

Team Patroclus G1 Analysis

Table of Contents

1.0 Game Use Cases.....	2
2.0 Level Builder Use Cases.....	8
3.0 Game UML Diagrams	13
4.0 Level Builder UML Diagrams	15
5.0 Game GUI.....	17
6.0 Level Builder GUI.....	19

1.0 Game Use Cases

Use case Name:	Initialize Game
Participating actor:	Player
Flow of Events:	1. Player requests to initialize the game 2. Kabasuji is updated to reflect new state
Entry Condition:	Game has not yet started
Exit Condition:	Flash screen appears, go to the menu

Use case Name:	See past level accomplishments
Participating actor:	Player
Flow of Events:	1. Player requests to see past level accomplishments 2. Kabasuji is updated to reflect new state
Entry Condition:	Game has not yet started and accomplishment appears
Exit Condition:	Go to level accomplishments

Use case Name:	Start game
Participating actor:	Player
Flow of Events:	1. Player requests to start game 2. Kabasuji is updated to reflect new state
Entry Condition:	Game has not yet started
Exit Condition:	Game starts, go to game menu

Use case Name:	Load level
Participating actor:	Player
Flow of Events:	1. Player requests to load level 2. Kabasuji is updated to reflect new state

Entry Condition:	Game has not yet started
Exit Condition:	Level uploaded. Board, bullpen updated

Use case Name:	Go back to initial menu
Participating actor:	Player
Flow of Events:	1. Player requests to go back to initial menu 2. Kabasuji is updated to reflect new state
Entry Condition:	Initial menu is available
Exit Condition:	Initial menu appears, game data is saved.

Use case Name:	Play level
Participating actor:	Player
Flow of Events:	1. Player requests to play level 2. Kabasuji is updated to go to one level
Entry Condition:	At least one available level, which is unlock
Exit Condition:	Go to level, game starts

Use case Name:	Go to next level
Participating actor:	Player
Flow of Events:	1. Player requests to go to next level 2. Kabasuji is updated to go to next level
Entry Condition:	At least two level
Exit Condition:	Go to next level which is available

Use case Name:	Go to previous level
Participating actor:	Player

Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to go to previous level 2. Kabasuji is updated to go to previous level
Entry Condition:	At least two levels
Exit Condition:	Go to last level which is available

Use case Name:	Add Piece
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to add one piece on the board 2. Board or bullpen is updated to reflect new state
Entry Condition:	At least one piece on the board or outside board
Exit Condition:	Piece is added and board data is updated

Use case Name:	Remove Piece
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to remove one piece on the board 2. Board or bullpen is updated to reflect new state
Entry Condition:	At least one piece on the board or outside board
Exit Condition:	Piece is removed and board data is updated

Use case Name:	Move Piece
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to move piece 2. Board is updated to reflect new state
Entry Condition:	At least one piece on the board or outside board, an empty place in the board
Exit Condition:	Piece is moved and board data is updated

Use case Name:	Rotate piece
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to rotate piece 2. Player chooses one direction to rotate 3. Piece is updated to reflect new state
Entry Condition:	At least one piece
Exit Condition:	Piece is rotated, and piece data is updated

Use case Name:	Flip piece
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to rotate piece 2. Piece is updated to reflect new state
Entry Condition:	At least one piece
Exit Condition:	Piece is flipped, and piece data is updated

Use case Name:	Complete Game
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to finish current level 2. Game completes
Entry Condition:	At the end of level
Exit Condition:	Pass the end of level

Use case Name:	See Hint
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to help 2. Kabasuji responds by placing help interface

Entry Condition:	None
Exit Condition:	Help interface appears, game stops

Use case Name:	See Help
Participating actor:	Player
Flow of Events:	1. Player requests to help 2. Kabasuji responds by placing help interface
Entry Condition:	None
Exit Condition:	Help interface appears, game stops

Use case Name:	Collect Number
Participating actor:	Player
Flow of Events:	1. Player requests to move the pieces and cover the number 2. Game is updated with new state
Entry Condition:	Player is playing in releasing mode
Exit Condition:	The collected number is stored into state, the board is updated with the pieces

Use case Name:	Get one or two stars
Participating actor:	Player
Flow of Events:	1. Player requests to get one or two stars 2. Kabasuji responds by updating new state
Entry Condition:	Game starts, player gets one star or two stars requirement
Exit Condition:	One or two stars update, this level is unlocked and current game ends

Use case Name:	Get three stars
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to get three stars 2. Kabasuji responds by updating new state
Entry Condition:	Game starts, player place all or cover all sets
Exit Condition:	Three stars update, this level is unlocked and current game ends

Use case Name:	Terminate Game
Participating actor:	Player
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to terminate game 2. Kabasuji responds by terminating game
Entry Condition:	Game starts
Exit Condition:	All the data is saved, Kabasuji is turned off

2.0 Level Builder Use Cases

Use case Name:	Initialize Level builder
Participating actor:	Builder
Flow of Events:	1. Player requests to initialize Level builder 2. Level builder is updated to reflect new state
Entry Condition:	Level builder has not yet started
Exit Condition:	Go to level editor menu

Use case Name:	Create a new level
Participating actor:	Builder
Flow of Events:	1. Player requests to create a new level 2. Level builder is updated to reflect new state
Entry Condition:	Initialize Level builder
Exit Condition:	New level is created.

Use case Name:	Undo in level builder
Participating actor:	Builder
Flow of Events:	1. Player requests to undo in level builder 2. Level builder is updated to reflect new state
Entry Condition:	Exist last status
Exit Condition:	Undo to the last status

Use case Name:	Save the level build
Participating actor:	Builder

Flow of Events:	1. Player requests to save the level builde 2. Level builder is updated to reflect new state
Entry Condition:	Level builder data is complete
Exit Condition:	Save the data into file

Use case Name:	Delete level build
Participating actor:	Builder
Flow of Events:	1. Player requests to delete level build 2. Level builder is updated to reflect new state
Entry Condition:	At least one complete level
Exit Condition:	Level is deleted successfully

Use case Name:	Select level mode
Participating actor:	Builder
Flow of Events:	1. Player requests to select level mode 2. Level builder is updated to go to select level interface
Entry Condition:	At least one level mode exist
Exit Condition:	One level mode is selected, Level builder changes into select level

Use case Name:	Set time limit
Participating actor:	Builder
Flow of Events:	1. Player requests to set time limit. 2. Level builder is updated to new state
Entry Condition:	In lightning level mode
Exit Condition:	The time limit is set and stored

Use case Name:	Set moves limit
Participating actor:	Builder
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to set moves limit 2. Level builder is updated to new state
Entry Condition:	In puzzle level mode
Exit Condition:	The moves limit is set and stored

Use case Name:	Set colored number
Participating actor:	Builder
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to set colored number 2. Level builder is updated to new state
Entry Condition:	In releasing level mode
Exit Condition:	The colored number is set and stored

Use case Name:	Add color number into a square
Participating actor:	Builder
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to add color number into a square 2. BoardCreator is updated to new state
Entry Condition:	In releasing level mode, at least one square
Exit Condition:	The square with colored number is set and stored

Use case Name:	Build piece
Participating actor:	Builder

Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to build piece 2. The piece and data is updated to new state
Entry Condition:	Player working on building a level
Exit Condition:	Six square is connected

Use case Name:	Delete piece
Participating actor:	Builder
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to delete piece 2. The piece and data is updated to new state
Entry Condition:	At least one piece available
Exit Condition:	Piece is deleted from bullpen

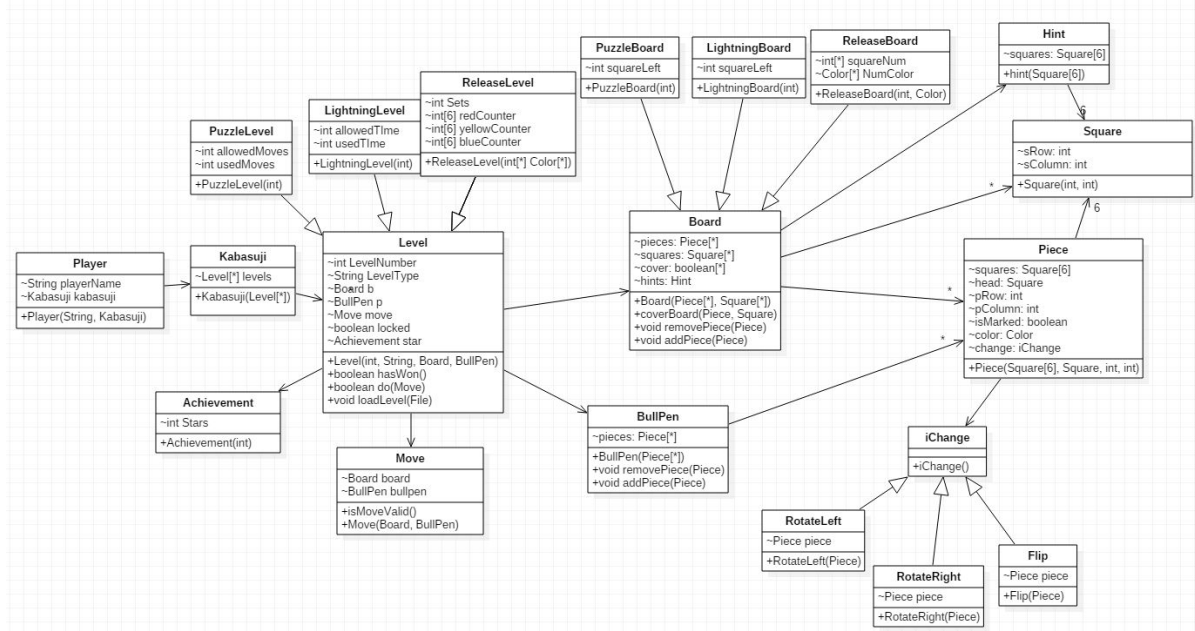
Use case Name:	Build board
Participating actor:	Builder
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to build board 2. The board and data is updated to new state
Entry Condition:	At least one piece
Exit Condition:	Board has 6n squares

Use case Name:	Create hint
Participating actor:	Builder
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to create hint 2. Level builder updates to represent new state
Entry Condition:	Player currently building a Puzzle /or Realease level
Exit Condition:	Level builder shows the hint on the board

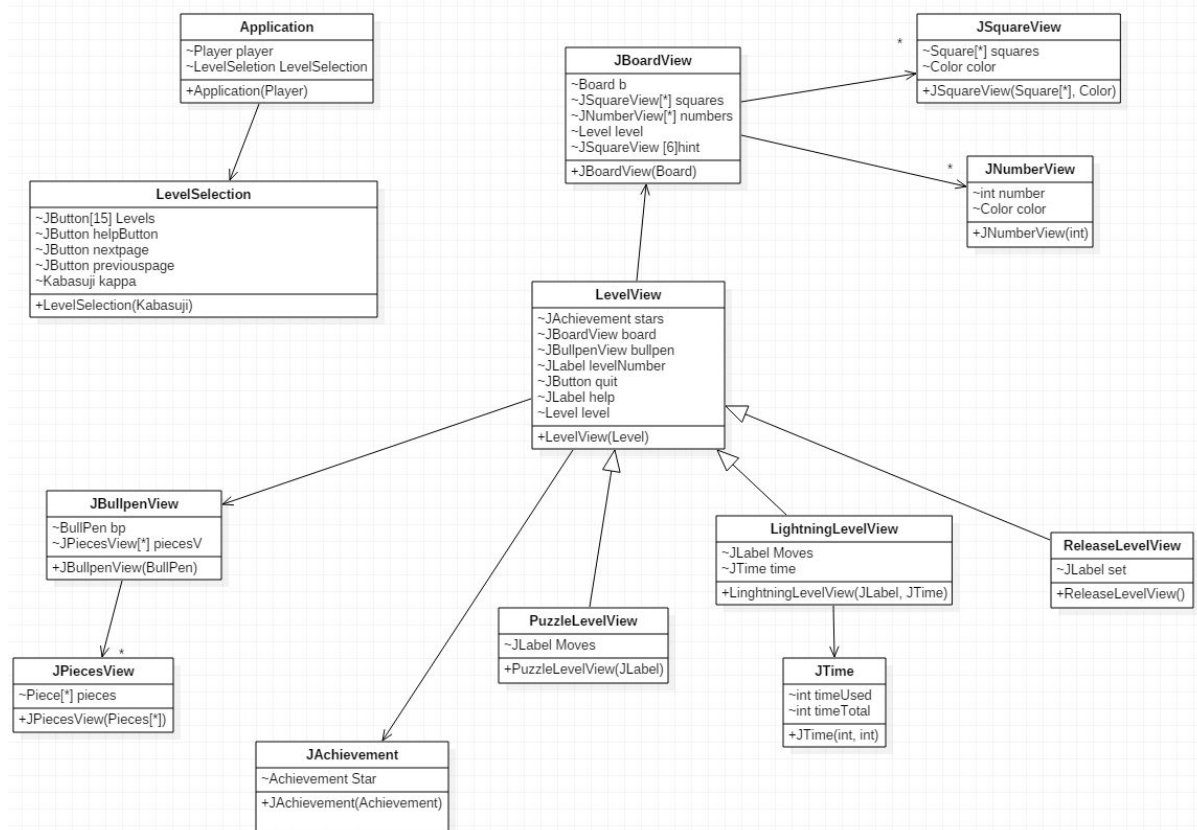
Use case Name:	Complete the level builder
Participating actor:	Builder
Flow of Events:	<ol style="list-style-type: none"> 1. Player requests to quit the level builder 2. Level builder exits
Entry Condition:	Player currently not working on any levels or the current work has been saved
Exit Condition:	Level builder exit with all data saved

3.0 Game UML Diagrams

Entity:



Boundry:



Controller:

PreviousPageController
~LevelSelection menu
~Player player
+actionPerformed()
+PreviousPageController(LevelSelection, Player)

HelpViewController
~LevelSelection menu
+actionPerformed()
+HelpViewController(LevelSelection)

LevelSelectController
~LevelSelection menu
~Player player
+actionPerformed()
+LevelSelectController(LevelSelection, Player)

NextPageController
~MainMenu menu
~Player player
+actionPerformed()
+NextPageController(LevelSelection, Player)

QuitGameController
~LevelView level
~Player player
+actionPerformed()
+QuitGameController(LevelView, Player)

RemovePiecesBoardController
~JBoardView board
~Player player
+actionPerformed()
+RemovePiecesBoardController(JBoardView, player)

MovePiecesController
~JBoardView board
~Player player
+actionPerformed()
+MovePiecesController(JBoardView, Player)

AddPiecesController
~JBoardView board
~Player player
+actionPerformed()
+AddPiecesController(JBoardView, Player)

RemovePiecesBulpenController
~JBulpenView bulpen
~Player player
+actionPerformed()
+RemovePiecesBulpenController(JBulpenView, player)

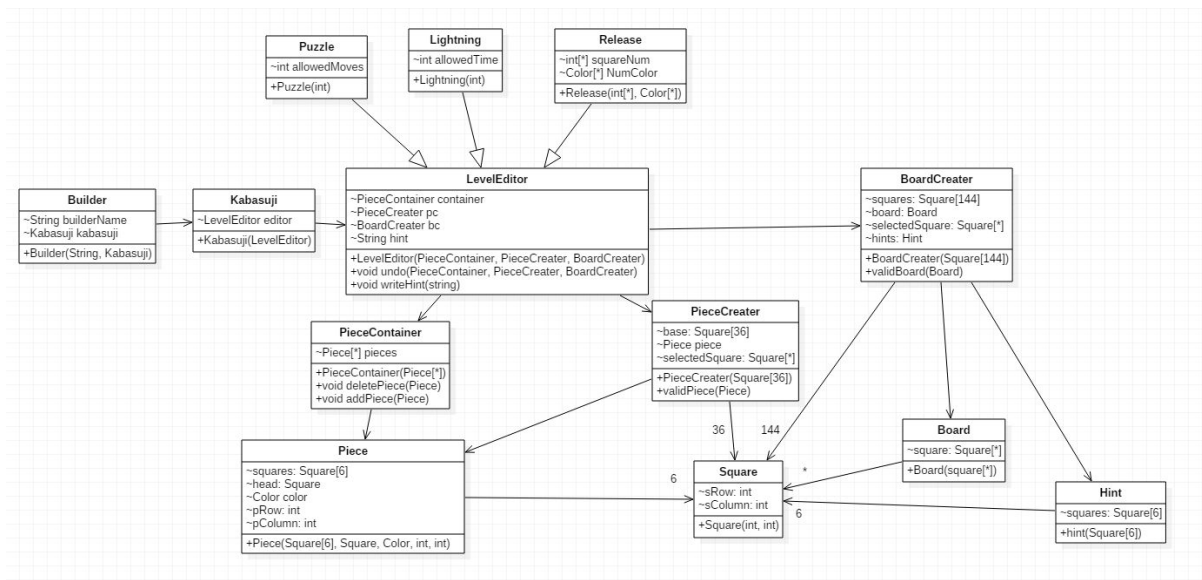
CollectNumberController
~JBoardView Board
~Player player
+actionPerformed()
+ColectNumberController(JBoardView, Player)

PiecesViewController
~JPiecesView piece
~Player player
+actionPerformed()
+PiecesViewController(JPiecesView, player)

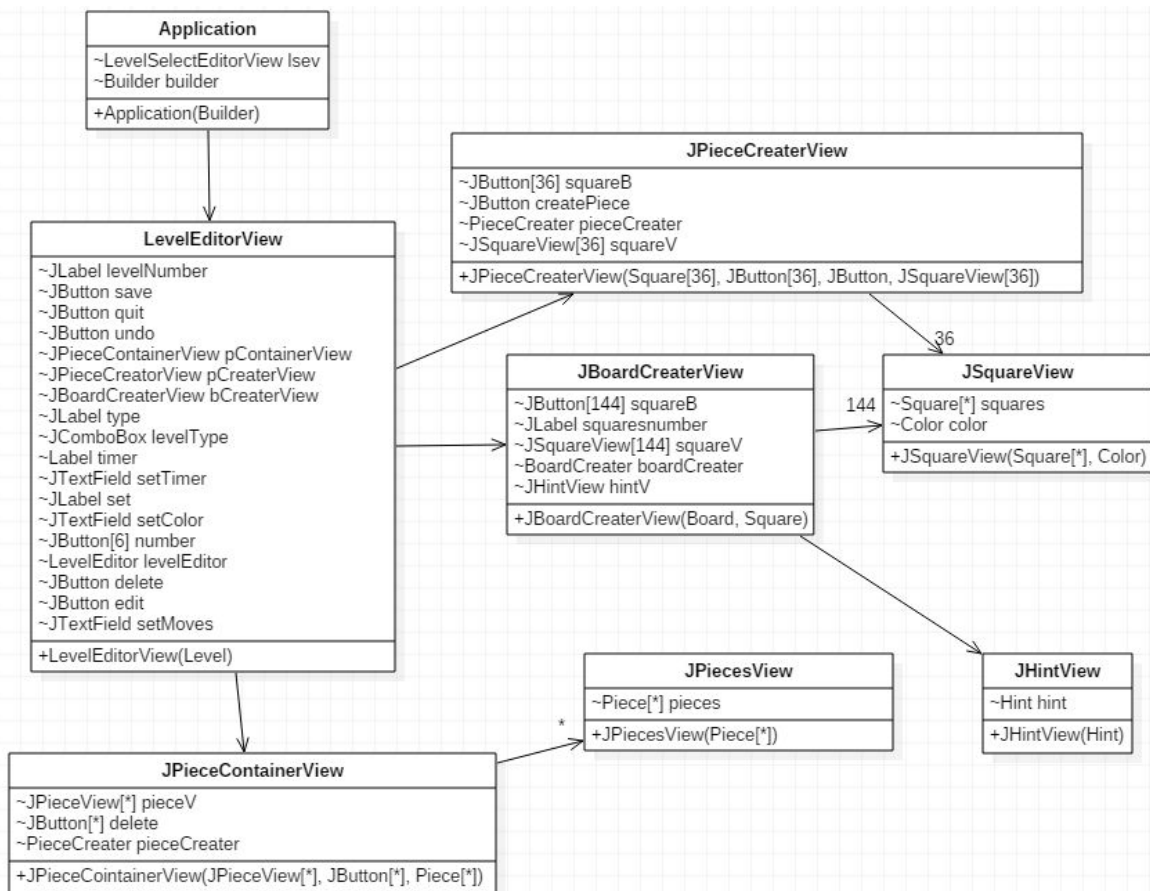
AchievementController
~LevelView level
~Player player
+actionPerformed()
+AchievementController(LevelView, Player)

4.0 Level Builder UML Diagrams

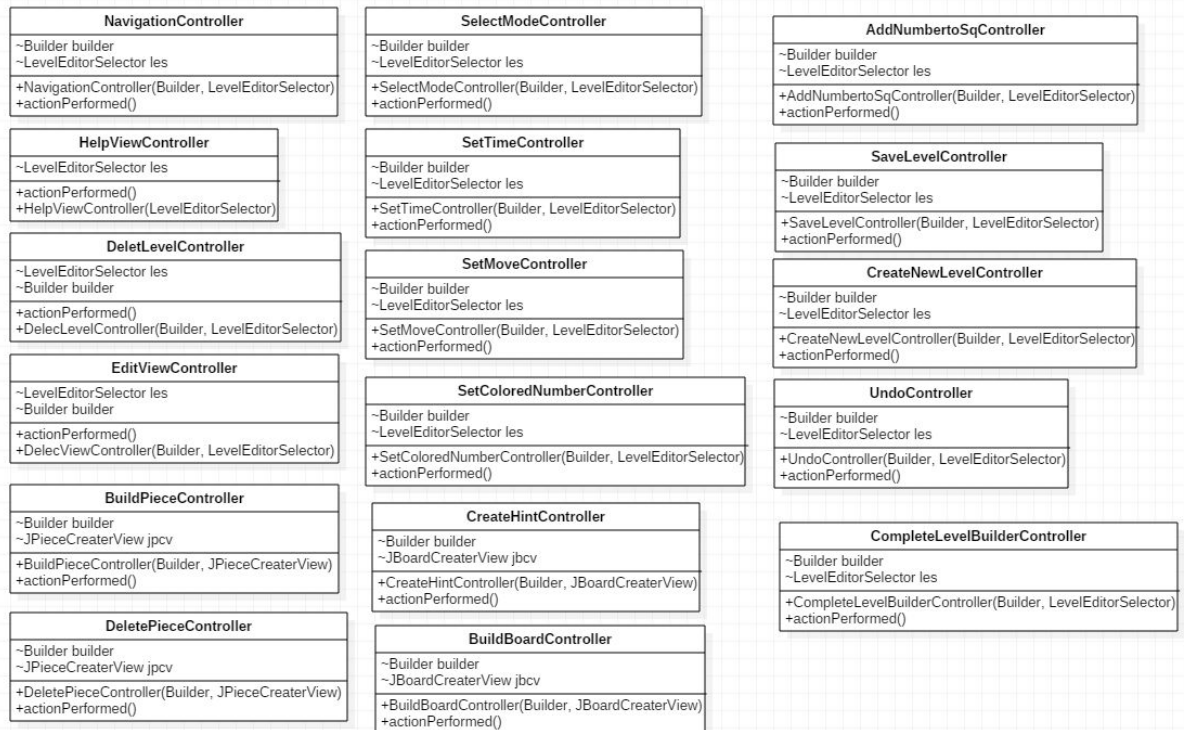
Entity:



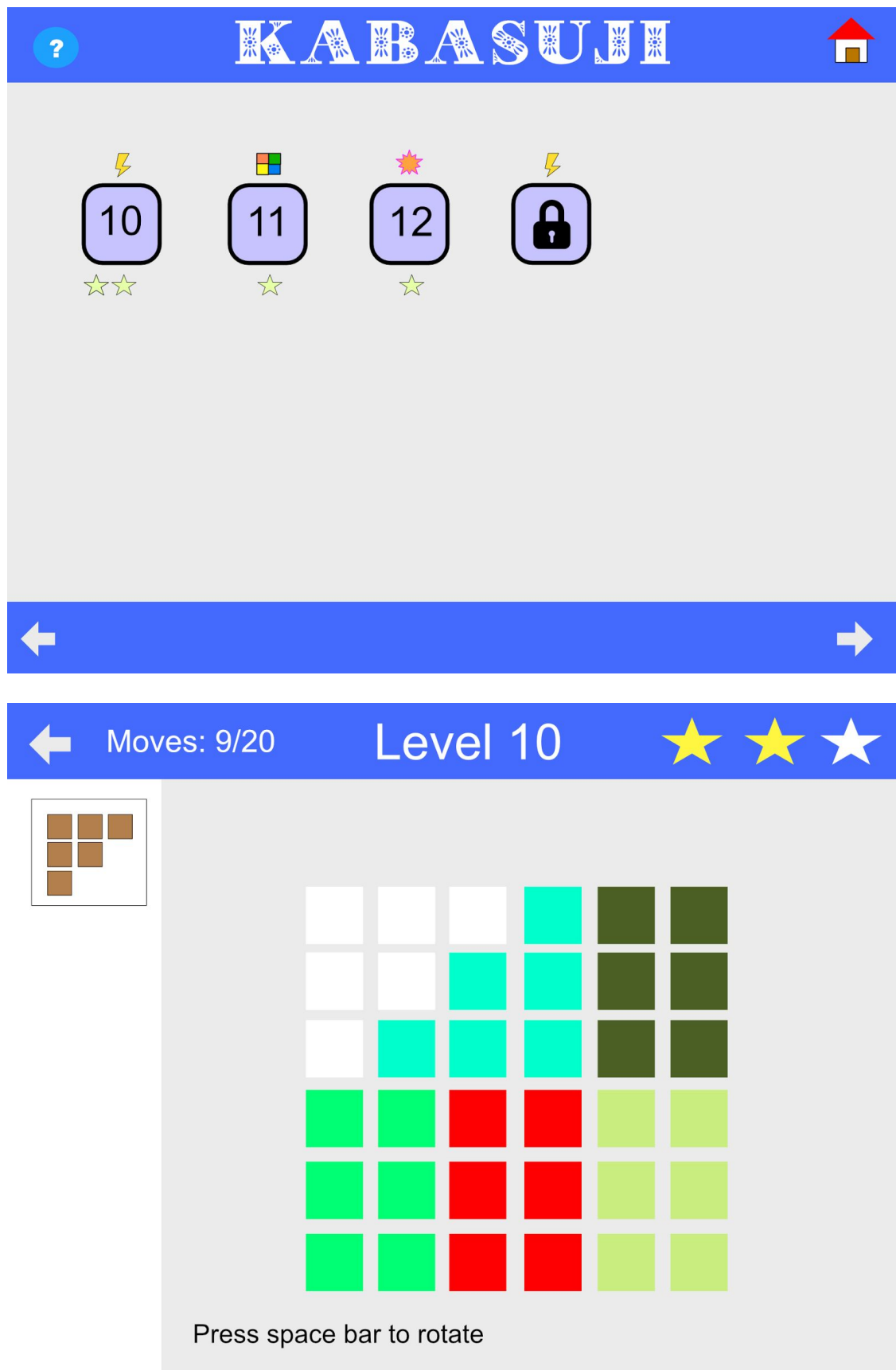
Boundry:

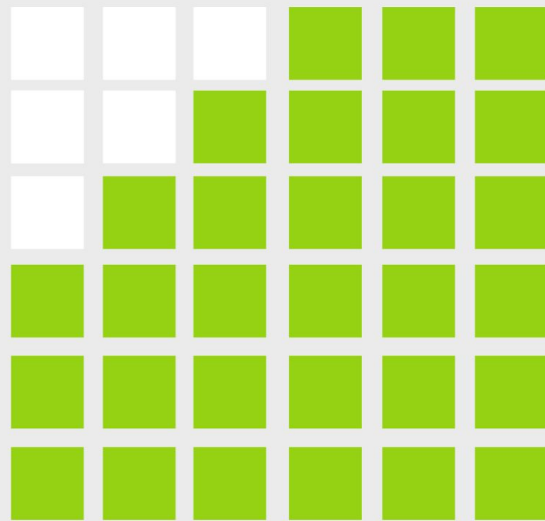
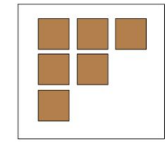


Controller:



5.0 Game GUI





Press space bar to rotate, Up D-pad to flip

6.0 Level Builder GUI

?

KABASUJI

+

10

11

12

LEVELBUILDER (ò_ó) ✨

Level 10 Save

Hint

Place random

Type:

Puzzle ▼

Timer:

0

Set:

1

1

2

3

4

5

6

Moves:

8

Squares: 15

19