**Program Description**

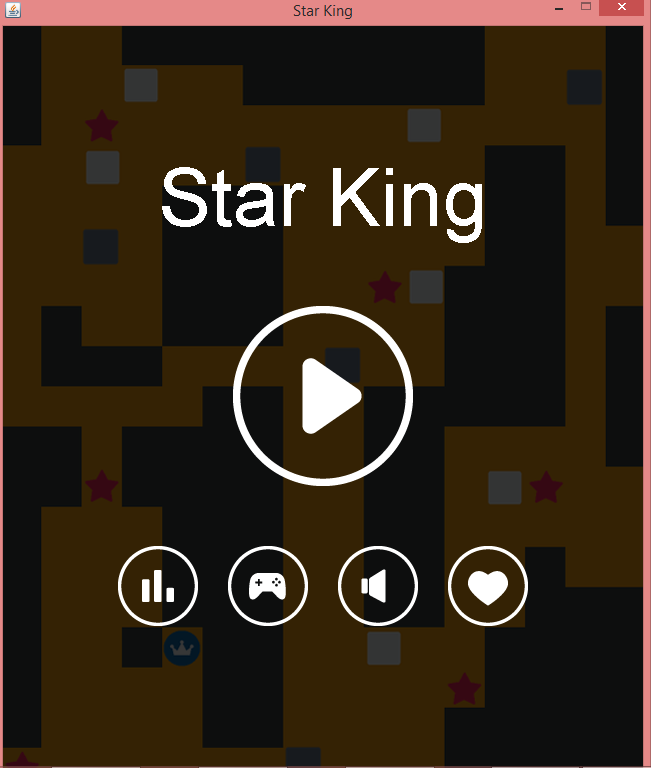
a puzzle game to collected all object in the map with some obstacles.

Play with keyboard and mouse

**How to use**

Here is how to navigate the game

MainScreen



On the first screen you arrive in the game is called GameScreen

There are 5 button to click the largest one is the play button to take you directly to the game

The lower four button from left to right is HighScoreButton,InstructionButton,SoundButton and LikeButton

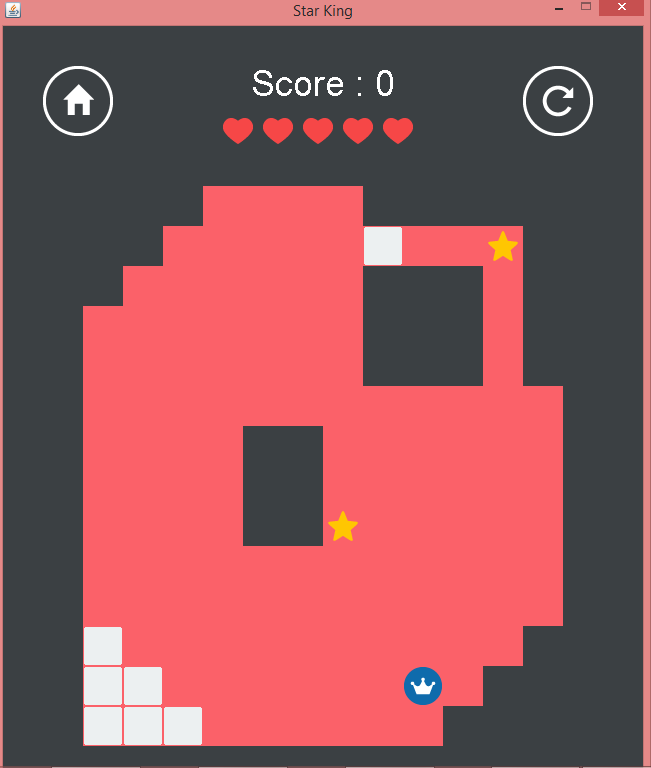
HighScoreButton - let you see the HighScore page. In this page you can press the button in the bottom right to go back to MainScreen

InstructionButton - let you see the Instruction page. There is the same button with HighScoreButton to go back to MainScreen

SoundButton - click to toggle background music on or off

LikeButton - Button for cheat mode

GameScreen



There is two button on the top the left one is HomeButton and the other is RefreshButton

HomeButton - go to MainScreen

RefreshButton - Restart current level but you will lose heart and some score so be careful

Game Instruction

the objective is to get all the star on the map

game is grid of 12 width and 14 height

control the player with keyboard's arrow key whenever you press any direction the player will move in that direction until it can't be moved

In-Game Object

Player – the unit you control

Movable Block – you will stop once you reach them and you can press that direction again to move it and the player by 1 grid

Destructible Block – you will stop once you reach them and it will be automatically destroyed

PointObject – you will move past them and collect them automaticly

Cheat Mode

activated by LikeButton on MainScreen

use enter to clear any leftover character

use cheat by typing then press enter

nextlevel - go to next level

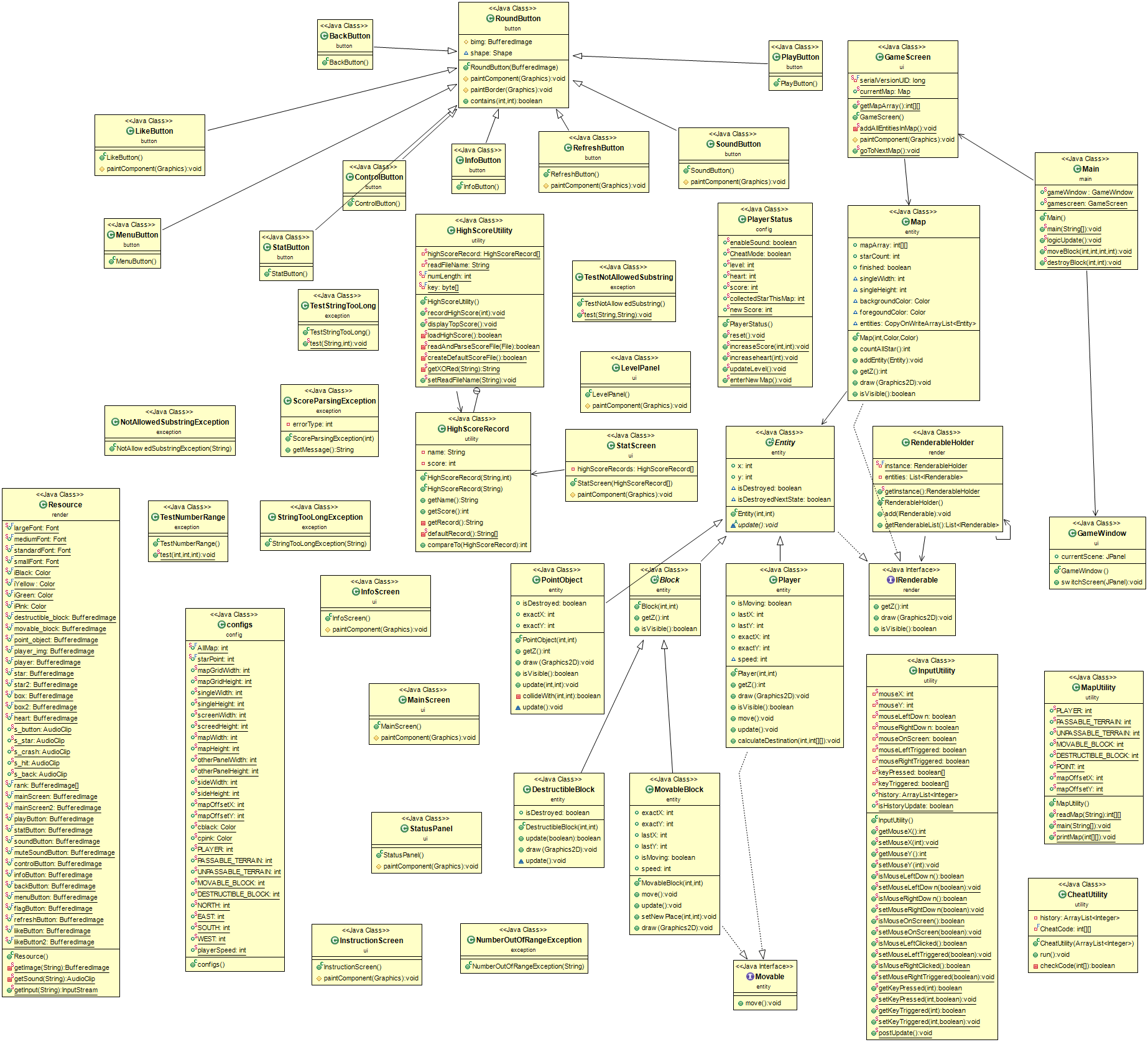
addscrore - add score

addheart - add heart

gotolevel - go to any level

setscore - set your score

**UML**



**Class Description**

**Class button.BackButton**

a RoundButton that takes you back to MainScreen

**Class button.ControlButton**

a RoundButton the leads you to InstructionScreen

**Class button.LikeButton**

a RoundButton to enable cheat mode

**Class button.MenuButton**

Has the same function with button.BackButton but different Icon

**Class button.SoundButton**

a RoundButton that controls background music

**Class button.PlayButton**

a RoundButton that start the game

**Class button.RefreshButton**

A RoundButton that let you redo the current Level but you will lose heart

**Class button.RoundButton**

A RoundButton that has round border

Field

|  |  |
| --- | --- |
| # BufferedImage bimg | icon image of this button |

Constructor

|  |  |
| --- | --- |
| + RoundButton() | constructor that set the size to circle and not to paint background |

Methodology

|  |  |
| --- | --- |
| # void paintComponent(Graphics g) | Paint this component |
| # void paintBorder(Graphics g) | do nothing just overridden method |
| + boolean contain() | If the button has changed size make a new shape object. |

**Class button.StatButton**

a RoundButton that take you to StatScreen

**Class config.configs**

this class hold major of the setting of the game

**Class config.PlayerStatus**

this class hold variables of player such as level, heart and score

It also has method to control these variables.some of the method are thread-safe

**Class entity.Block**

abstract class for blocks

**Class entity.DestructibleBlock**

a block that can be destroyed when player bumped them

Field

|  |  |
| --- | --- |
| + boolean isDestroyed | variable to check if this block is destroyed by player |

Method

|  |  |
| --- | --- |
| + boolean update(boolean isMoving) | update the isDestroyed variable according to Player.isMoving |

**Class entity.Entity**

abstract class for all entity

**Class entity.Map**

class the represent the map with array and and list of entities

Field

|  |  |
| --- | --- |
| + int[][] mapArray | array of map |
| + int starCount | number of stars in map |
| + CopyOnWriteArrayList<Entity> entities | ArrayList to hold all entites on this map |

Method

|  |  |
| --- | --- |
| + int countAllStar() | count all of the star in mapArray |

**Class entity.MovableBlock**

block that can be moved if player is next to them and there are nothing blocking

Method

|  |  |
| --- | --- |
| + void move() | change that exact value that doesn't equal final value to get closer |
| + void update() | update this entity position |
| + void setNewPlace(int finalX,  int finalY) | setter of lastX and lastY |

**Class entity.Player**

Player entity of this game

Method

|  |  |
| --- | --- |
| + void move() | change that exact value that doesn't equal final value to get closer |
| + void update() | update this entity position |
| + void calculateDestination(  int direction,int[][] mapArray) | calculate lastX and lastY |

**Class entity.PointObject**

Point Object that is to be collected

Method

|  |  |
| --- | --- |
| + void update(int playerExactX,  int playerExactY) | check if player collected this entity |
| - boolean collideWith(  int playerExactX, int playerExactY) | check collision with player |

**Class main.Main**

Main class to run game

Method

|  |  |
| --- | --- |
| + static void main(String[] args) | main function initial gameWindow and gameloop |
| + static void logicUpdate() | game logic update |
| + static void moveBlock(  int currentX, int currentY,  int lastX,int lastY) | move movableBlock |
| + static void destroyBlock(  int currentX, int currentY) | destroy destructibleBlock |

**Class render.RenderableHolder**

this class hold all entities in this game

**Class render.Resource**

This class have major of resources of this game imported by ClassLoader

**Class ui.GameScreen**

GameScreen Panel

Method

|  |  |
| --- | --- |
| - static void addAllEntitiesInMap() | add all entities in currentMap to the RenderableHolder |
| + static void goToNextMap() | update map to be nextMap,this method is thread-safe |

**Class ui.GameWindow**

Main frame of the game

Method

|  |  |
| --- | --- |
| + void switchScreen(JPanel screen) | switch currentScene panel |

**Class ui.InstructionScreen**

InstructionScreen Panel

**Class ui.LevelPanel**

Level Panel on GameScreen

**Class ui.MainScreen**

MainScreen Panel

**Class ui.StatScreen**

StatScreen panel

**Class ui.StatusPanel**

StatusScreen Panel

**Class exception.ScoreParsingException**

Exception for score parsing

**Class exception.TestNumberRange**

Test if number is out of range

**Class exception.NumberOutOfRangeException**

return number out of range message

**Class exception.TestNotAllowedSubstring**

Test if forbidden character exist

**Class exception.NotAllowedSubstringException**

return false string message

**Class exception.TestStringTooLong**

Test if String is too long

**Class exception.StringTooLongException**

return String too long message