RAID THE BOXES

About the game

What you were supposed to do in this game is to kill all of monster until you die. Every monster you had killed you will get a score to make new high score record, have fun and save the world !!

How to play

You will be the knight that has to kill a Dracula to get the score (level). The Dracula will be created automatically if it die.

You will have to press the spacebar when the moving bar is on any boxes.

* If you hit blackbox you will attack( your’s attack points is 10 )
* If you hit greenbox you will get heal for 10 points.
* If you hit purplebox, the redbox will be stopped until you hit blackbox.
* If the moving redbox had bounced for 3 times, you will be attacked by Dracula (the default attack is 2, and will be increased every time Dracula dies by 1).



Figure 1: to show intro screen with start button and high score button, so you can choose to start or see the record.

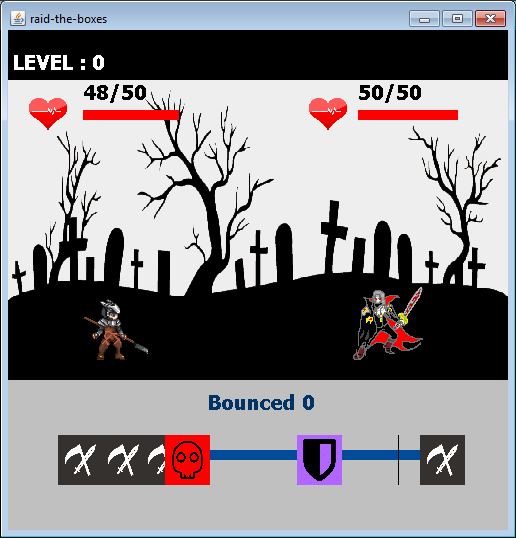


Figure 2: to show the game screen containing knight, monster and box’s bar.

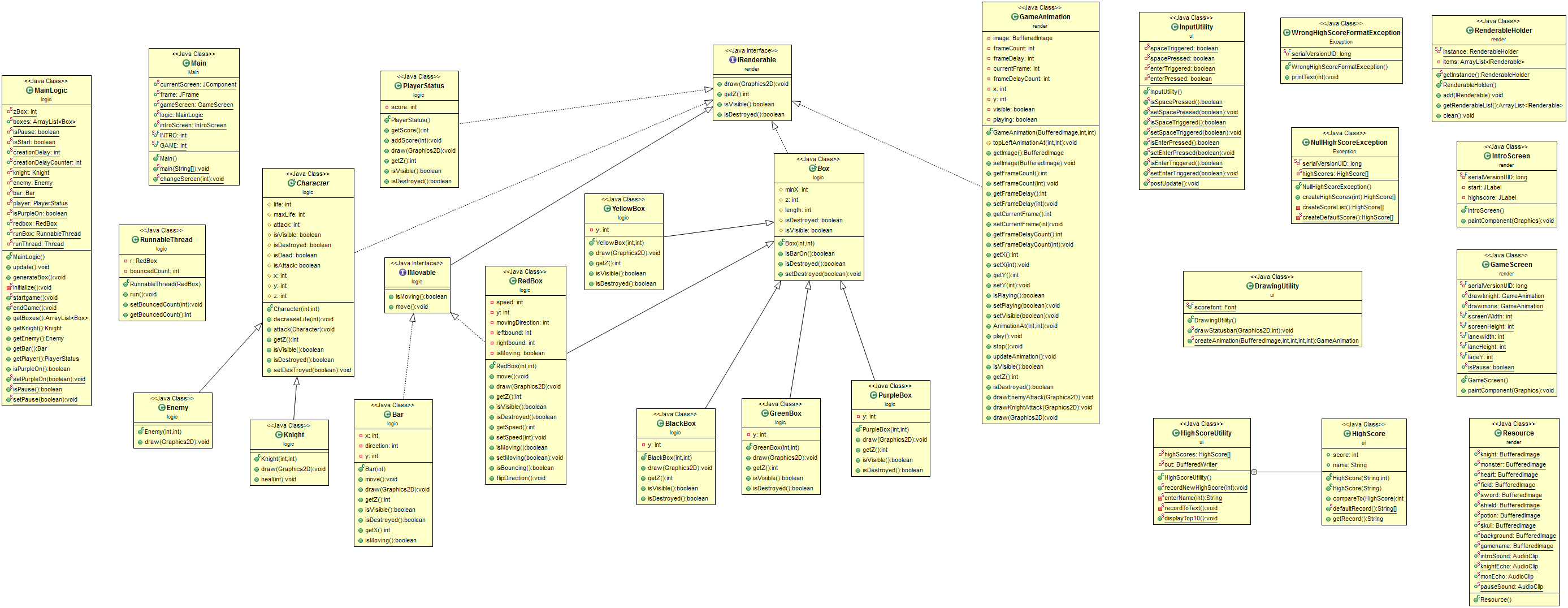


Figure 3: UML diagram

**BASIC GAME INFORMATION**

Player: you will be the knight that has to kill monsters.

* You have max life 50 points.
* Your attack is 10 points.
* You were supposed to fight until you die.

Enemy:

* Each enemies you killed you get 1 score.
* At first enemy’s attack is 2 points, and will be increased every time it had died 1 point each.

Boxes: there are 4 types of boxes which are listed below

Red box: It is always move toward and bounce within the border of game screen however you cannot do anything to destroy them. You can only stop them by clicking any purple box. Furthermore the enemies will attack you if the red box had bounced three times.

Green potion box: It is randomly created. By the way, if it had been destroyed your life will be increased 10 points.

Black sword box: It is randomly created as well. If it had been destroyed you will attack the enemy.

Purple shield box: It is randomly created. If it had been destroyed, the red box will be stopped until you hit any black sword box, then the red box will move again.

Bar: there is the bar that is always moving around the lane to indicate the time to destroy the boxes.

**CLASS INFORMATION**

Part Logic

Abstract Class Box

Field

* int minX ; x position of each boxes to create.
* int z ; z position of each boxes.
* int length ; the length of each boxes.
* boolean isDestroyed ; to check if this box is destroyed or not.
* Boolean isVisible ; to check if the box have to be drawn.

Contructor

* Box(int length, int z) ; to set the box’s length and z position.

Method

* Getter & setter of isDrstroyed
* Void isBarOn() ; check if that the bar is on. If yes, return true

Class RedBox : Extend Box and implement IMovable

Field

* int y ; y position of this box.
* int speed ; speed of this box.
* int movingDirection ; if moving direction = 1 then moving left and if moving direction = -1 then moving right
* int leftbound ; the left position of gamescreen.
* Int rightbound ; the right position of gamescreen.
* boolean isMoving ; to check if this box is moving or not.

Contructor

* RedBox(int speed, int lenght) ; to set the box’s length which is always be 45 and speed. In addition to default minX to the right of the lane, isMoving to be true and movingDirection to be 1.

Method

* Getter & setter of isMoving and speed.
* Void Move() ; to move the box by speed each times
* Void Draw(Graphics2D g2) ; to draw the red box with red recutant , skull icon and display bounced times.
* Int getZ() ; to get z position of red box.
* Boolean isVisible() ; return isVisible
* boolean isDestroyed() ; return isDetroyed
* boolean isBouncing() ; to check if minX is out of the frame
* void flipDirection() ; to change the direction of the box

Class GreenBox : Extend from Box

Field

* int y ; y position of this box.

Contructor

* GreenBox(int length, int z) ; to set the box’s length which is always be 45 and z position up to Class Box.Then default y position to be lane position and x position to be randomly created within the lane.

Method

* Void Draw(Graphics2D g2d) ; to draw the green box with green recutant and potion icon.
* Int getZ() ; to get z position
* boolean isVisible() ; return isVisible
* boolean isDestroyed() ; return isDetroyed

Class PurpleBox : Extend from Box

Field

* int y ; y position of this box.

Contructor

* PurpleBox(int lenght, int z) ; to set the box’s length which is always be 45 and z position up to Class Box. Then default y position to be lane position and x position to be randomly created within the lane.

Method

* Void Draw(Graphics2D g2d) ; to draw the purple box with purple recutant and shield icon.
* Int getZ() ; to get z position
* boolean isVisible() ; return isVisible
* boolean isDestroyed() ; return isDetroyed

Class BlackBox : Extend from Box

Field

* int y ; y position of this box.

Contructor

* BlackBox(int lenght, int z) ; to set the box’s length which is always be 45 and z position up to Class Box. Then default y position to be lane position and x position to be randomly created within the lane.

Method

* Void Draw(Graphics2D g2d) ; to draw the black box with black recutant and sword icon.
* Int getZ() ; to get z position
* boolean isVisible() ; return isVisible
* boolean isDestroyed() ; return isDetroyed

Abstract Class Character

Field

* int life ;character’s life points
* int maxLife ; character’s max life points
* int attack ; character’s attack points
* boolean isVisible ; true if character is visible
* boolean isDestroyed ; true if character is destroyed
* boolean isDead ; true if character is dead
* boolean isAttack ; true if character is attacking
* int x : character’s x position
* int y : character’s y position
* int z : character’s z position

Contructor

* Character(int attack,int maxLife);to set character’s maxLife and attack .Then default isDestroyed, isAttack, isDead to be false and isVisible to be true.

Method

* Void decreaseLife(int amount) ; to decrease life by amount. If life is out then isDead be true.
* Void Attack(Character C) ; call method decreaseLife and set isAttack to be true
* int getZ() ; to get z position
* boolean isVisible() ; return isVisible
* boolean isDestroyed() ; return isDetroyed

Class Knight: extends from Character

Constructor

* Knight(int attack, int maxLife) ; to set default of attack and maxLife using super() and set life equals maxLife.

Method

* Void heal(int amount) ; to increase life by amount and it cannot more than maxLife.
* Void draw(Graphics2D g2) ; to draw knight image
  + when knight is attack, draw animation attack
  + when knight is dead or destroyed draw dead
  + Else draw knight stands. And you must draw knight status such as life every ticks

Class Enemy: extends from Character

Constructor

* Enemy(int attack, int maxLife) ; to set default of attack and maxLife using super() and set life equals maxLife.

Method

* Void draw(Graphics2D g2) ; to draw enemy image
  + When enemy is attack, draw animation attack
  + when enemy is dead or destroyed draw dead
  + Else draw enemy stands. And you must draw enemy status such as life every ticks

Interface Class IMovable extends from IRenderable

Method

* Boolean isMoving() ;
* Void move() ;

Class Bar

Field

* Int x ;
* Int y ;
* Int direction ;

Constructor

* Bar (int x) ; to set x and y to be on the lane and set direction to be 1

Method

* Void move() ; to move x 5 pixels each times and if bar is bounced then flip the direction
* Void draw (Graphic2D g2d) ; to draw the bar and the smaller bar, easier to play
* int getZ() ; to get z position return MAX VAULE.
* boolean isVisible() ; always true
* boolean isDestroyed() ; return false

Class PlayerStatus

Field

* int score ; the player score is level of monster

Constructor

* PlayerStatus() ; to initialize the score to be 0;

Method

* Getter & setter
* Void addScore(int add) ; to increase score by add amount.
* Void draw (Graphics2D g2) ; drawStatusbar from DrawingUtility
* int getZ() ; to get z position return -9999 to draw it below everything but background image
* boolean isVisible() ; return isVisible
* boolean isDestroyed() ; return isDetroyed

Class RunnableThread using Runnable as an interface

This class is been used for making RedBox run

Field

* int bouncedCount ; how many times the box had bounced
* RedBox r ; the redbox which is running

Constructor

* RunnableTread(RedBox r) ; to initialize the value of redbox , bouncedCount.

Method

* Getter & setter
* Void run () ;While the redbox is not destroyed
  + make the redbox move by using synchronized block object and try catch block.
  + if it is not moving make thread wait until get notify (r.wait()).
  + If the redbox is bounced, increase bouncedCount and flipDirection.

Class MainLogic : main logic that running this game

Field

* static int *zBox* ; static z value for generate Box
* static ArrayList<Box> *boxes* ; Arraylist of all Box
* static boolean *isPause*; to check if game is pausing or not
* static boolean *isStart*; true if game is start or not
* static int *creationDelay*; delay to create Box
* static int *creationDelayCounter*; delay counter to create Box
* static Knight *knight*; Knight in game
* static Enemy *enemy*;Enemy that is now in game
* static Bar *bar*;Bar in game
* static PlayerStatus *player*; PlayerStatus in game
* static boolean *isPurpleOn*; to check if there is PurpleBox in screen or not
* static RedBox *redbox* ; RedBox in game
* static RunnableThread *runBox ; RunnableThread for redbox*
* static Thread *runThread* ; Thread to move red box

Constructor

* MainLogic();initialize all field

Method

* update() ;
  + to update the game when it is start and not pause
    - check if enter is triggered : if yes pause game
    - reset all character isAttack
    - move bar
    - check if space bar is triggered : if yes check what thing that bar is on to do it effect and set it destroy value
    - check if redbox’s bounce count equal 3: if yes enemy attack knight
    - check if creation delay counter = creation delay : if yes
      * check if character is dead : if yes set it to be destroy after sleep
      * create new enemy when the old one is destroyed
      * change screen to intro screen if game is over
  + if game is paused
    - check if enter is triggered : if yes unpause
* generateBox() ; randomly create Box and add 1 to zBox value
* initialize() ; initialize all field , create new redbox and add all IRenderable object to RenderableHolder
* startGame() ; start new game
* endGame() ; end the game, destroy redbox and clear IRenderableHolder
* GETTERs & SETTERs

Part ui

Class InputUtility

Field

* static boolean spaceTriggered ; to know if the spacebar is triggered, that mean there is no different between one press or continuously press.
* static boolean spacePressed ; to know if the spacebar is pressed.
* static boolean enterTriggered ; to know if the enter button is triggered.
* static boolean enterPressed ; to know if the enter button is pressed.

Method

* getters & setters
* static postUpdate() ; to update all of triggered value to be false

Class DrawingUtility

Field

* static final Font scorefont ; to set font of the score with white “Tahoma” size 20 and bold.

Method

* static void drawStatusBar(Graphic2D g2, int score) ; used to draw player’s score with black square 500\*50.
* static GameAnimation createAnimation (BufferedImage imahe, int x, int y, int frameCount, int frameDelay) ; to create and return GameAnimation, set default of position and play GameAnimation.

Class HighScore : it is innerclass in class HighScoreUtility

Field

* int score : score of the high score
* String : name : name of the high score maker(can’t have no letter name or have “:” in it)

Constructor

* HighScore(String s,int n) ; initialize name and score
* HighScore(String record) ; initialize name and score by String in record format <aaa:000> (aaa for name and 000 for score) if it is in wrong format set name to be “player” and score be 0
* DefaultRecord;return Array of String that contain default top 10 record
* getRecord();return String in record format

Class HighScoreUtility : utility about high score

Field

* static HighScore[] highScores : Array of highScore to store top 10 HighScore
* static BufferedWriter out : BufferedWriter to record the top 10 highScore in text file

Method

* recordNewHighScore(int scoreIn);check if score in is a highScore : if yes let player input name and record the highScore in both highScores and text file
* enterName(int rank);create input dialog for player to input name (also check if it is in right format)
* recordToText();record top 10 high score from highScores into text file
* displayTop10();display top 10 high score in new message dialog

Part Exception

Class NullHighScoreException: extends class Exception,have been thorwn when highscores in HighScoreUtility is null or have null HighScore or text file have wrong format

Field

* static HighScore[] highScores; Array of top 10 highscore

Method

* createHighScores(int type);
  + if type is 1(text file have wrong format): return createDefaultScore()
  + otherwise(highScores in HighScoreUtility is null): return createScoreList()
* createdeScoreList();return array of HighScore that contain highscore from text file
* createdefaultScore();return array of HighScore that contain default top 10 high score record

Class WrongHighScoreFormatException: extends class Exception, have been thrown when player input unusable name and when highscore text file has wrong format

Method

* printText(int type);
  + if type is 1(unusable name):show warning message
  + if type is 2(text file has wrong format):show warning message

Part render

Class IntroScreen: extends from JPanel (Figure 1)

Field

* JLabel start; the label had been using as start button with Bold Font “Verdana” size 30.
* JLabel highscore ; the label had been using as high score button to list top 10 high score players

Constructor

* IntroScreen() ; default all the value and set size as background picture from Resource Class. Then add mouselistener to start and highscore.
  + If start had been clicked, change the screen to GameScreen by Main.changeScreen
  + If highscore had been clicked, display top 10 player lists by HighScoreUtility.displayTop10

Method

* Void paintComponent(Graphics g) ; to draw background picture and gamename picture from Resource Class

Class GameScreen: extends from JComponent (Figure 2)

Field

* Static GameAnimation drawknight ; to create the initial GameAnimation for knight’s attack.
* Static GameAnimation drawmons ; to create the initial GameAnimation for enemy’s attack.
* Static final int screenWidth ;
* Static final int screenHeight ;
* Static final int landwidth ;
* Static final int landHeight ;
* Static final int laneY ; y position of the lane
* Static boolean isPause ; to check if the game is paused

Constructor

* GameScreen() ; set default size by screenWidth and screenHeight and set Visible to be true.

Method

* Void paintComponent(Graphics g) ; to set the default drawing as field, lane, and draw everything in RenderableHolder here if it is visible.

Class GameAnimatio: have IRenderable as interface

Field

* BufferedImage image ; get an image
* Int frameCount ; to define how many frame for an animation
* Int frameDelay ; framedelay for animation
* Int currentFrame ; to know the current frame to draw and check with frameCount
* Int frameDelayCount ; to check if it equals frameDelay to make sure that the animation will be too fast
* Int x ; x position of an animation
* Int y ; y position of an animation
* Boolean visible ;
* Boolean playing ;

Constructor

* GameAnimation(BufferedImage image, int frameCount, int frameDelay) ; default all of vaule and currntFrame to be 0

Method

* getters $ setters
* void topLeftAnimationAt(int x, int y) ; to default x and y position of an animation.
* Void play() ; set currentFrame to be 0, playing and visible to be true to start an animation
* Void stop() ; set currentFrame to be 0, playing and visible to be false. It is the end of an animation
* Void updateAnimation() ; if playing is true, and if frameDelayCount is more than 0 then frameDelayCount decrease by 1 and return method. If not more than 0, then reset frameDelayCount and go to next frame. If there is no next frame anymore, then stop an animation.
* Void drawEnemyAttack(Graphics2D g2) ;

To draw attack animation, use subimage of the image

* Void drawKnightAttack(Graphics2D g2) ;

To draw attack animation, use subimage of the image

* Void draw(Graphics2D g2) ; do nothing because we already done from the method above
* int getZ() ; to get z position return 0
* boolean isVisible() ; return isVisible
* boolean isDestroyed() ; return isDetroyed

Class Resource

Field

* static BufferedImage knight ;
* static BufferedImage monster ;
* static BufferedImage heart ;
* static BufferedImage field ; gamescreen picture
* static BufferedImage sword ; blackbox picture
* static BufferedImage shield ; purplebox picture
* static BufferedImage potion ; greenbox picture
* static BufferedImage skull ; redbox picture
* static BufferedImage background ; introscreen picture
* static BufferedImage gamename ;
* static AudioClip introSound ; introscreen sound
* static AudioClip knightEcho ; knight’s attack sound
* static AudioClip monEcho ; enemy’s attack sound
* static AudioClip pauseSound ; it is loud when you had paused the game.

Interface Class IRenderable

Method

* void draw(Graphics2D g2D) ;
* int getZ() ;
* boolean isVisible() ;
* boolean isDestroyed() ;

Class RenderableHolder using singleton pattern

Field

* ArrayList<IRenderable> items ; to list all of things that can be drawn

Method

* getter
* Void add (IRenderable item) ; to add item to arraylist and sort by getZ to know which one is come first.
* Void clear() ; to make new arraylist for new game.

Part Main

Class Main:main class of this game

Field

* static JComponent *currentScreen*;current screen to display
* static JFrame *frame*;main frame of this game
* static GameScreen *gameScreen*;
* static MainLogic *logic*;
* static IntroScreen *introScreen*;
* static final int *INTRO* = 1;
* static final int *GAME* = 2;

Method

* main(String[] args);run the game
  + create frame
  + initialize all fields
  + add keyListener
  + game loop
    - update logic
    - paint component in game screen
    - post update of InputUtility
* changeScreen(int screen);change current screen

------------------------------------THANK YOU ---------------------------------------