**Class Main.ScreenManager**

This class use to manage whole screen.

Field:

+int INTROSCREEN; declare intro screen key

+int SELECTSCREEN; declare select screen key

+int GAMESCREEN; declare game screen key

+int ATTACKSCREEN; declare attack screen key

+int WINNINGSCREEN; declare winning screen key

+int PAUSESCREEN; declare pause screen key

-int FADING; declare fading in method key

-int FADEOUT; declare pause fading out method key

-boolean chagingScreen; key to tell that the screen is changing

­­- AudioClip bgm : declare background music

- NorthScreen northScreen : declare north screen

­- SouthScreen southScreen : declare south screen

- NorthScreenLogic northScreenLogic : declare north screen logic

- SouthScreenLogic southScreenLogic : declare south screen logic

­- IntroScreen introScreen : declare intro screen

- PauseScreen pauseScreen : declare pause screen

- GameOverScreen gameOverScreen : declare gameover screen

- JFrame MainFrame : declare Main frame

- JPanel panelInsideFrame : declare panel that will contain in main frame

currentScreen : ArrayList<JComponent>

currentLogic : ArrayList<Logic>

Constructor:

-public void ScreenManager(); setup MainFrame and panel inside MainFrame then put this panel into MainFrame ,loop the game process such as screen and logic method.

Method:

-generated getter settle of fields.

-public void resetScreen(); create new every screen object

-public void changeScreen(int); method that manage how to change between screen.

-public void fadeScreen(JPanel, boolean) method that mange how to fade between screen.

-public void addListener(JPanel) add mouse and keyboard listener to an argument panel.

**Class render.AnimationManager**

This class used to collect Animation and image data that used to play or show in this project.

Field:

+s int DONOTTHING : primarykey that used to told do not thing in every special method.

+s int FlipToUsual : primarykey that used to told flip method to flip animation to usual side.

+s int FlipToUnUsual : primarykey that used to told flip method to flip animation to the opposite side from usual.

+s int FLIP : primarykey that used to told flip method to flip animation to the other side from now.

+s int RotateRight : primarykey that used to told flip method to flip animation to rotate right.

+s int RotateLeft : primarykey that used to told flip method to flip animation to rotate left.

+s int BufferOPTIMIZED : primarykey that used to told render.ImageReader.get(int) method to load image with special method for “optimized gif”.

-s Boolean isPlay : contain this animation play status that play or stop

-s Boolean isLoop : contain data for loop if true means looping

- boolean isFinish : contain data for told that is this animation is finish playing or not.

- int frame : told index of current frame animation .

­- ImageData[]img : contain image data by ImageData object.

- int width : told this animation width.

- int height : told this animation height.

- int delayTime : told this animation delay time between each frame in millisecond

- int delayCounter : counter that’s used to count when call update method .

- int charWidth : contain this character width.

- int charHeight : contain this character height.

- int setX : contain this character’s foot in x-coordinate.

- int setY : contain this character’s foot in y-coordinate

- int flipInfo : data to collect this animation flip status (false means animation was not flipped and vice versa).

Constructor:

public WireCut();initialize the value of each field. generate gap and wire with dual primary key.

Method:

-public void flip(int): flip animation with primary key.

-public void flip(): flip animation to the another way.

-public void flipToUnUsual(): flip animation to unusual side.

-public void flipToUsual(): flip animation to normal side.

-public void flipImage(): flip animation to the other side.

-public int getCharWidth(): get character width.

-public int getCharHeight(): get character height.

-public int getCharWidthByHeight(int): get character width by calculate form height’s character argument to scale by original ratio.

-public int getCharHeightByWidth(int): get character height by calculate form width’s character argument to scale by original ratio.

-public int getSetX(): get character foot’s x-coordinate location .

-public int getSetY(): get character foot’s y-coordinate location .

-public int getFrame(): get this number of all frames.

-public int getWidth(): get this animation’s image width.

-public int getHeight(): get this animation ’s image height.

-public int getWidthByHeight(int): get image width by calculate form height’s image argument to scale by original ratio.

-public int getHeightByWidth(int): get image height by calculate form width’s image argument to scale by original ratio.

-public void play(): told this animation to play.

-public void loop(): told this animation to loop.

-public void stop(): told this animation to stop playing.

-public boolean isFinish(): get boolean that told is this animation has been finish play.

-public void setFinish(boolean): set this animation to finish play or not.

-public void update(): call this method to update this animation frame.

-public ImageData[] getAllImage(): get All image data in ImageData array object.

-public ImagteData getCurrentImageData(): get current image data in ImageData object.

-public ImagteData getCurrentImageData(int): get image data by index argument in ImageData object.

-public BufferdImage getCurrentBufferedImage(): get current image data in BufferdImage object.

-public BufferdImage getCurrentBufferedImage(int): get image data by index argument in BufferdImage object.

-public Object clone(): throw CloneNotSupportedException to told that this object cannot be cloned.

**Class render.RenderAnimationHelper**

This class used to help developer to draw Animation easier by draw animation like **RenderHelper class** but this class draw with ratio of character width and height.

Field:

#int x, y; the position that you want the player be.

#int charWidth, charHeight; describe player's size.

Constructor:

* None.

Method:

-public static void draw (Graphics2D g2d,AnimationManager animation,int x,int y,int userCharWidth,int userCharHeight): used to help developer to draw animation on Graphics2D which coordinate x and y is position that point to character’s foot position and it can scale the animation by userCharWidth or userCharHeight argument which you can leave one width or height to draw with width or height relate to its original character ratio.

**Class render. RenderHelper**

This class used to help developer to draw Animation easier by draw animation with special key word to told this class to draw image in which position relate to x and y argument.

Field:

+staic int LEFT, CENTER, RIGHT , TOP, MIDDLE , BOTTOM, REPEAT : special key word to told which position to draw image that relate with x, y coordinate.

Constructor:

* None.

Method:

* public static void draw(Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position) : throw all arguments to next draw(…) method which send event as null.

* public static void draw(Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position,RenderHelperMouseEvent event) : draw BufferedImge on Graphics2D which draw by used special key word to told which position to draw image that relate to x, y coordinate argument and check that is event has object or not if yes then call check event method.
* private static void checkEvent(RenderHelperMouseEvent event, int x, int y, int width, int height) : check event and handle event by event in RenderHelperMouseEvent object.
* private static boolean isMouseEntered(int x,int y,int width,int height) : check that is user’s mouse position has been in image area.
* public static void addAntiAlising(Graphics2D g2d) : add anti-alising in Graphics.
* public static void removeAntiAliasing(Graphics2D g2d) : remove anti-alising in Graphics.

**Class render.RenderHelperMouseEvent**

This is an abstract class that use to make mouse event in RenderHelper class.

Field:

* none.

Constructor:

* none.

Method:

- public abstract void mouseClicked(); call this event when mouse has been clicked on image area.

- public abstract void mousePressed(); call this event when mouse has been pressed on image area.

- public abstract void mouseReleased();call this event when mouse has been released on image area.

- public abstract void mouseEntered();call this event when mouse has been moved in image area.

- public abstract void mouseExited();call this event when mouse has been moved out image area.

**Class resource.** **FethResourceException**

This class extends from Exception use to throw Exception type with message by error type when get Animation from ArrayList by wrong key.

Field:

+ static final int ANIMATION ; primary key for animation error type.

+ static final int AUDIO ; primary key for audio error type.

- int errorType; used to collect error type.

- String url; used to collect url of error file.

Constructor:

* Public FethResourceException(int errorType,String url) : Get and set the value of each field.

Method:

- public String getMessage(); get error message.

**Class resource.** **GetResourceException**

This class extends from Exception use to throw Exception type with message by error type when class loader get resource and that file was not found.

Field:

+ static final int ANIMATION ; primary key for animation error type.

+ static final int AUDIO ; primary key for audio error type.

- int errorType; used to collect error type.

- String url; used to collect url of error file.

Constructor:

* Public FethResourceException(int errorType,String url) : Get and set the value of each field.

Method:

- public String getMessage(); get error message.

**Class resource.Resource**

This class declare and mange every resources that used in this project.

Field:

- static HashMap<String,AnimationManager> rs = new HashMap<>(); contain Animation resource.

- static HashMap<String,AudioClip> audio = new HashMap<>(); contain audio resource.

+ static final Font standardFont : contain font style.

+ static BufferedImage[] playButton :

+ BufferedImage[] rankButton : BufferedImage[]

+ BufferedImage[] continueButton : BufferedImage[]

+ BufferedImage[] exitButton : BufferedImage[]

+ BufferedImage[] restartButton : BufferedImage[]

+static Cursor CURSOR\_DEFAULT : Cursor

+static Cursor CURSOR\_HAND : Cursor

Constructor:

initialize the resource file location and put them to ArrayList.

Method:

- public AnimationManager read(String url,int setX,int setY,int setCharWidth,int setCharHeight,int mode) throws FethResourceException : get AnimationManager object by its url location with set its x,y foot location , character width and height and special primary keyword which told ImageReader class to get gif file with OPTIMIZED mode or not.

- public AnimationManager read(String, int, int, int, int): return AnimationManager like the others method in previous mention.

- public AnimationManager read(String, int) : return AnimationManager like the others method in previous mention.

- public AnimationManager read(String) : return AnimationManager like the others method in previous mention.

- public AudioClip audioRead(String) : return AudioClip by string keyword.

- public static AnimationManager get(String) : return AnimationManager by string keyword.

- public static BufferedImage getImage(String) : return BufferedImage by string keyword.

- public static BufferedImage getImage(String, int) : return BufferedImage by string keyword and its index of frame.

- public static AudioClip getAudio(String) : return AudioClip by string keyword.

**Class ui.GameOverScreen**

This class extends from JComponent use to display Game over screen.

Field:

- AnimationManager BG; Background animation.

- BufferedImage img; Background current BufferedImage.

- int width,height; Screen width and height.

Constructor:

initialize the value of each field and setup components.

Method:

- protected void paintComponent(Graphics g): draw components.

- private void drawStartBT(Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.

**Class ui.IntroScreen**

This class extends from JComponent use to display intro screen.

Field:

- AnimationManager BG; Background animation.

- BufferedImage img; Background current BufferedImage.

- int width,height; Screen width and height.

Constructor:

initialize the value of each field and setup components.

Method:

- protected void paintComponent(Graphics g): draw components.

- private void drawStartBT(Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.

- private void drawRankBT(Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.

**Class ui.** **NorthScreen**

This class extends from JComponent use to display north screen of game.

Field:

- AnimationManager bgAnimation; Background animation.

- BufferedImage img; Background current BufferedImage.

- int width,height, statusHeight; Screen width , height and status height.

- ArrayList<IRenderable> entity; contain north renderable list.

Constructor:

initialize the value of each field and setup components.

Method:

- protected void paintComponent(Graphics g): draw components.

**Class ui.** **PauseScreen**

This class extends from JComponent use to display pause screen.

Field:

- AnimationManager BG; Background animation.

- BufferedImage img; Background current BufferedImage.

- int width,height; Screen width and height.

Constructor:

initialize the value of each field and setup components.

Method:

- protected void paintComponent(Graphics g): draw components.

- private void drawContinueBT (Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.

- private void drawReStartBT (Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.

- private void drawExitBT (Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.

**Class ui.** **SouthScreen**

This class extends from JComponent use to display south screen of game.

Field:

- AnimationManager bgAnimation; Background animation.

- BufferedImage img; Background current BufferedImage.

- int width,height, statusHeight; Screen width , height and status height.

- ArrayList<IRenderable> entity; contain south renderable list.

Constructor:

initialize the value of each field and setup components.

Method:

- protected void paintComponent(Graphics g): draw components.

**Class ui.** **WinningScreen**

This class extends from JComponent use to display winning screen.

Field:

- AnimationManager BG; Background animation.

- BufferedImage img; Background current BufferedImage.

- int width,height; Screen width and height.

Constructor:

initialize the value of each field and setup components.

Method:

- protected void paintComponent(Graphics g): draw components.

- private void drawReStartBT (Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.

**Class ui.IntroScreen**

This class extends from JComponent use to display intro screen.

Field:

- AnimationManager BG; Background animation.

- BufferedImage img; Background current BufferedImage.

- int width,height; Screen width and height.

Constructor:

initialize the value of each field and setup components.

Method:

- protected void paintComponent(Graphics g): draw components.

- private void drawStartBT(Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.

- private void drawRankBT(Graphics2D g, BufferedImage img, int x, int y, int width, int height, int position): method that help developer to draw button on Graphics2D and setup its mouseEvent.