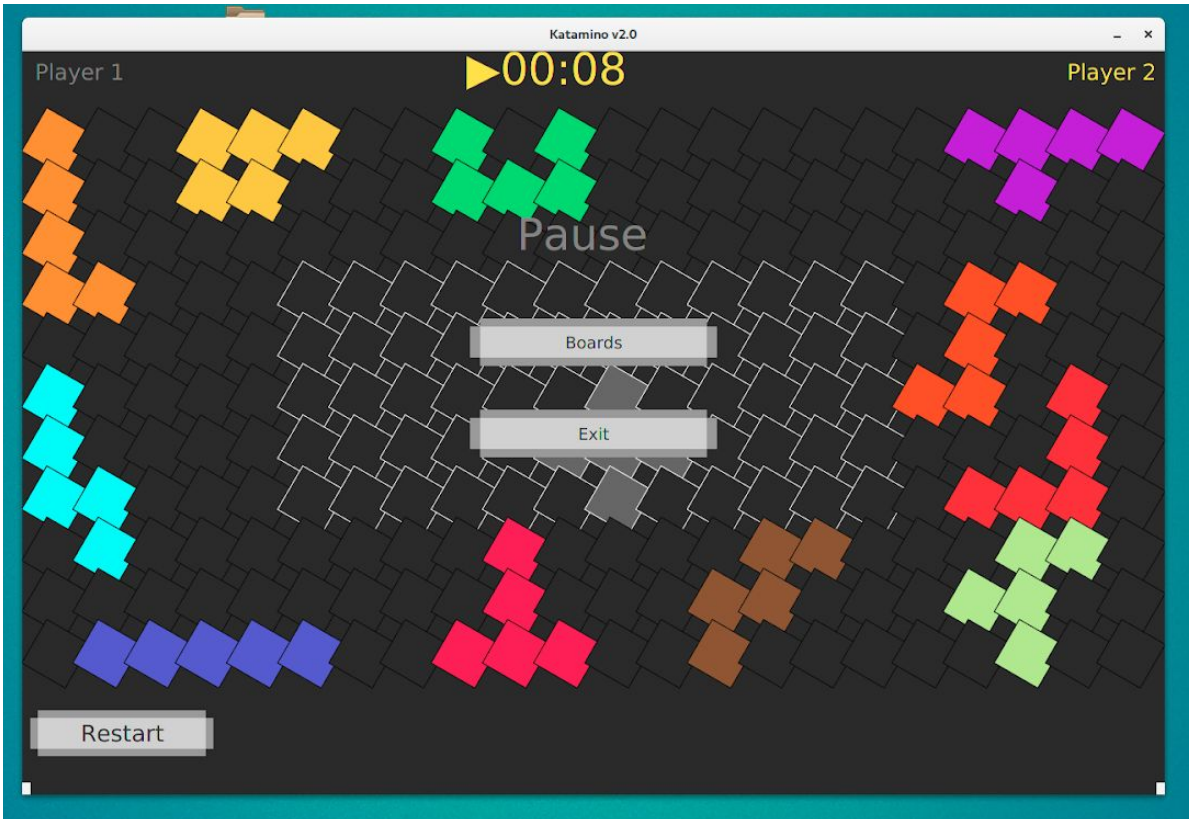
4.2.4.2 Multiplayer Game

In a multiplayer game, there is a turn change in every 15 seconds unless a move made. Yellowed label has the current turn and gray one waits to other player finish up. Also during the game if players want to restart with an empty board they can click Restart button below. Also clicking timer make the game paused and a pause menu displayed where player can choose restart,choose different board and exit the game.



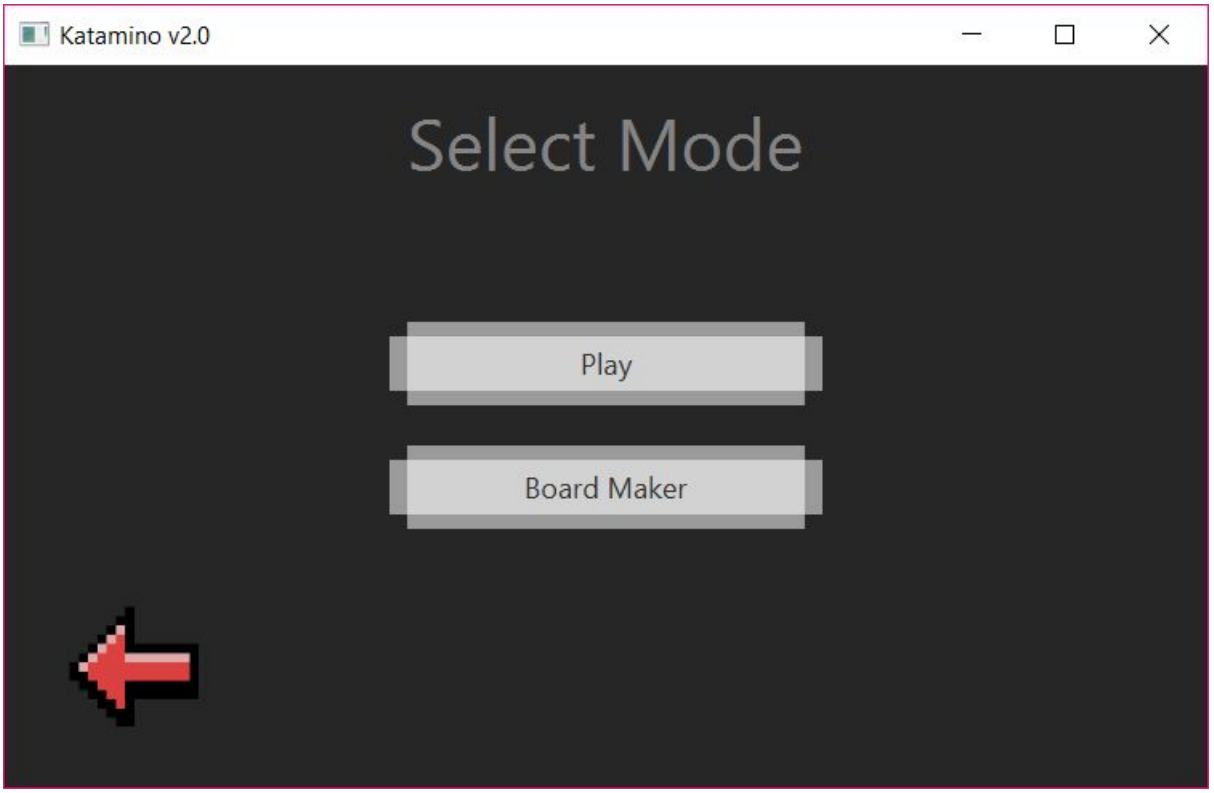
4.2.4.3 Multiplayer Pause Menu

In this pause menu player can click restart to restart the current board or click to Boards and directed to choose a different board from board selection menu or click Exit to turn back to main menu.



4.2.5 Single Player Custom Gameplay Operations

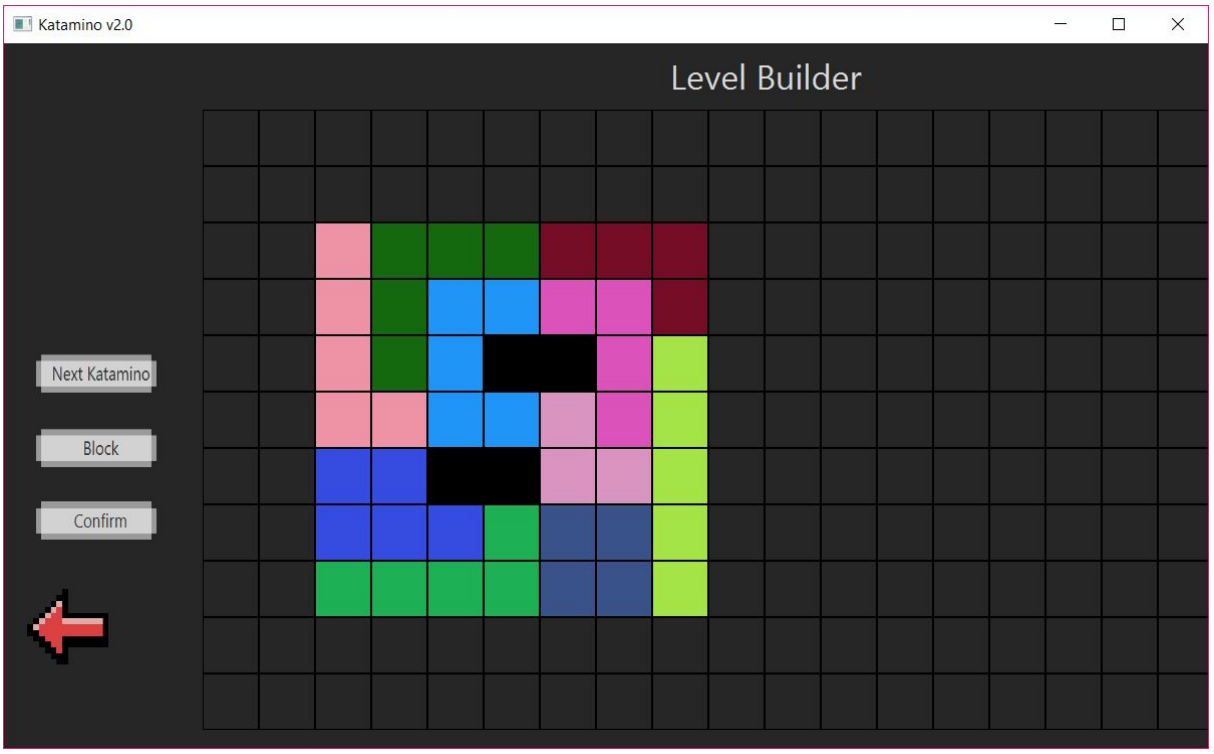
4.2.5.1 Custom Game Menu



In custom game menu, player has two options; first, to play on custom board that is created before, second, to create a new custom board.

Player can select the name of the custom board they want to play from the spinner. Then, they can play single player custom game. The gameplay is the same with the arcade mode, the only difference is that the board is customly created.

4.2.5.2 Custom Board Maker



In Board Maker, player can build their Katamino level. When a tile is clicked, the clicked tile is added to the board and also to the current katamino. In Board Maker, kataminoes are not necessariily pentaminoes so that they can have less or more cells than five in contrast with the classic arcade mode of the game. When player clicks the “Next Katamino” button, a new katamino with a new color is created. In this mode, player can also add blocks. In the gameplay, you cannot put the kataminoes on the blocks. When the player confirms the board, the player should give a name to the board. Then, the board is saved to the Custom Boards collection to play afterwards in the Custom Gameplay mode.

4.2.5.3 Custom Gameplay

The game mechanics in the custom gameplay is the same with classic gameplay. However, there are some small quirks that makes the game even more entertaining! In custom gameplay, the player is not limited with pentominoes, they can play with different sizes of kataminoes. They can play the games which they have created. They can play with different sized boards with different shapes.

5. Work Allocation

Sena Er - 21502112

- First iteration analysis
    - Sequence diagram scenarios with Zeynep and Hüseyin
    - Object and class diagram with Hüseyin and Zeynep
    - Editing/Formatting
    - Use case diagram and its explanations with Görkem and Hüseyin
- Second iteration analysis
    - Revise and update object and class diagram with Hüseyin and Görkem
    - Improvement summary with team
- First iteration design
    - Low level design and explanation
- Second iteration design
    - Improvement summary with team
- First iteration final
    - Editing/Formatting
- Second iteration final
    - Custom gameplay operations
- Implementation
    - Single Player Game Level Menu
    - Initial creation of Game and Gameboard classes
    - Stopwatch
    - Game Controller clash check feature
    - Flip and Rotate Logic of the game with
    - Rotate bug fix with Hüseyin
    - Board Maker (Make your own custom board feature)
        - Saving and loading custom boards
        - Custom game menu
- Tutorial character design

Hüseyin Orkun Elmas - 21501364

- First iteration analysis
    - sequence diagrams
    - object and class diagram with Sena and Zeynep
    - use case diagram and its explanations with Sena and Görkem
    - state diagrams and explanations

- Second iteration analysis
    - revise and update object and class diagram with Sena and Görkem
    - improvement summary with team
    - activity diagram and explanations update
- First iteration design
    - Introduction with Zeynep
    - High-level Software Architecture(includes subsystem decomposition diagram) with Zeynep
- Second iteration design
    - update Introduction with Zeynep
    - update High-level Software Architecture with Zeynep
    - update Low level design
    - improvement summary with team
- First iteration final
    - Introduction
- Implementation
    - Player Selection Menu
    - Flip and Rotate Logic of the game with Sena
    - Rotate Bug fix with Sena
    - Hint feature
- First Presentation Outline

Utku Görkem Ertürk - 21502497

- First iteration analysis
    - use case diagram and its explanations with Sena and Hüseyin
    - activity diagrams' cases
    - object and class diagram's explanations
    - Mockups
- Second iteration analysis
    - added new state diagrams
    - Mockups revise and update
    - improvement summary with team
    - revise and update object and class diagram with Sena and Hüseyin
- First iteration design
    - Subsystem Services with Bartu
- Second iteration design
    - update Subsystem Services with Bartu
    - improvement summary with team
- First iteration final
    - Design changes with Zeynep
    - how to use with Bartu
- Implementation
    - Special components for Katamino Game
    - Settings and main menu controller(menu controllers)
    - Drop feature and update drop and drag feature
    - Clascheck and full state checks with sena
    - Tutorial for the first time play
    - Pause Menu popup for Single Player Game
- Demo video with Bartu
- Presentation with Bartu

Bartu Atabek – 21602229

- First iteration analysis
    - Introduction,
    - Overview,
    - Functional & non-functional requirements,
    - Mockups with Görkem
- Second iteration analysis
    - Revision and update of the use case diagram and its explanations with ,Zeynep.
    - Improvement summary with team,
- First iteration design
    - Subsystem Services with Görkem,
- Second iteration design
    - Update Subsystem Services with Görkem,
    - Single Player game & cell state diagrams,
    - Improvement summary with team,
- First iteration Final Report
    - how to use with Görkem,
    - Presentation design & layout,
    - Demo video with Görkem
- Implementation
    - Custom UI components (i.e. Katamino Drag Cell, Katamino Drag Block, ScoreBoard label…),
    - Drag and Drop features in Game Controller,
    - Continuous gameplay and level transitions (level animation),
    - Overall UI layout and scene transitions,
    - Connection of player data through the game,
    - Stopwatch fix
- Second Iteration Final Report
    - Presentation with Görkem,
    - Demo video with Görkem

Zeynep Sonkaya – 21501981

- First iteration analysis
    - object and class diagram
    - sequence diagram scenarios with Sena and Hüseyin
    - activity diagram
- Second iteration analysis
    - revise and update use case diagram and its explanations with Bartu
    - improvement summary with team
    - revise and update scenarios and sequence diagrams
    - revise and update non-functional requirements
- First iteration design
    - Introduction with Hüseyin
    - High-level Software Architecture with Hüseyin
    - Rewrite class diagrams
- Second iteration design
    - update Introduction with Hüseyin
    - update High-level Software Architecture (includes subsystem decomposition diagram) with Hüseyin
    - Rewrite class diagrams
    - improvement summary with team
- First iteration final
    - Design changes with Görkem
    - Lesson Learnt
- Implementation
    - File Manager class that saves player information and retrieve information from file,adds new players to file
    - Connect models and controllers according to subsystem decomposition
    - Load last game of an old player if a new player start it from first level logic
    - bug fix for old level selection and coming back to current level
    - Multiplayer game mode with algorithm to check any movements left ,not letting put pentominoes where should not be put, and restart and interface of those by using single game interface

    - Seperate game to single and multiplayer and arrange their controller according to inheritance
    - Scoreboard and its interface
    - First version of cells and gameBoard
- Second Iteration Final Report
    - Rewrite Introduction
    - Rewrite Design Changes
    - Rewrite Lesson Learnt
    - Rewrite How to Use
    - Work Allocation