Cs 319-Object-Oriented Software Engineering

Fall 2019

Analysis Report

Super Katamino

Group 2K / Section 02

- Hasan Doğan - 21402109
- Mert Özerdem - 21300835
- Sine Mete - 21302158
- Sencer Umut Balkan - 21401911
- Mustafa Azyoksul - 21501426

Instructor - Eray Tüzün

Table Of Contents

1. Introduction

Super Katamino is a puzzle game which project group 2K is inspired by a board game called “Katamino”[1]. The basic gameplay of “Katamino” the board game and “Super Katamino” is similar to each other with some improvement on to the vanilla Katamino the board game, which is the completing given board with available pieces without missing a spot on the table. Board game and Super Katamino shares same type of pieces such as 3x1,1x2 and 2x2 pieces. Pieces are rotatable in order to get more combinations and longer gameplay time with different designs.

We will add some new features to vanilla Katamino board game in order to make the project more fun in the long run and to fit course requirements:

1. Animations
2. Random level generator
3. Sound and Music
4. Achievement displayer(in-game)

To implement a random level generator we will use our self-made algorithm. We will use JavaFX to implement our project. A mouse or touchpad is a must for use of Super Katamino the game.

2. Overview

2.1 Gameplay

Super Katamino is a 2D puzzle game which is inspired by Katamino the board game. So gameplay of Katamino and super Katamino are similar to each other with some improvements added here and there. The aim of the game is to fill all available space on the template table, which varies from 4x4 to 10x10 square template tables to 12x5 rectangle template table, by using generated pieces that designed for specifically to fit on the template table. The player needs a mouse to drag and drop the pieces on the template table in order play the game. If

the player misses the place on the template table or spot is already filled, the game returns the piece its original place. The game finishes after template table is filled without a spot and continues to the next level if it is not the last level of the game. Then his or her score will be displayed if a player can manage to beat higher scores on the scoreboard.

2.2 Pieces

Pieces of the Super Katamino is the core part of the game and they are generated randomly. By algorithm that divides nxn template tables by n different pieces which can vary in size and shape such as L, Z and U shaped pieces. These pieces interact with the mouse and moves with the desire of the player's will.

2.3 Template Table

Template table is another crucial part of the Super Katamino The game. Template tables are generated according to the difficulty of the level. These tables can vary from 4x4 to 10x10 square and 12x5 rectangle table. These tables create win condition for the game.

2.4 Time Counter

Time counter implemented to give a sense of achievement to the player. Time counter counts the completion time of the level and assigns a point according to level’s difficulty and time spend on the level.

2.5 Score Keeping

Scores add to each other after each level completion until the game is finished or the player exits from the game. Then these scores are displayed on the scoreboard if the point is higher than the previously scored points.

3. Functional Requirements

3.1 Play Game

The main function required of the game is “Play Game” function. This game function runs the game and displays on the screen for user to enjoy it. A player can start this function by just clicking the start function on the main menu. Then this function generates a level and pieces in order to start the game.

The game is designed to be played by drag and drop approach. The player clicks on the piece that he wants to drop on the template table and drags it to the place that he sees fit, then the player can drop the piece to fit the place. If the piece is out of place or dropped on a place that is already occupied it returns to the original place that the player picked up the piece.

The main objective in the game is to fill all the available places on the template table without skipping a place on the table. Players can get more score by completing harder difficulty levels or completing in less time interval. If conditions are satisfied game moves on the next level and keeps the score.

3.2 Help

This function designed to help people to play and learn the game as the name “Help” suggests. This Function explains:

1. Game Rules
2. How to interact with the game
3. how scores are kept
4. how to mute sound and music

Players can reach the information they needed to play the game by just clicking the help button on the maşn menu. Which makes the game more accessible to players of all ages.

3.3 Settings

This is an and main menu function that is designed for shut off or reactivate the music and sound of the game to get more optimal gameplay experience for the player. This function sets up:

1. Music
2. Sound

3.4 ScoreBoard

Players can get some sense of achievement by the implementation of this function. A player can reach this board by clicking the “High Scores” function on the main menu. This function displays players with high scores on the UI when clicked. These scores are calculated by formulating difficulty of the level x completion interval of the level which equals to score of the player’s end round score. After the level score is calculated it is added to the previous level’s score until the last level of the game is reached or game is prematurely stopped.

4. Nonfunctional requirements

4.1 Frame rates

The frame rate should be in a reasonable range. The minimum and average frame rate should be 20 and 30 frames per second respectively.

4.2 Response time

The average response time between click and reaction should be less than half seconds. The maximum response time between click and reaction must be two seconds.

4.3 Required resources

The game is going to be able to run with 1024 MB of RAM. Also the game will use less than one gigabyte of hard disk space. Checking the total size of the folder in which the game was installed and checking the physical memory in the Windows Task Manager Performance tab are how we are going to know these requirements are satisfied.

4.4 Platform

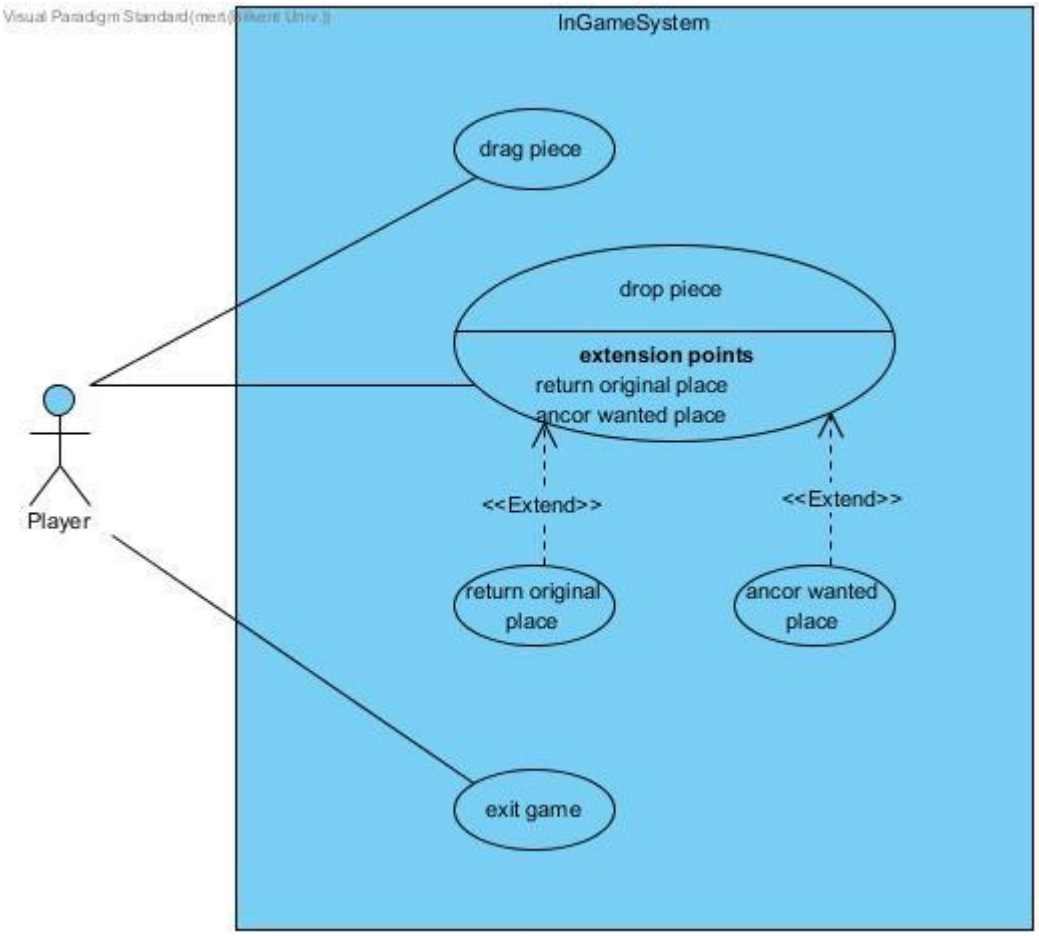
The game will be compatible with Windows 7. To play the game, user will be required to use Window 7 or above.

5. System models

5.1 Use case model

In this part of the UML models, we show interactions between the user and the system by using UML use case models with different models that each emphasizes a different aspect of user system interaction.

5.1.1 InGameSystem



Use Case 1: drag piece
Primary Actor: Player
Stakeholders and Interest:

1. Player drags Player drags piece in order to fill all places template table.
2. System moves piece with the direction of the mouse.

Entry Conditions:

1. Player must click on valid piece object.

Use Case 2: drop piece
Primary Actor: Player
Stakeholders and Interest:

1. Player drops the piece in order to fill available space in template table.

2. System returns or anchors the piece.

Entry Condition: -

Use Case 2: return original place
Primary Actor: Player
Stakeholders and Interest:

1. Player drops the piece in order to fill available space in template table.
2. System returns the piece to original spot.

Entry Condition:

1. piece must be dropped on to some place on game UI.

Use Case 3: ancor wanted place
Primary Actor: Player
Stakeholders and Interest:

1. Player drops the piece in order to fill available space in template table.
2. System ancors the piece to original spot.

Entry Condition:

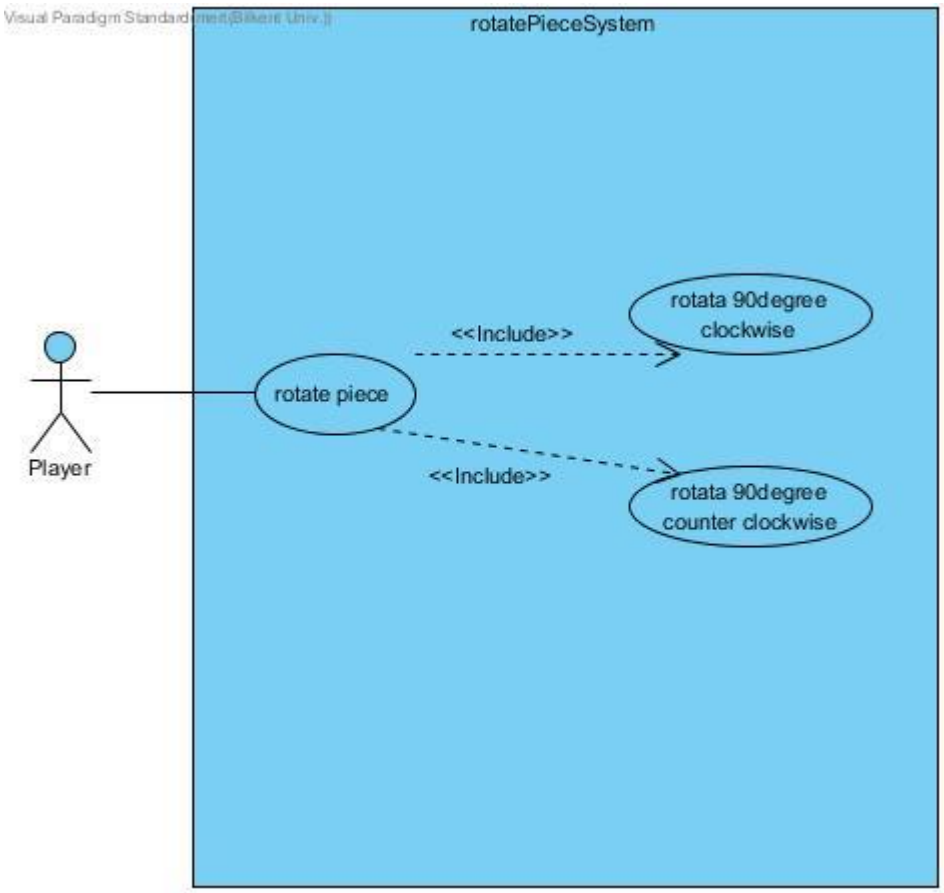
1. piece must be dropped on to free space on template table.

Use Case 4: exit
Primary Actor: Player
Stakeholders and Interest:

1. Player clicks on exit button.
2. System calculates points gathered.

Entry Condition:

1. Player clicks on exit button.



Use Case 1: rotate piece
Primary Actor: Player
Stakeholders and Interest:

1. Piece rotates.
2. System rotates piece according to object specifications.

Entry Condition:

1. Player clicks on rotate button.

Use Case 2: rotate 90degree clockwise
Primary Actor: Player
Stakeholders and Interest:

1. Piece rotates 90 degree clockwise.
2. System rotates piece 90 degree clockwise according to object specifications.

Entry Condition:

1. Player clicks on rotate button.

2. A piece must be selected.

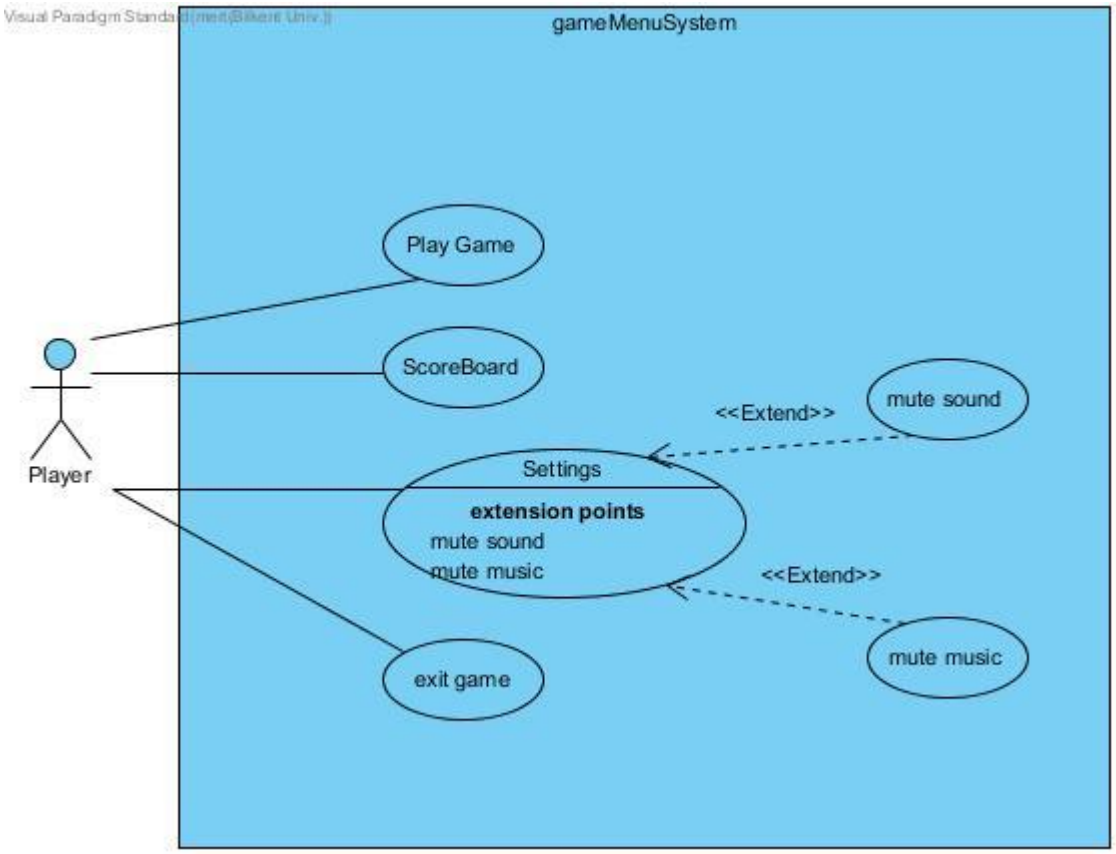
Use Case 3: rotate 90degree counter clockwise
Primary Actor: Player
Stakeholders and Interest:

1. Piece rotates 90 degree counter clockwise.
2. System rotates piece 90 degree counter clockwise according to object specifications.

Entry Condition:

1. Player clicks on rotate button.
2. A piece must be selected.

5.1.3 GameMenuSystem



Use Case 1: Play Game
Primary Actor: Player
StakeHolders and Interest:

1. Player clicks on “Play Game” button to initialize game.

2. System initialize game.

Entry Condition: -

Use Case 2: ScoreBoard
Primary Actor: Player
StakeHolders and Interest:

1. System displays scoreboard.

Entry Condition: -

Use Case 3: Settings
Primary Actor: Player
StakeHolders and Interest:

1. Player clicks on “mute sound” or “mute music”
2. System cuts off music or sound.

Entry Condition: -

Use Case 4: mute sound
Primary Actor: Player
StakeHolders and Interest:

1. Player clicks on “mute sound”
2. System cuts off sound.

Entry Condition: -

1. Settings menu must be opened.

Use Case 5: mute music
Primary Actor: Player
StakeHolders and Interest:

1. Player clicks on “mute music”
2. System cuts off music.

Entry Condition:

1. Settings menu must be opened.

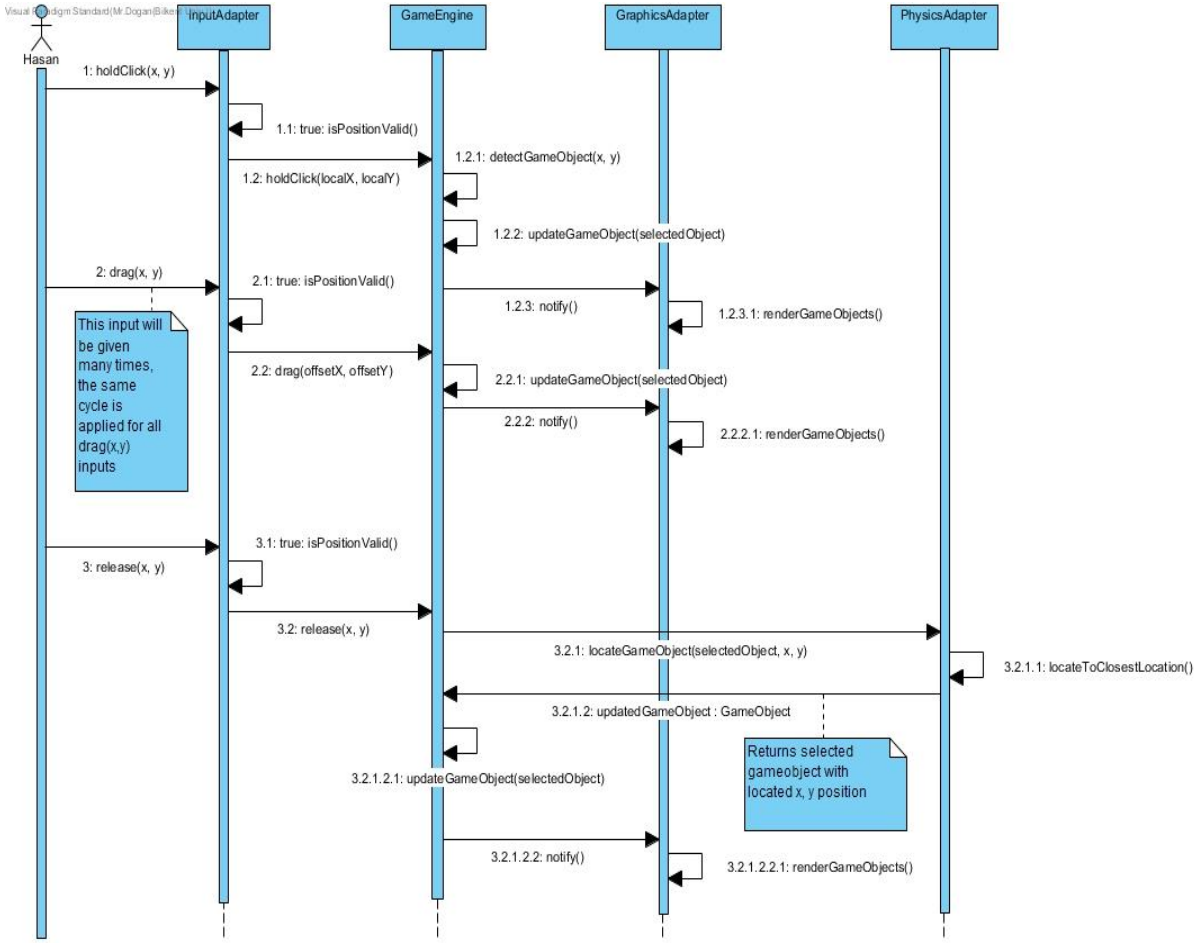
Use Case 2: exit game

Primary Actor: Player
StakeHolders and Interest:

1. Player clicks on “exit game” button
2. System shuts down the game.

Entry Condition: -

5.2. Dynamic models

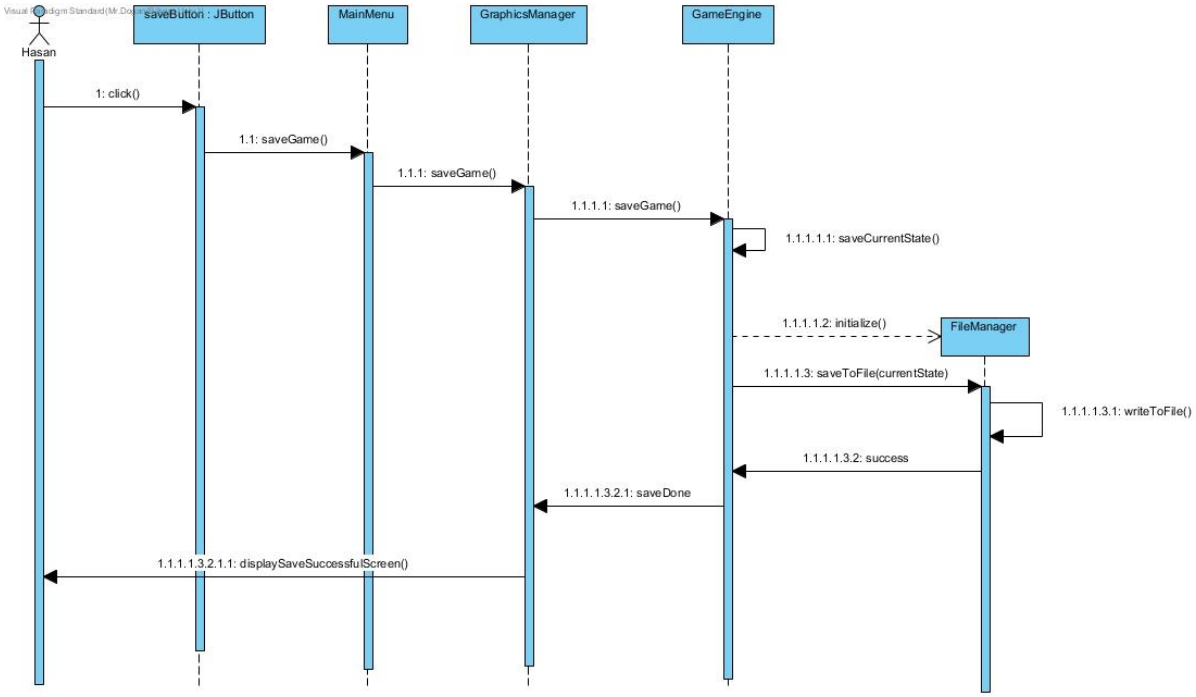


We need to indicate that in this sequence diagram, it is accepted that user behaves as expected.

Flow 1: User gives system an input by clicking somewhere on screen. InputAdapter checks if user has clicked on an object or not. Considering s/he did, InputAdapter transports the information to the GameEngine which detects which object is clicked. Then GraphicsAdapter renders the object in order to let user understand that object is being hold.

Flow 2: After holding an object, user starts to drag it. InputAdapter controls if the object is being dragged in the allowed area and trasports the information to GameEngine which will again notify the GraphicsAdapter to make the object rendered and seen by user.

Flow 3: User stops dragging and wants to place the object somewhere on the board. User releases object, InputAdapter makes its routine control and then gives the input to GameEngine. As user cannot place the object to certain pixels, PhysicsAdapter updates x and y values of object as the closest spots on the board that the object can be placed. Then GameEngine lets GraphicAdapter know the object in its new place and render it and display to the user



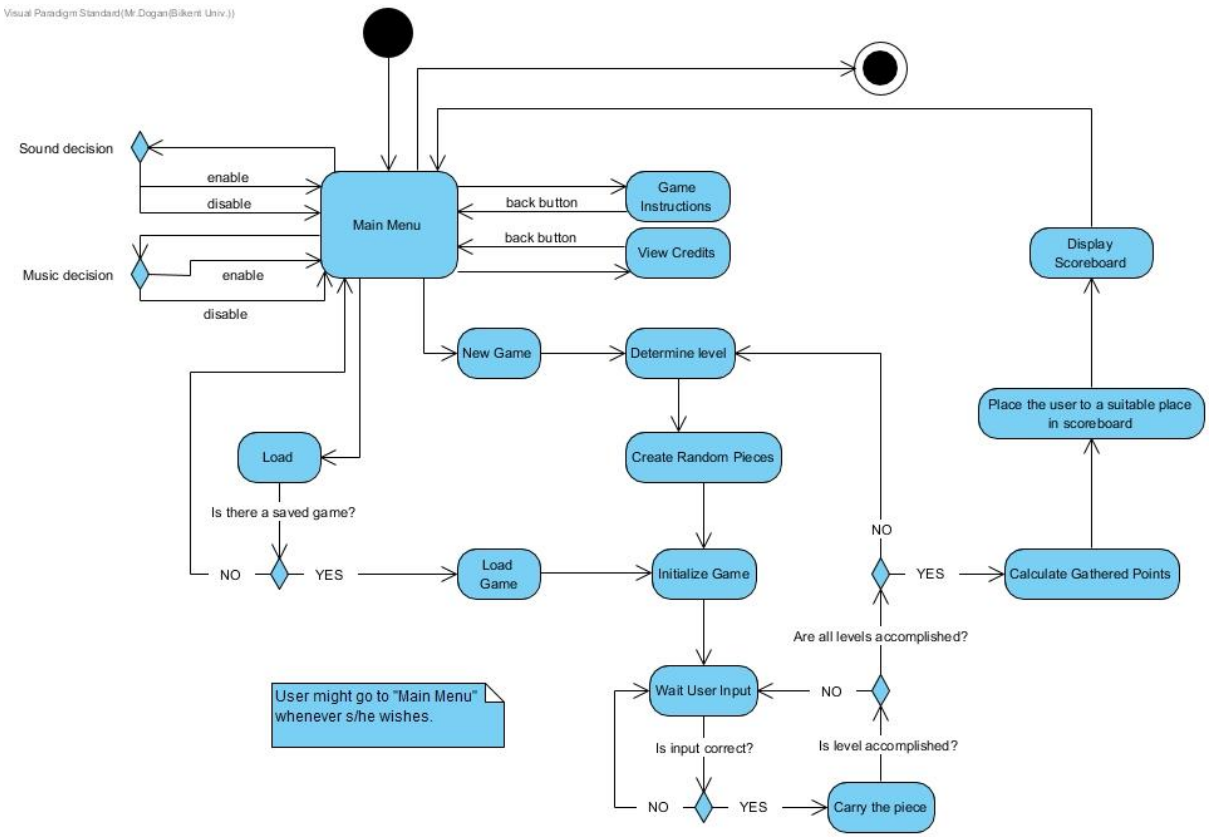
The sequence diagram above shows how to save a game using FileManager.

5.2.1 Activity Diagram

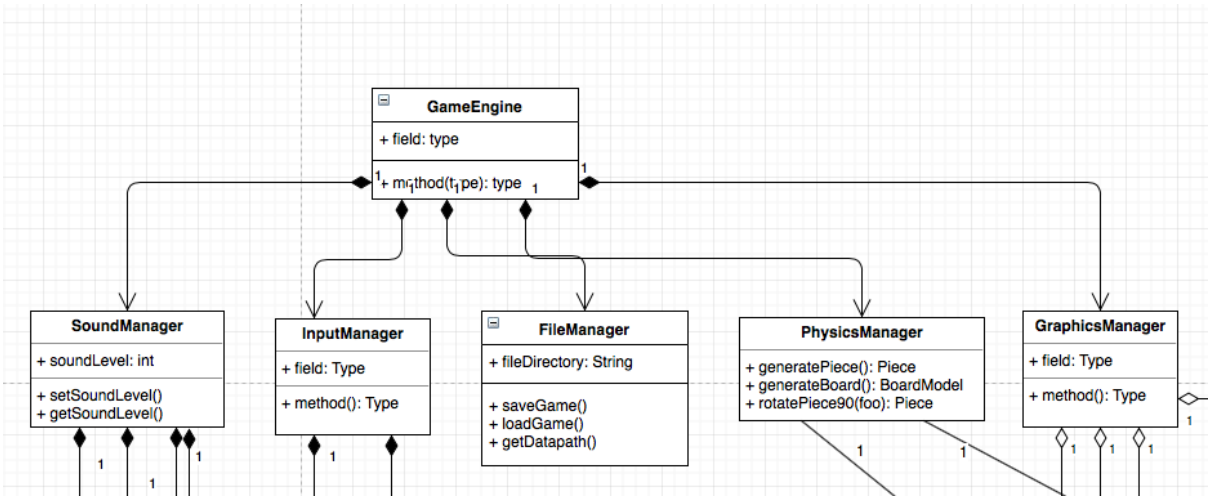
User encounters with main menu when .exe is executed. In main menu there are options to enable and disable in game sound and music. Also player can reach game instructions to learn how to play

SuperKatamino. Credits also can be seen by user. If user wishes to start a new game, level is determined, which is 1st level in this case, and pieces that will fit on the board are created randomly and game initialized. If user wishes to load a game which is saved before, system continues from game initialization step and moves forward. After the initialization of the game, system awaits inputs from user and checks if the inputs are correct. If the input is wrong, system keeps waiting till an accurate input is provided. When a given input is correct, system checks if that was the last piece that fulfills the board. If it is not, level is not accomplished and some new inputs are waited from user. If the level is accomplished and if it is not the last level, system moves on to the next level. If it was already the last level, then system calculates the total point that player gathered and puts in a suitable place on scoreboard. Then player is directed to main menu and is let act according to his/her will. S/he can exit or keep playing starting a new game or a saved one.

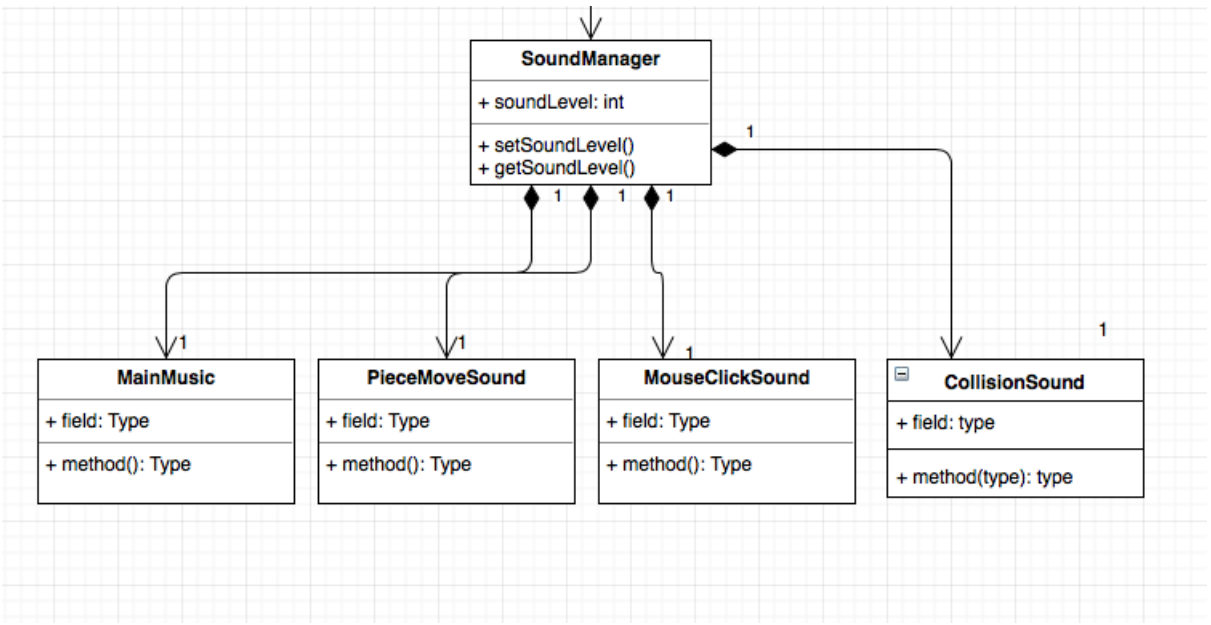
In SuperKatamino, exits are from main menu, and user might go main menu whenever s/he wishes.



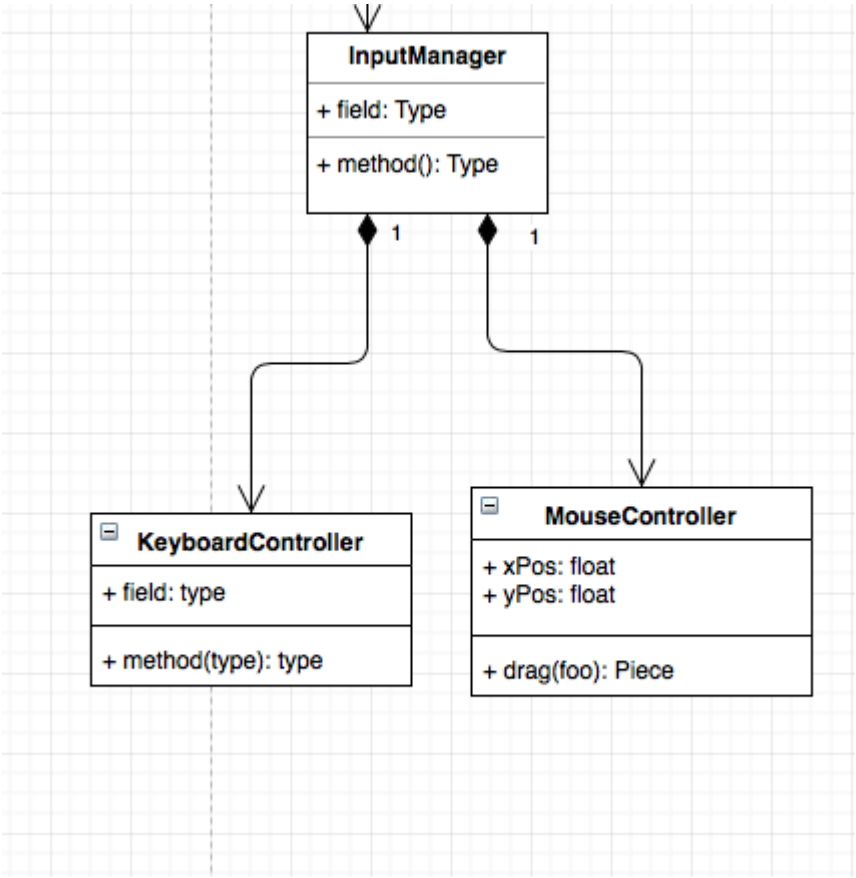
5.3. Object and class model



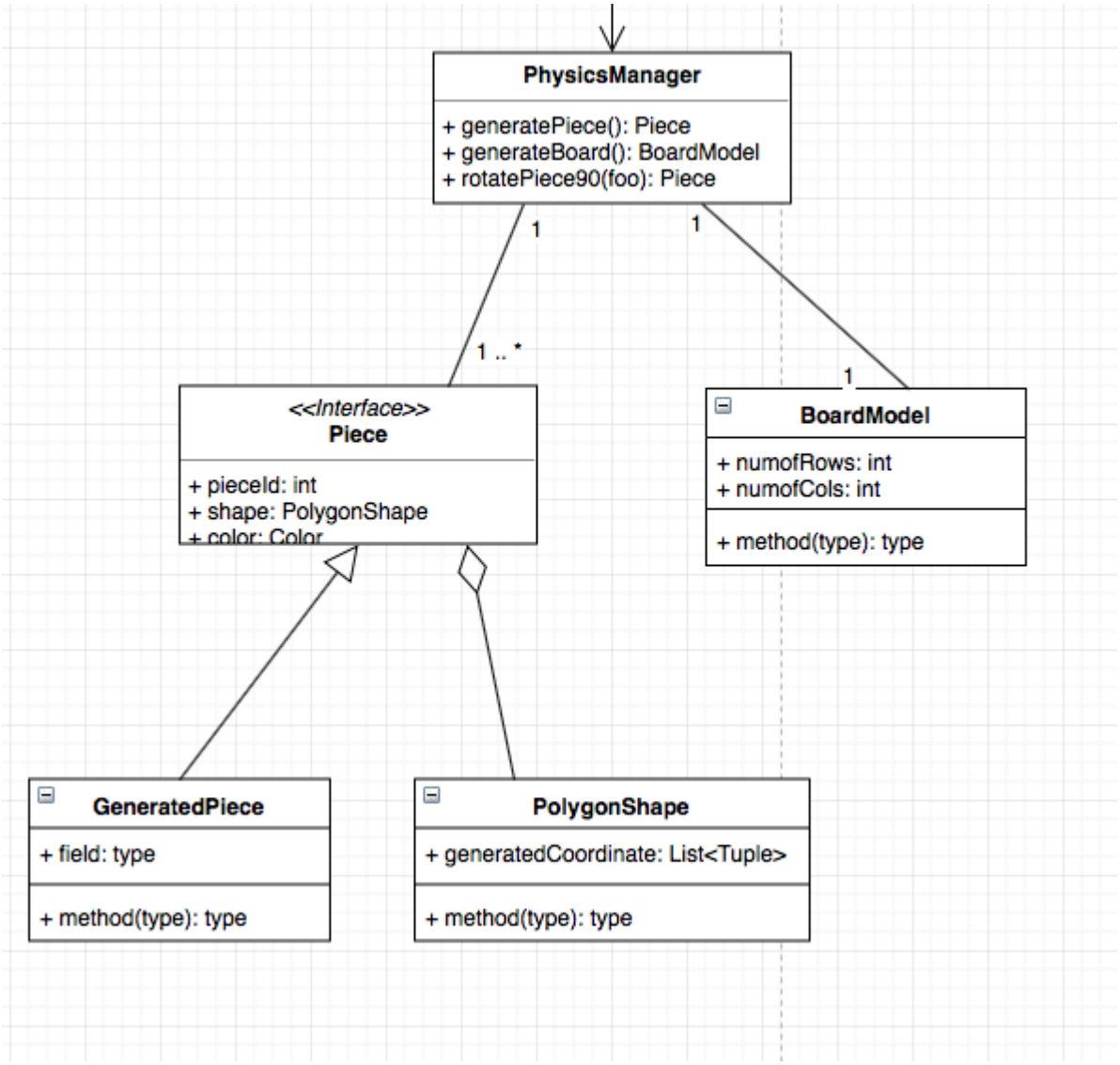
GameEngine controls five other engines within the same package and assures their communication, simply it has the game loop.



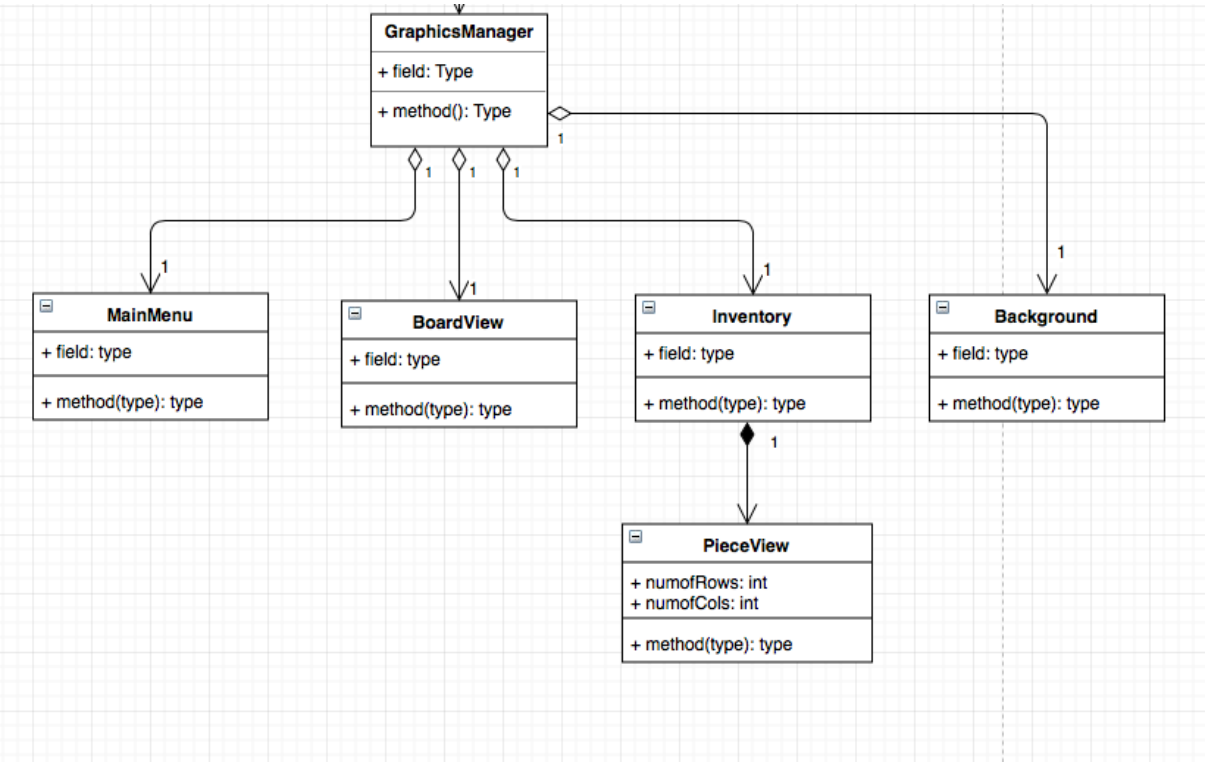
SoundManager is responsible for sound oriented events and sound threads.



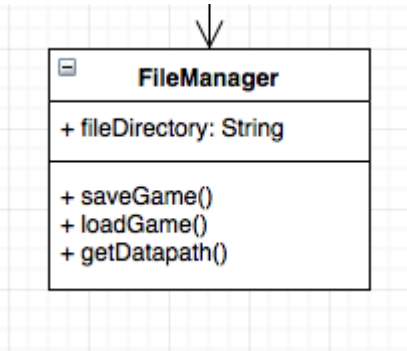
InputManager receives input through keyboard and mouse, controls and manipulates their data with respect to GameEngine and other sub-engines.



PhysicsManager is a controller for the Model package such as Piece and Board. Keeps track of their location, size, color etc.



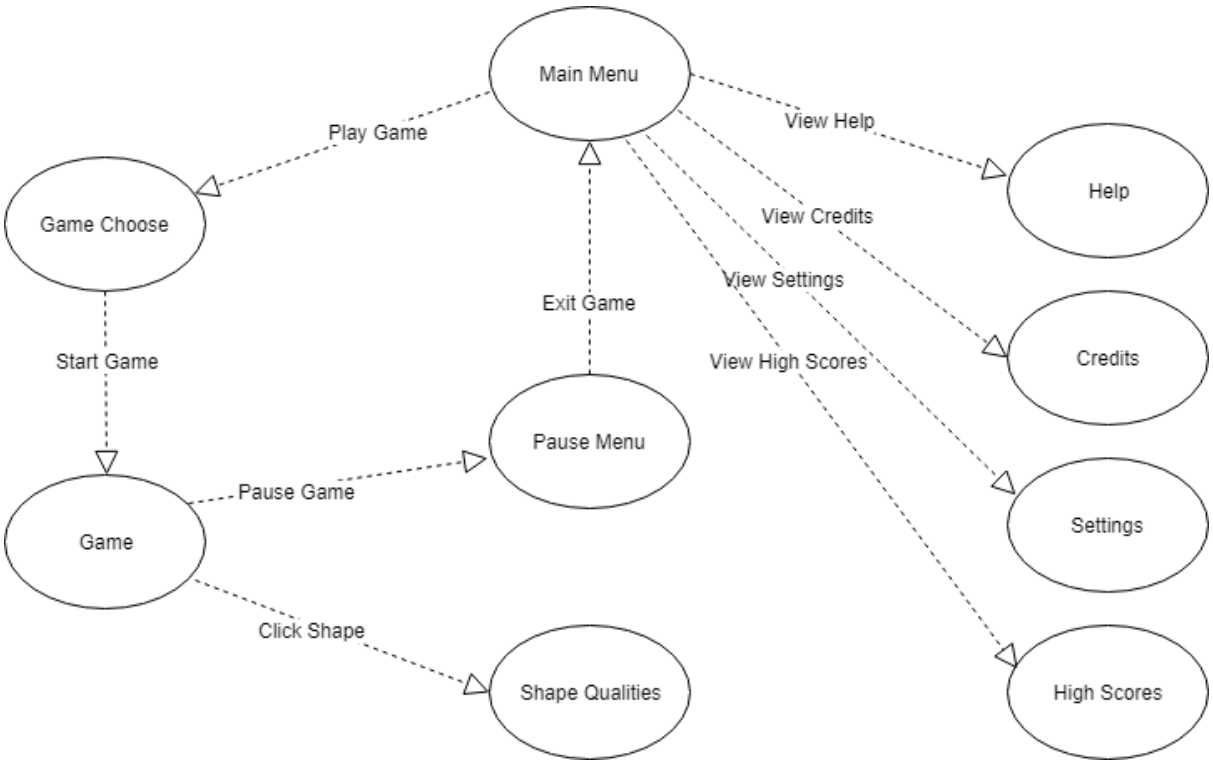
GraphicsManager is a controller for rendered view objects such as Menu, graphical representation of the board and inventory which has number of PieceView objects.



FileManager is the class responsible for save & load game functionality. Saved files structured with Java's FileWriter class and Katamino's internal game logic.

5.4. User interface - navigational paths and screen mock-ups

5.4.1 Navigational Path



Navigational path diagram shows the path/relations between the screens. The program starts with Main Menu. Then when clicked play game, game choose screen appears. In this screen, game can be loaded or new game created. Then by clicking start game, game starts. Game screen appears. From game screen, when pause button clicked, Pause manu can be reached. From pause menu screen, user can exit the game or return the game after a while. On game screen, when clicked any shape, shape qualities screen appears in order to rotate shapes. From main menu, when clicked help, user can view help screen. When clicked credits, user can view Credits screen. When clicked settings, user can view Settings screen. When clicked high scores, user can view High Scores screen.

Figure 5.4.2.1: Main Menu

Main Menu is the welcome screen of the game. There are 6 buttons for starting the game, view help, credits, settings, high scores and exit the game. Also there are exit and wide screen buttons.

When clicked the play game button, there is another popup screen to load a game or start a new game.

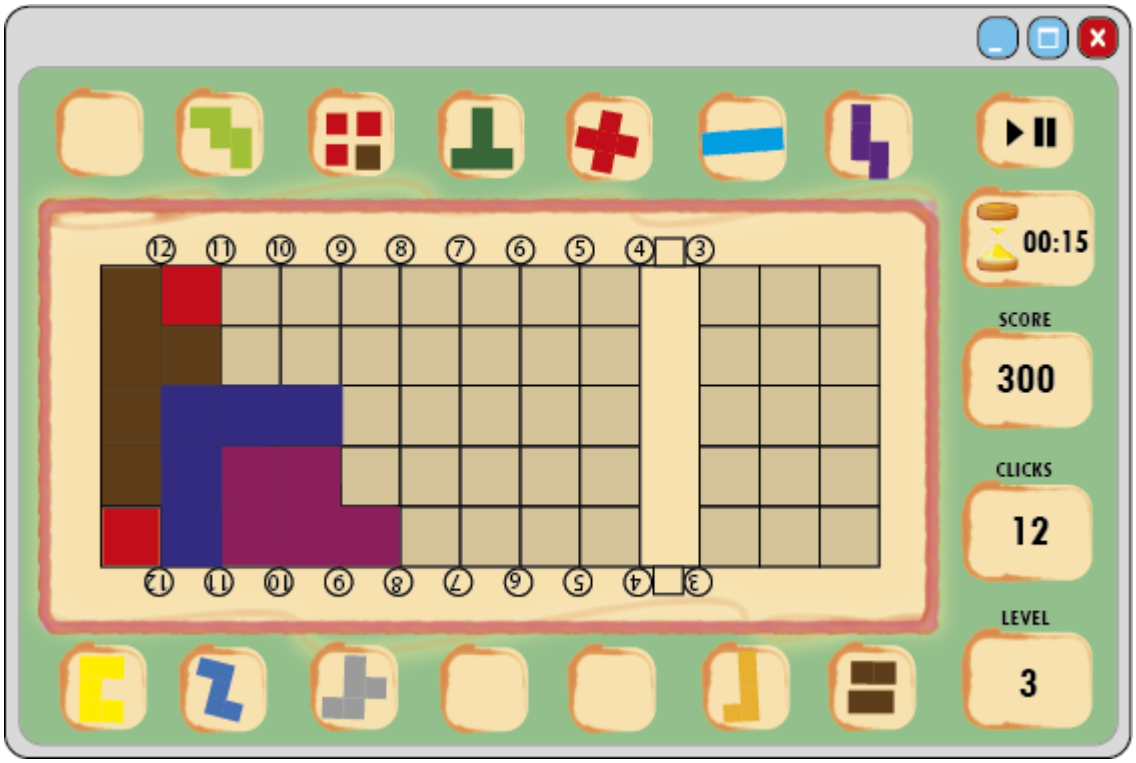


Figure 5.4.2.2: Game Screen

The game screen, shows the user the facilities. It contains table the game is played and shaped to drag. There some information about remained time, level, score and clicks. There is also play/pause button.

Figure 5.4.2.3: High Scores Screen

This screen shows high scores.

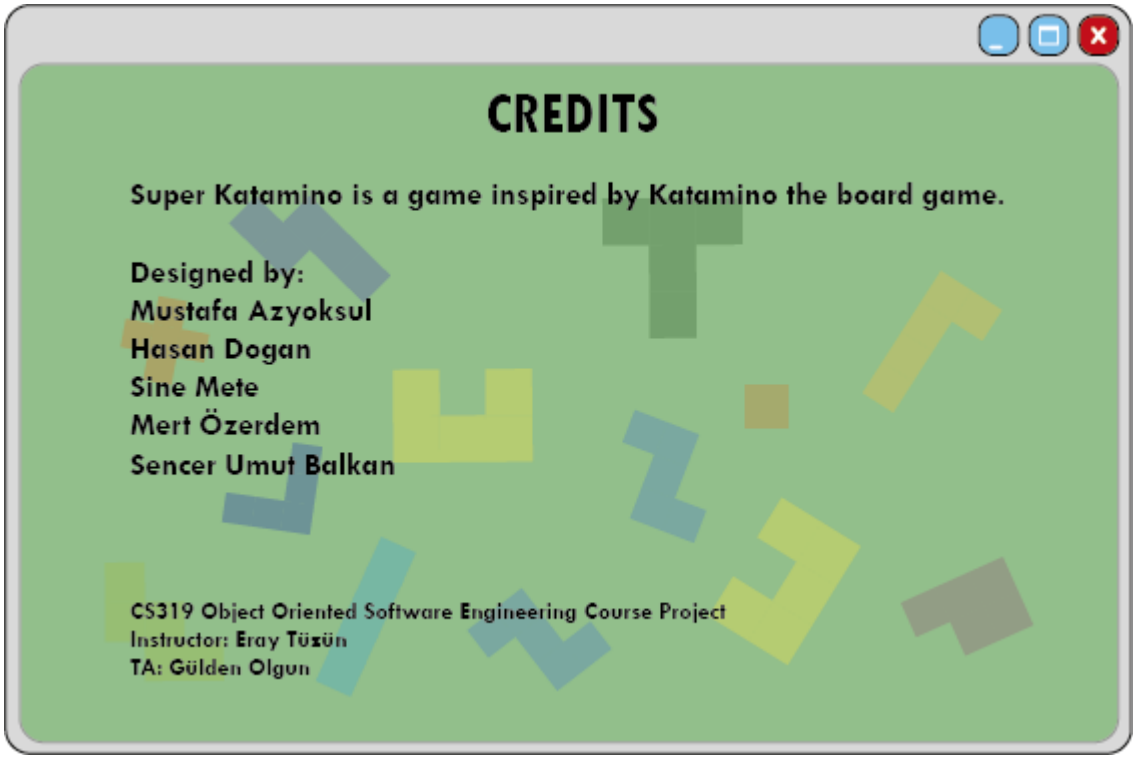


Figure 5.4.2.4: High Scores Screen

This screen shows credits.

Figure 5.4.2.5: Settings Screen

This screen shows settings for sound and music.

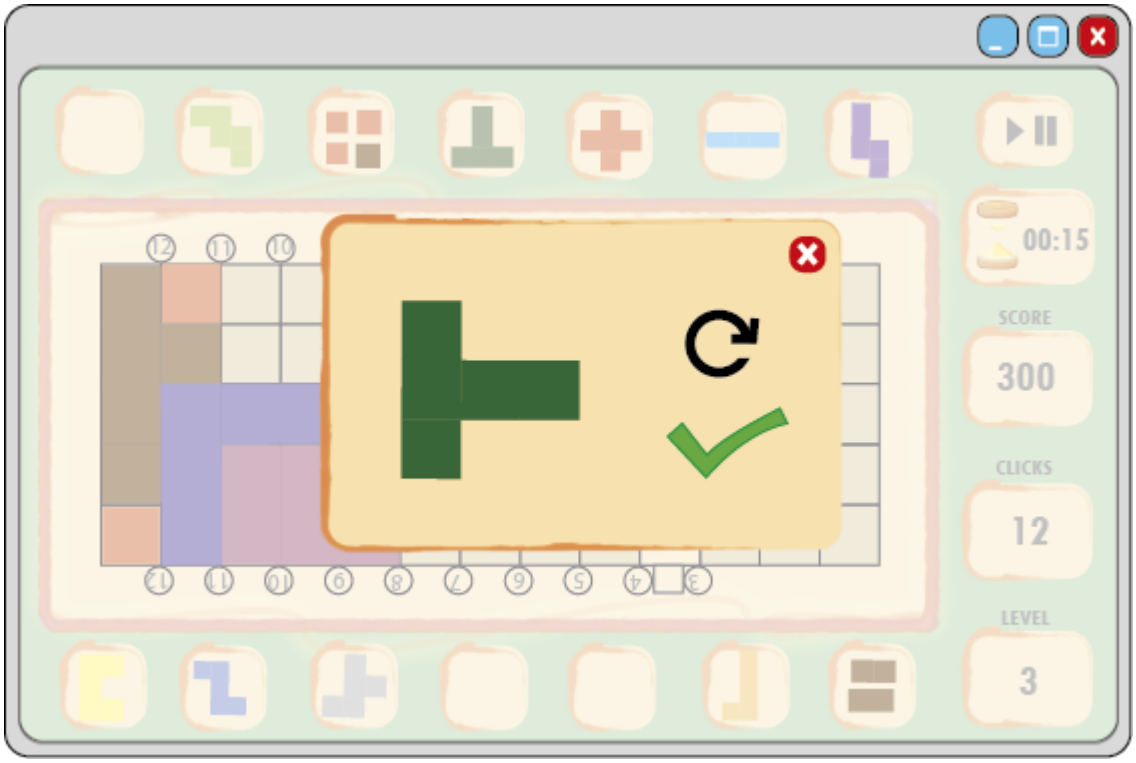


Figure 5.4.2.4: Shape Qualities Screen

When clicked on any shape, this screen appears in order to rotate the shape.

6. Glossary & references

[1] https://boardgamegeek.com/boardgame/6931/katamino