

welcome

Load Runner Game

**2st Year Students project**

*Project Team*

*1-mohamed abdl ghfar(leader)*

*2-mohamed saeed hosny*

*3-mohamed abdl Rahman araby*

*4-mohamed abdl sammed helmy*

*5-mohamed sabry ahmed*

*6-mohamed Reda Mansour*

*7-mohamed abdl baset ameen*

**Abstract**

**In this project , we are create a game which called load runner .**

**This game considered entertainment for the user .**

**Introduction**

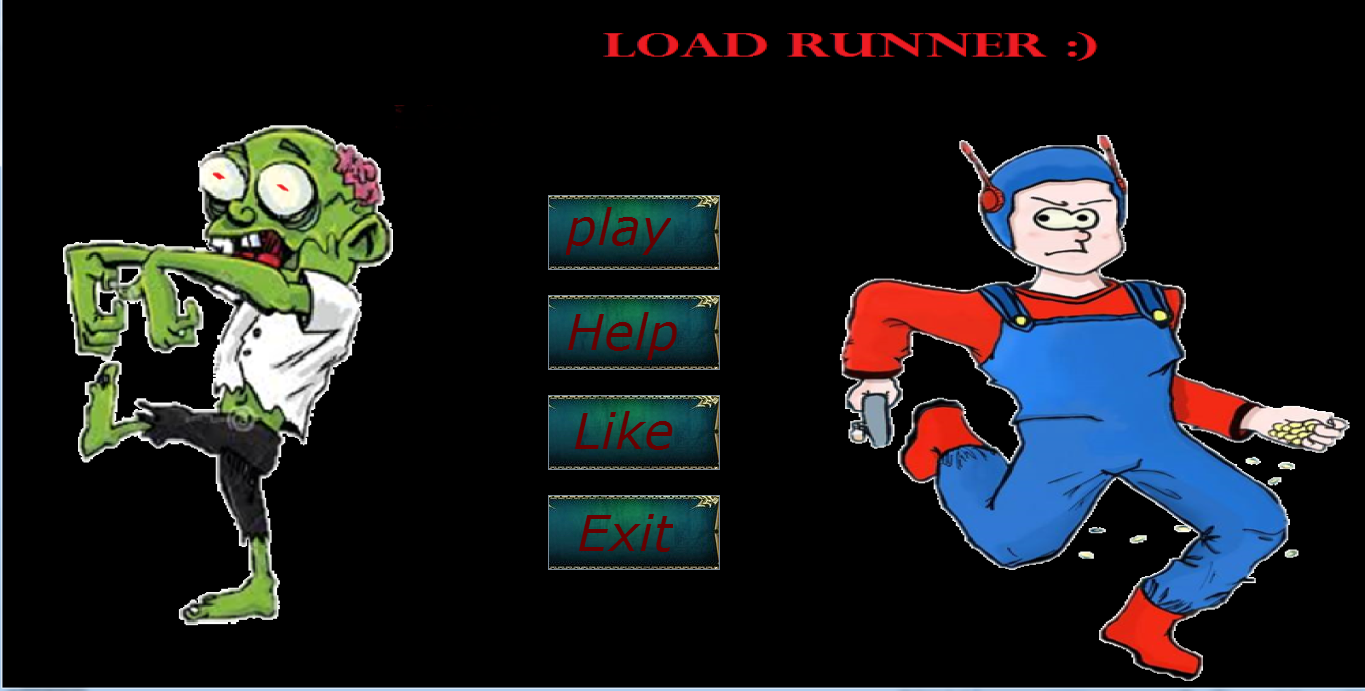
**This game is about the thief running on the bricks trying to steal gold .**

**there is a group of zombies trying to prevent him ...**

**This player can break the rocks to get rid of the zomby.**

**The game consists of 5 levels**

**Program Interface**

****

**This section explains how the user can play the game.**

**Is the first frame there four button he can**

**1-choose start game button to start game**

**2-choose help to learn about game moves**

**3-choose like to communicate with us in facebook**

**4-choose Exit to leave game**

**program Structure**

**This section explains how the game consist of**

**The main parts of game is**

1. **Graphics**
2. **Player**
3. **Zombie**
4. **Coin**

**And other classes.**

**1-Graphics**

**C:\Users\mohamed reda\Pictures\3.PNG   **

**Is about of images that appear at the back of the game such brick, coin, player and zomby.**

**Graphics consists of several class such**

* + **Imageloader class**
  + **SpriteSheet class**
  + **Assets class**
  + **Animation class**

**This is some codes from this classes**

* **Image loader class**

**That use to load image.**

**Public class Image Loader {**

**public static BufferedImage loadImage(String path){**

**try {**

**return ImageIO.read(ImageLoader.class.getResource(path));**

**} catch (IOException e) {**

**e.printStackTrace();**

**System.exit(1);**

**}**

**return null;**

**}**

* **Sprite Sheet class**

**This class to cut image.**

**Public class Sprite Sheet {**

**private BufferedImage sheet;**

**public SpriteSheet (BufferedImage sheet){**

**this.sheet=sheet;**

**}**

**public BufferedImage crop(int x,int y,int width,int height){**

**return sheet.getSubimage(x, y, width, height);**

* **Assets class**

**This class help to assign the cut image to variable.**

**private static final int width = 64, height = 64;**

**public static BufferedImage space\_run,space, tree, dirt,rock,stair;**

**public static BufferedImage[] player\_down, player\_up, player\_right, player\_left,player\_stand\_up;**

**public static BufferedImage[] zompy\_down, zompy\_up, zompy\_right, zompy\_left,zompy\_stand\_up;**

**public static BufferedImage[] zombie\_down, zombie\_up, zombie\_right, zombie\_left;**

**public static BufferedImage[] coin;**

**public static BufferedImage[] btn\_start;**

**public static int w;**

**public static BufferedImage stone;**

**public static void init() {**

**double x=Math.random()\*3;**

**w=(int)x;**

**SpriteSheet sheet = new SpriteSheet(ImageLoader.loadImage("/texture/space.jpg"));**

**space = sheet.crop(0, 0, width, height);**

**sheet = new SpriteSheet(ImageLoader.loadImage("/texture/stone-"+(w)+".png"));**

**stone = sheet.crop(0, 0, width, height);**

**sheet =new SpriteSheet(ImageLoader.loadImage("/texture/coins/coin1.png"));**

**coin[0] = sheet.crop(0, 0, width, height);**

**////////////////////////player image animation**

**player\_stand\_up = new BufferedImage[13];**

**player\_down = new BufferedImage[8];**

**player\_up = new BufferedImage[8];**

**player\_right = new BufferedImage[8];**

**player\_left = new BufferedImage[8];**

**sheet = new SpriteSheet(ImageLoader.loadImage("/texture/player\_down/p1.png"));**

**player\_down[0] = sheet.crop(0, 0, width, height);**

**/////////////////////////////zomby image animation**

**zompy\_stand\_up = new BufferedImage[12];**

**zompy\_down = new BufferedImage[8];**

**zompy\_up = new BufferedImage[8];**

**zompy\_right = new BufferedImage[8];**

**zompy\_left = new BufferedImage[8];**

**////////////////////// PLAYER STAND UP**

**sheet = new SpriteSheet(ImageLoader.loadImage("/texture/zompy\_stand\_up/p1.png"));**

**zompy\_stand\_up[0] = sheet.crop(0, 0, width, height);**

* **Animation class**

**That help to make player ,zombie and coin change image by time.**

**public class Animation {**

**public void tick() {**

**timer += System.currentTimeMillis() - lastTime;**

**lastTime = System.currentTimeMillis();**

**if (timer > speed) {**

**index++;**

**timer = 0;**

**if (index >= frames.length) {**

**index = 0;**

**}} }**

**public BufferedImage getCurrentFrame() {**

**return frames[index];**

**}**

**//setter**

**} }**

**2-player class**

**This class to control with player**

**This is this code**

**public void tick() {**

**anim\_stand\_up.tick(); animDown.tick(); animUp.tick();**

**animRight.tick(); animLeft.tick();**

**sensorXdown= (int) (handler.getWorld().getEntitiyManager().getPlayer().getX()+Tile.TILEWIDTH/2)/Tile.TILEWIDTH;**

**sensorYdown = (int) (handler.getWorld().getEntitiyManager().getPlayer().getY() + Tile.TILEHEIGHT)/Tile.TILEHEIGHT;**

**sensorXup= (int) (handler.getWorld().getEntitiyManager().getPlayer().getX()+Tile.TILEWIDTH/2)/Tile.TILEWIDTH;**

**sensorYup = (int) (handler.getWorld().getEntitiyManager().getPlayer().getY())/Tile.TILEHEIGHT;**

**sensorXleft= (int) (handler.getWorld().getEntitiyManager().getPlayer().getX())/Tile.TILEWIDTH;**

**sensorYleft = (int) (handler.getWorld().getEntitiyManager().getPlayer().getY() + Tile.TILEHEIGHT/2)/Tile.TILEHEIGHT;**

**sensorXright= (int) (handler.getWorld().getEntitiyManager().getPlayer().getX()+Tile.TILEWIDTH)/Tile.TILEWIDTH;**

**sensorYright = (int) (handler.getWorld().getEntitiyManager().getPlayer().getY() + Tile.TILEHEIGHT/2)/Tile.TILEHEIGHT;**

**NO\_WAY\_TO\_LIFE();**

**if (isSpace(sensorXdown, sensorYdown)) {**

**y += speed;**

**}**

**if(handler.getWorld().getEntitiyManager().getZomby().getX()==this.x&&handler.getWorld().getEntitiyManager().getZomby().getY()==this.y){**

**this.die();**

**}**

**//movement**

**getInput();**

**move();**

**handler.getGameCamera().centerOnEntity(this); // to make the camera zoom to the player and his place only**

**//ATTACK**

**checkAttacks();**

**}**

**3-Zomby class**

**This class to control with zombie**

**public void tick() {**

**Altx = this.x; Alty = this.y; anim\_stand\_up.tick();**

**animDown.tick(); animUp.tick(); animRight.tick(); animLeft.tick();**

**sensorXdownZ = (int) (this.x + Tile.TILEWIDTH / 2) / Tile.TILEWIDTH;**

**sensorYdownZ = (int) (this.y + Tile.TILEHEIGHT) / Tile.TILEHEIGHT;**

**sensorXupZ = (int) (this.x + Tile.TILEWIDTH / 2) / Tile.TILEWIDTH;**

**sensorYupZ = (int) (this.y) / Tile.TILEHEIGHT;**

**sensorXleftZ = (int) (this.x) / Tile.TILEWIDTH;**

**sensorYleftZ = (int) (this.y + Tile.TILEHEIGHT / 2) / Tile.TILEHEIGHT;**

**sensorXrightZ = (int) (this.x + Tile.TILEWIDTH) / Tile.TILEWIDTH;**

**sensorYrightZ = (int) (this.y + Tile.TILEHEIGHT / 2) / Tile.TILEHEIGHT;**

**if (handler.getWorld().getEntitiyManager().getPlayer().getX() == this.x && handler.getWorld().getEntitiyManager().getPlayer().getY() == this.y) {**

**handler.getWorld().getEntitiyManager().getPlayer().die(); }**

**if (isSpace(sensorXdownZ, sensorYdownZ) && this.id != 0) {**

**y += speedZomby;**

**die();**

**}**

**if(sensorXleftZ==handler.getWorld().getEntitiyManager().getPlayer().sensorXleft&&sensorYrightZ==handler.getWorld().getEntitiyManager().getPlayer().sensorYright)**

**{**

**handler.getWorld().getEntitiyManager().getPlayer().die();**

**}**

**System.err.println(sign);**

**if (this.id != 0) {**

**direction(sign);**

**}**

**handler.getGameCamera().centerOnEntity(this); // to make the camera zoom to the player and his place only**

**if (this.timing < 300 && this.id == 0) {**

**timing++;**

**} else if (this.id == 0 && this.timing >= 300) {**

**back\_place();**

**}**

**}**

**4-coin class**

**public void tick() {**

**coin.tick();**

**if(this.x/64==handler.getWorld().getEntitiyManager().getPlayer().getX()/64&&**

**( this.y/64)==(int)((handler.getWorld().getEntitiyManager().getPlayer().getY())/64)){**

**handler.getWorld().getEntitiyManager().getEntities().remove(this);**

**handler.getWorld().setScore(handler.getWorld().getScore() +100);**

**}**

**if((int)this.x/64-(int)handler.getWorld().getEntitiyManager().getPlayer().getX()/64>-2&&**

**(int)this.x/64-(int)handler.getWorld().getEntitiyManager().getPlayer().getX()/64<2&&**

**(int)( (int)this.y/64)==(int)(((int)handler.getWorld().getEntitiyManager().getPlayer().getY())/64)){**

**handler.getWorld().getEntitiyManager().getEntities().remove(this);**

**handler.getWorld().setScore(handler.getWorld().getScore() +100);**

**}**

**}**

**There is some class that help to control this classes with each other such……………………**

* **Game**

**Is the main class in the game that connect the classes**

**And have start and stop function.**

**It have object from game class to make tick ,render to world.**

**public synchronized void start() {**

**running = true;**

**thread = new Thread(this);**

**thread.start();**

**//IT IS ACTUALLY CALL FOE RUN METHOD**

**}**

**public synchronized void stop() {**

**if (!running) {**

**return;**

**}**

**running = false;**

**try {**

**thread.join();**

**} catch (InterruptedException ex) {**

**Logger.getLogger(Game.class.getName()).log(Level.SEVERE, null, ex);**

**}}**

* **Display**

**this class to the frame that have canvas that contain the game.**

* **GameCamera**

**This class to make camera follow the player when it move in the frame from up to down.**

* **Handler**

**That help to control to game from some function.**

* **Frames**

**Is the first frame that run in the game.**

* **keyManger**

**that help to move player and zombies.**

**The user can play with**

**(w) to move up**

**(s) to move down**

**(a) to move left**

**(d) to move right**

* **Tiles**

**In this class we hold every image with number to**

**Make draw easly.**

**(0) mean space**

**(2) mean brick**

**(4) mean stair**

* **world**

**that help to draw image graphics in file by number that**

* **utils**

**that read from file and pass it to world to draw.**

* **State**

**This is abstract class that have two mothed tick,render**

**That responsible to divide the game by selling parts to work with each other and how much we can control the parts and this part is gamestate, menuestate.**

* **GameState**

**This class extend State and make in it implement to tick , render motheds.**

**This responsible to make tick , render to world for the game .**

* **menuState**

**is a class that responsible for frame in game and how to work with it.**

* **Entity**

**It make rectangle around the player and zombie.**

**It have function to show if zomby hit player or not called die.**

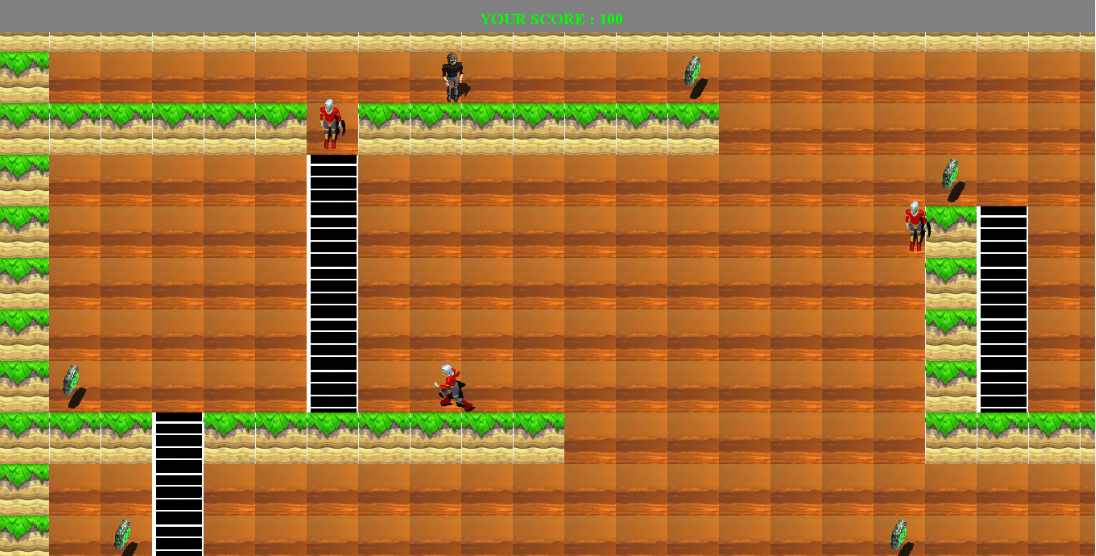
* **EntityManger**

**It have functions that add and delete player or zomby.**

**Conclusion**

* **In this game 5 levels .**

**Such**



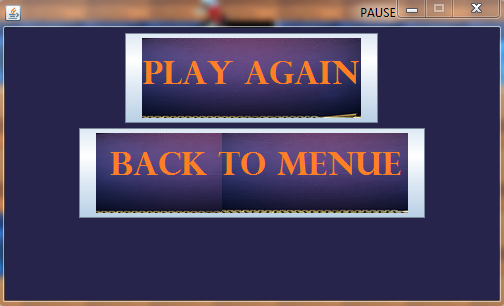


**It can able to enter the second level and etc when the player play the first level and end it Successfully.**

* **When the player die**

**There are frame appear yhat have two button**

* **Play again**
* **Back to menu**

****