Doodle Jump

Technical Documentation

Ist computers and systems
Zagazig University
Programming 3

The program has two packages:

1- Game Logic
2- Pages

- Game Logic

It contains all the code for the Game pane to function; which includes classes for the player, obstacles, monsters and powerups.
As well as the Keyboard Listener which inputs the key presses, And the Arduino Listener which is used to input from a JoyStick or a Gyroscope.

- Pages

It contains classes for every page the user sees, as well as Images, Audio and FileIO handler classes.

Game Logic

"GamePage" Class:

Start Function:

Initializes the game pages fully by creating arrays of objects for every part of the game, which are Obstacles, Monsters And Powerups.

It also has the game loop which is an instance of the class AnimationTimer from the JavaFX library, it contains all the logic that needs to be looped 60 times a second.

FPS Counter Function:

It counts how many frames per second the game is outputting.

Lose Check Function:

This function is in a loop which checks for certain conditions to stop the game and show the "gameover" page when the player loses.

Lose Function:

This is called from the "LoseCheck" function to stop everything and show the "gameover" Page.

Create Function:

This is called to create the game page pane with a certain resolution specified from the settings.

"KeyBoardListener" Class:

Start Function:

Sets up the event handlers to check for key presses and releases as well as mouse clicks, if an event is detected a condition is put to true.

Loop Counter Function:

This function gets looped through, it checks for the conditions the start function sets, and call the functions for movement of the player.

"ArduinoListener" Class:

Start Function:

Checks if the selected open port for serial communication is open, and it also starts a new thread for reading the serial ports.

MyRunnable class:

This class contains the code the thread runs, it listens to the serial ports and processes it for the movement functions.

Serial class:

This class contains the functions for communicating with the serial ports of the Arduino. Which include OpenPort, GetAvaliablePorts, WriteToPort and ListenToPort.

"Player" Class:

MoveY Function:

This function is called to move the Player image on the Y-Axis while checking for collisions of every other object. We send the y velocity, Player object, obstacle array and powerup array; to check collisions for all of them before moving the player.

MoveX Function:

This function is called to move the Player image on the X-Axis while checking for collisions of every other object. We send the x velocity, Player object, obstacle array and powerup array; to check collisions for all of them before moving the player.

We separated the two functions to X and Y because we want the player to be able to move on one axis if the other got collided with.

ScreenSroll Function:

This method is called in a loop when the player reaches a certain threshold on the Y-axis of the screen. So when that condition is true, every object gets moved down until the player is under the threshold again.

GravitCycle Function:

This method moves the player down with the acceleration of gravity. It updates the y velocity constantly and sends that velocity to the moveY function.

GravitCycle Function:

This function rotates the nozzle of the doodle when a projectile object is created. It gets rotated by the angle computed in the projectile object.

"Obstacle" Class:

Initialize Function:

This method when called and get passed an array of obstacles, it initializes every object of that array.

Swing Function:

This method moves obstacles that are marked for moving from side to side of the screen. It is called in a loop.

TeleportUp Function:

This method is called when the obstacle is under the screen so it can be used again by teleporting it up the screen. It also teleports the monster or the power up associated with it. It is also called in a loop.

XRandom Function:

This method returns a random X value for the obstacle with certain constraints, it is called when the object is created or teleported Up.

"Monster" Class:

Initialize Function:

This method when called and get passed an array of monsters, it initializes every object in that array.

Activate Function:

This method activates the monsters. Called by the initialize function.

RadnomActivation Function:

This method is called for a random chance of activating the monster.

CheckDuplicates Function:

This method checks if there is already an object linked to the obstacle in question, if there is; then it finds another obstacle index.

RadnomActivation Function:

This method is called for a random chance of activating the monster.

TeleportUp Function:

This method is called when the Monster is under the screen so it can be used again by teleporting it up the screen.

BoundTo Function:

This method chooses the type of monster randomly and calls the setType method.

SetType Function:

This method sets up the properties for the selected type sent.

Loop Function:

This method is used to animate the monsters' pictures, by switching the place of a Rectangle2D object in the setViewPort Function.

"PowerUp" Class:

Initialize Function:

This method when called and get passed an array of PowerUps, it initializes every object of that array.

InitializeAnimations Function:

Call this static method to initialize a PowerUp array containing PowerUpAnimation objects.

Activate Function:

This method activates the powerups. Called by the initialize function.

CheckDuplicates Function:

This method checks if there is already an object linked to the obstacle in question, if there is, then it finds another obstacle index.

RadnomActivation Function:

This method is called for a random chance of activating the PowerUp.

TeleportUp Function:

This method is called when the PowerUp is under the screen so it can be used again by teleporting it up the screen.

BoundTo Function:

This method choses the type of monster randomly and calls the setType method.

SetType Function:

This method sets up the properties for the selected type sent.

Excute Function:

This method is called when the player collides with a PowerUp, So it starts its execution depending on its type. Every execution is different.

"Projectile" Class:

Loop Function:

This Method moves every projectile its assigned x and y velocities, it is called in a loop and goes through an array list of projectile objects.

Create Function:

This Method Creates a projectile when the user clicks on the screen. It checks for a lot of conditions of the position of the click. If it passes it creates the object and does the necessary computations and adds it to the arraylist.

"PausePage" Class:

This Class creates a pane that gets added to the game pane when the pause button is clicked. It also stops the animationtimer that loops the game.

"Animation" Class:

This Class gets creates an object that is used to animate the picture tiles of the sent object using Rectangle2D with setViewPort.

Pages

"Audio" Class:

This class is used to create AudioClip objects. We made it for the static property "Mute" so when we call the start function it checks if the Mute property is false or true.

"FileIO" Class:

Read Function:

This function gets called and get passed a file name to read form it, Which returns a string of the contents read.

Write Function:

This function gets called and get passed a file name and a string to Write in the file specified.

SettingsRead Function:

This function is specialized to read from settings.txt file the certain line which is being passed when called. And then it returns that line as a string.

SettingWrite Function:

This function is specialized to Write onto the settings.txt file with a certain line index which is being passed when called. And then it writes the string onto the line specified onto the file.

"Images" Class:

This Class is called to initialize all the game images on the start up of the program to prevent lag or delay from happening between switching scenes.

For the Following Classes:

A scene is created when the create function is called and get passed to the primary stage.

"MainPage" Class:

This page is the first page called when the game is launched. It has all the primary buttons.

"SelectPage" Class:

This page has the NewGame and the continue buttons. Newgame calls the login page and the continue calls the difficulty page.

"DifficultyPage" Class:

This page has three buttons for the three difficulties. It changes the number of monsters and powerup depending on which one is clicked. It then sends the user to the game page.

"CreditsPage" Class:

This page shows the credits for the developers who made the game.

"SettingPage" Class:

This page has all the settings for the game, which includes cheats, input mode, resolution, themes and mute sounds. Everything gets updated as soon as the buttons are clicked.

"GameOverPage" Class:

This page gets called when the user loses. It plays a sound and then gives the player two options. A "Play Again" or go the "Main" menu options.

"ScoresPage" Class:

This page has all the users that are logged in and their scores. It is also being sorted by highest scores.

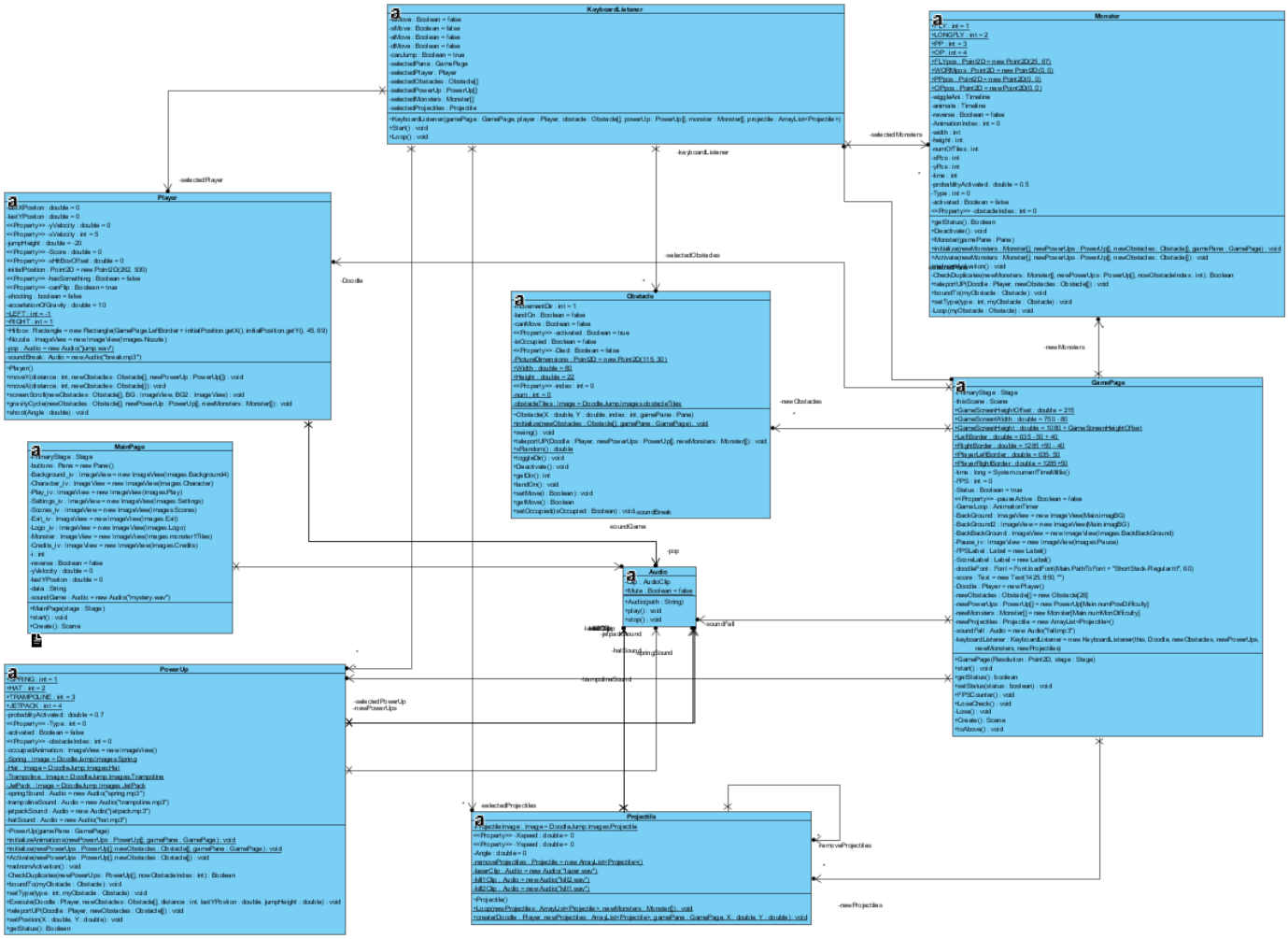
"LoginPage" Class:

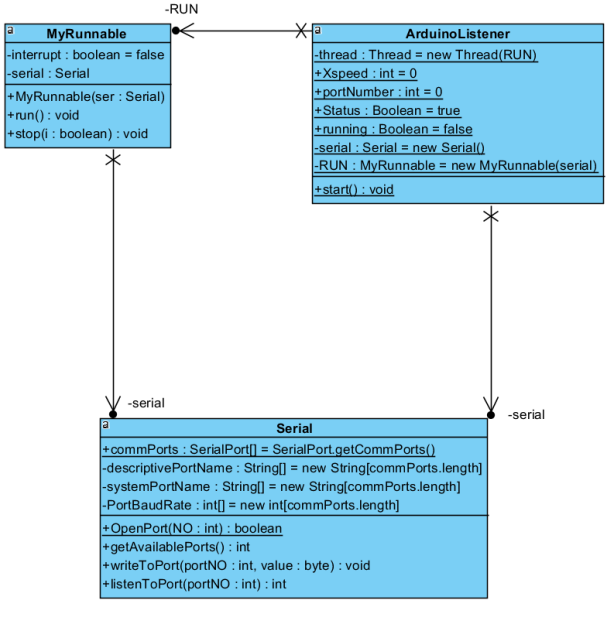
This page lets the user enter their name and their age to save it and use it for the scores page.

The Game was developed by:

1- Karim Salah-Eldin
2- Mahmoud Galal
3- Mohammed Gehad
4- Karim Wael
5- Mohammed Ghanem
6- Mohammed Hossien

UML


KeyboardListener
Monster
Player
Obstacle
MainPage
GamePage
Audio
PowerUp
Projectile


-RUN
MyRunnable
-interrupt : boolean = false
-serial : Serial
+MyRunnable(ser : Serial)
+run() : void
+stop(i : boolean) : void
ArduinoListener
-thread : Thread = new Thread(RUN)
+Xspeed : int = 0
+portNumber : int = 0
+Status : Boolean = true
+running : Boolean = false
-serial : Serial = new Serial()
-RUN : MyRunnable = new MyRunnable(serial)
+start() : void
-serial
-serial
Serial
+commPorts : SerialPort[] = SerialPort.getCommPorts()
-descriptivePortName : String[] = new String[commPorts.length]
-systemPortName : String[] = new String[commPorts.length]
-PortBaudRate : int[] = new int[commPorts.length]
+OpenPort(NO : int) : boolean
+getAvailablePorts() : int
+writeToPort(portNO : int, value : byte) : void
+listenToPort(portNO : int) : int

SelectPage

-PrimaryStage : Stage
-Background_iv : ImageView = new ImageView(Images.Background8)
-NewGame_iv : ImageView = new ImageView(Images.NewGame)
-Continue_iv : ImageView = new ImageView(Images.Continue)
-X_iv : ImageView = new ImageView(Images.X)
~xx : double = 720
~yy : double = 600

+SelectPage(stage : Stage)
+start() : void
+Create() : Scene

SettingsPage

-PrimaryStage : Stage
-buttons : Pane = new Pane()
-Background3_iv : ImageView = new ImageView(Images.SettingsPage)
-X_iv : ImageView = new ImageView(Images.X)
-Voff_iv : ImageView = new ImageView(Images.Voff3)
-Von_iv : ImageView = new ImageView(Images.Von)
-Save_iv : ImageView = new ImageView(Images.Save)
-Saved_iv : ImageView = new ImageView(Images.Empty)
-Res_iv : ImageView = new ImageView()
-zom_iv : ImageView = new ImageView()
-nor_iv : ImageView = new ImageView()
-ice_iv : ImageView = new ImageView()
-inputMode : ImageView = new ImageView()
-fly_iv : ImageView = new ImageView(Images.Fly)
-nojump_iv : ImageView = new ImageView(Images.noJump)
-immune_iv : ImageView = new ImageView(Images.Immune)
~click : boolean = false
~click2 : boolean = false

+SettingsPage(stage : Stage)
+start() : void
+Create() : Scene

ScoresPage

-PrimaryStage : Stage
-Background5_iv : ImageView = new ImageView(Images.Background5)
-X_iv : ImageView = new ImageView(Images.X)
-topscore : Text = new Text(1050,475,FileIO.Read("TopScore.txt"))
-playerName : Text = new Text(650,475,FileIO.Read("PlayerName.txt"))

+ScoresPage(stage : Stage)
+start() : void
+Create() : Scene

Main

+ResolutionFullHD : Point2D = new Point2D(1920, 1080)
+ResolutionHD : Point2D = new Point2D(1280, 720)
+ResolutionCustom : Point2D = new Point2D(1600, 900)
+OffsetFullHD : Point2D = new Point2D(0, 0)
+OffsetHD : Point2D = new Point2D(-213, -119)
+OffsetCustom : Point2D = new Point2D(-133, -75)
+SelectedResolution : Point2D = DoodleJump.Main.ResolutionFullHD
+SelectedOffset : Point2D = DoodleJump.Main.OffsetFullHD
+Factor : double = 3
+numRes : int = 1
+numChar : int = 1
+imagChar : Image
+imagBG : Image
+numInput : int = 1
+fly : Boolean = false
+nojump : Boolean = false
+immune : Boolean = false
+numMonDifficulty : int = 2
+numPowDifficulty : int = 6
+PathToFont : String = "file:C:\\"
+PathToResources : String = "C:\\Users\\User\\Desktop\\New Folder\\DoodleJump\\"

+start(stage : Stage) : void
+main(args : String[]) : void

Animation

-reverse : Boolean = false
-i : int = 0
-animate : Timeline

~Animation(image : ImageView, width : double, height : double, time : int, numOfTiles : int, cycleCount : int, offset : int)

FileIO

-data : String

+Read(fileName : String) : String
+Write(path : String, fileName : String) : void

LoginPage

-PrimaryStage : Stage
-Background_iv : ImageView = new ImageView(Images.Background7)
-Next_iv : ImageView = new ImageView(Images.Next)
-X_iv : ImageView = new ImageView(Images.X)
~Name : TextField = new TextField()
~Age : TextField = new TextField()
~b : ChoiceDialog = new ChoiceDialog()

+LoginPage(stage : Stage)
+start() : void
+Create() : Scene

DifficultyPage

-PrimaryStage : Stage
-buttons : Pane = new Pane()
-Background2_iv : ImageView = new ImageView(Images.Background2)
-Easy_iv : ImageView = new ImageView(Images.Easy)
-Medium_iv : ImageView = new ImageView(Images.Medium)
-Hard_iv : ImageView = new ImageView(Images.Hard)
-X_iv : ImageView = new ImageView(Images.X)

+DifficultyPage(stage : Stage)
+start() : void
+Create() : Scene

GameOverPage

-PrimaryStage : Stage
-Background6_iv : ImageView = new ImageView(Images.Background6)
-Character_iv : ImageView = new ImageView(Images.Character)
-PlayAgain_iv : ImageView = new ImageView(Images.PlayAgain)
-Main_iv : ImageView = new ImageView(Images.Main)
-time : int = 1
-time2 : long = System.currentTimeMillis()
-data : String
-score : Text = new Text(1425, 850, FileIO.Read("Score.txt"))
-Monster : ImageView = new ImageView(Images.monster1Tiles)
-i : int
-reverse : Boolean = false
-yVelocity : double = 0
-lastYPostion : double = 0

+GameOverPage(stage : Stage)
+start() : void
+Create() : Scene

CreditsPage

-PrimaryStage : Stage
-Background_iv : ImageView = new ImageView(Images.CreditsBG)
-X_iv : ImageView = new ImageView(Images.X)

+CreditsPage(stage : Stage)
+start() : void
+Create() : Scene

PausePage

-X_iv : ImageView = new ImageView(Images.X)
-PlayAgain_iv : ImageView = new ImageView(Images.PlayAgain)
-Main_iv : ImageView = new ImageView(Images.Main)
-Resume_iv : ImageView = new ImageView(Images.Resume)
-PausePage : ImageView = new ImageView(Images.PausePage)
-doodleFont : Font = Font.badFont(Main.PathToFont + "DoodleJump.ttf", 50)

+Pause(stage : Stage, scene : Scene, Doodle : Player, GameLoop : AnimationTimer, gamePage : GamePage) : void

Images

+Background1 : Image = new Image("Background/Background1.png")
+Background2 : Image = new Image("Background/Background2.png")
+Background3 : Image = new Image("Background/Background3.png")
+Background4 : Image = new Image("Background/Background4.png")
+Background5 : Image = new Image("Background/Background5.png")
+Background6 : Image = new Image("Background/Background6.png")
+Background7 : Image = new Image("Background/Background7.png")
+Background8 : Image = new Image("Background/Background8.png")
+Background9 : Image = new Image("Background/Background9.png")
+Credits50 : Image = new Image("Background/Credits.png")
+PlayAgain : Image = new Image("Buttons/PlayAgain.png")
+PlayAgain2 : Image = new Image("Buttons/PlayAgain2.png")
+Pause : Image = new Image("Images/PauseButo.png")
+Pause2 : Image = new Image("Images/PauseButton-On.png")
+Resume : Image = new Image("Buttons/Resume.png")
+Resume2 : Image = new Image("Buttons/Resume2.png")
+Easy : Image = new Image("Buttons/Easy.png")
+Medium : Image = new Image("Buttons/Medium.png")
+Hard : Image = new Image("Buttons/Hard.png")
+Easy2 : Image = new Image("Buttons/Easy2.png")
+Medium2 : Image = new Image("Buttons/Medium2.png")
+Hard2 : Image = new Image("Buttons/Hard2.png")
+Main : Image = new Image("Buttons/Main.png")
+Main2 : Image = new Image("Buttons/Main2.png")
+Play : Image = new Image("Buttons/Playing.png")
+Play2 : Image = new Image("Buttons/Play-on.png")
+Settings : Image = new Image("Buttons/Settings.png")
+Settings2 : Image = new Image("Buttons/Settings-on.png")
+Scores : Image = new Image("Buttons/Scores.png")
+Scores2 : Image = new Image("Buttons/Scores-on.png")
+Exit : Image = new Image("Buttons/Exit.png")
+Exit2 : Image = new Image("Buttons/Exit-on.png")
+Credits : Image = new Image("Buttons/Credits.png")
+Credits2 : Image = new Image("Buttons/Credits2.png")
+NewGame : Image = new Image("Buttons/NewGame.png")
+NewGame2 : Image = new Image("Buttons/NewGame2.png")
+Continue : Image = new Image("Buttons/Continue.png")
+Continue2 : Image = new Image("Buttons/Continue2.png")
+Next : Image = new Image("Buttons/Next.png")
+Next2 : Image = new Image("Buttons/Next2.png")
+Save : Image = new Image("Buttons/Save.png")
+Save2 : Image = new Image("Buttons/Save2.png")
+X : Image = new Image("Buttons/x.png")
+X2 : Image = new Image("Buttons/x2.png")
+Saved : Image = new Image("Images/Saved.png")
+Empty : Image = new Image("Images/Empty.png")
+Voff : Image = new Image("Buttons/Voff.png")
+Voff2 : Image = new Image("Buttons/Voff2.png")
+Voff3 : Image = new Image("Buttons/Voff3.png")
+Von : Image = new Image("Buttons/Von.png")
+Von2 : Image = new Image("Buttons/Von2.png")
+Von3 : Image = new Image("Buttons/Von3.png")
+Logo : Image = new Image("Images/Login.png")
+Character : Image = new Image("Character/Character_Right_1.png")
+Bat1 : Image = new Image("Character/bat1.png")
+Bat2 : Image = new Image("Character/bat2.png")
+BackBackGround : Image = new Image("Images/Background.png")
+Hat : Image = new Image("Images/Hat.png")
+Trampoline : Image = new Image("Images/Trampoline.png")
+JetPack : Image = new Image("Images/Jetpack.png")
+Spring : Image = new Image("Images/Spring.png")
+Nozzle : Image = new Image("Images/Nozzle.png")
+playerTiles : Image = new Image("Images/DoodleTiles.png")
+obstacleTiles : Image = new Image("Images/ObstacleTiles.png")
+monsterITiles : Image = new Image("Images/MonsterITiles.png")
+monster2Tiles : Image = new Image("Images/Monster2Tiles.png")
+Projectile : Image = new Image("Images/Projectile.png")
+HatAnimation : Image = new Image("Images/Hat-Animation.png")
+JetpackAnimation : Image = new Image("Images/Jetpack-Animation.png")
+PausePage : Image = new Image("Images/PausePage.png")
+SettingsPage : Image = new Image("Background/Settings.png")
+ResHD : Image = new Image("Buttons/ResHD.png")
+ResFullHD : Image = new Image("Buttons/ResFullHD.png")
+ResCustom : Image = new Image("Buttons/ResCustom.png")
+noJump : Image = new Image("Buttons/nojump.png")
+fly : Image = new Image("Buttons/fly.png")
+Immune : Image = new Image("Buttons/Immune.png")
+Keyboard : Image = new Image("Buttons/Keyboard.png")
+Arduino : Image = new Image("Buttons/Arduino.png")
+nor1 : Image = new Image("Buttons/nor1.png")
+nor2 : Image = new Image("Buttons/nor2.png")
+zom1 : Image = new Image("Buttons/zom1.png")
+zom2 : Image = new Image("Buttons/zom2.png")
+ice1 : Image = new Image("Buttons/ice1.png")
+ice2 : Image = new Image("Buttons/ice2.png")
+doodleTiles : Image = new Image("Images/DoodleTiles.png")
+zombieTiles : Image = new Image("Images/ZombieTiles.png")
+snowTiles : Image = new Image("Images/SnowTiles.png")
+BGDoodle : Image = new Image("Images/bgDoodle.png")
+BGZombie : Image = new Image("Images/bgZombie.png")
+BGSnow : Image = new Image("Images/bgSnow.png")