AP Computer Science Final Project - README

Revolve

Authors: Vihaan Chinthakindi, Kevin Valencia

Revision: 5/23/22

Introduction:

Our program is a 2D platformer puzzle game where the player has to traverse several levels by rotating the room in order to get to the exit. We wrote this program as an homage to the many flash games we played in our childhood of which many do not exist anymore. The player must use their minds to figure out the correct combination of rotations either clockwise or counterclockwise in order for the player to avoid spikes and other traps and reach the exit door to the next level. The player has the ability to rotate the entire level in 90 degrees clockwise and counterclockwise , as well as the ability to move left, right, and jump. Hitting the spikes results in instant death and will restart the player back to the beginning of the level. Rotating incorrectly into a wall will also result in player death and restarting of the level. There will also be other methods of death such as saws and projectiles for added difficulty. A level is completed if the player passes through the door at the end of a level. The primary feature of the game is the revolving mechanic of the levels, making for an interesting spin on platforming games, with background music and sound effects as well. There are also multiple powerups that can be used to complete and make the level easier. A shop also exists where coins found in the levels can be spent to upgrade the player. The final goal for the player is to finish all of the levels. Each level can be played multiple times. We hope to bring a nostalgic feeling to the players and remind them of the exciting and fun flash games that we all enjoyed in childhood.

<u>**Instructions**</u>:

- **Arrow Keys Will Denote movements left, right, and up**

Where will you need to click?

Click Escape to go back to the Main Menu.
Click '1' to rotate the level clockwise 90 degrees.
Click '2' to rotate the level counterclockwise 90 degrees.

Will you have menus that need to be navigated? What will they look like?

- **There will be a menu at the beginning of the game, instructing users to start first level.**

Do actions need to be taken in a certain order?

- **Players start out on Level One, and can go to the next level if they skip or complete it. Any completed or skipped levels can be played back again.**

<u>**Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER)**</u>:

Must-have Features:

- Main menu that initiates the game. Will have a play button, a select level button, a shop button, and an exit game button. Buttons that will allow the player to quit level and return back to the main menu., with a shop and level selection screens.
- **DONE**
- Different ways of death ie spikes, and saws, that when interacted with will prompt the user to restart the level.
- **DONE**
- There will always be a door and end of the level, signifying the end of the level. The player will enter the end door to complete the current level and will be prompted with a choice to enter the next level or replay a prior level. back to the menu.
- **DONE**
- 7 levels of varying difficulty, with all other features described here, with a start point and a door to denote the endpoint
- **DONE**
- There will be music for each level or menu, as well as sound effects for actions in the game such as death, jumping, completing level.
- **DONE**
- Gravity that always points down relative to the user will be present in each level, as well as the ability to move left, right, and jump relative to the user's orientation.
- **DONE**
- Rotate method that rotates all the obstacles in the screen 90 degrees in both directions. Player does not move during these rotations relative to the screen.
- **DONE**

Want-to-have Features:

- More detailed backgrounds/wallpapers for each level.
- **DONE**
- Special Features/Powerups/Buffs in level
- **DONE**
- Currency System where coins are earned and a user can purchase permanent power up
- **DONE**
- Option to play as different characters with a menu detailing the advantages/disadvantages of each different character.
- Animated running/walking for the character so motion is smooth, animated death, and animations for the projectiles and saws.

Stretch Features:

- Ability for the player to design their own level with all the different combinations of walls/spikes, ability to change settings such as jump height, colors/characters of the level, and then the ability to play it.
- Different game modes such as timed, classic, and a gamemode in which the player must beat the level before a White Fog envelops the entire game-screen.

Class List:

This class represents a Player, or User, of the Game
Player.java
This class represents the spike obstacles in the game
Spikes.java
This class represents the Saw obstacle in the game
Saw.java
This class represents the Main Menu in the game, which comes at the beginning of the program
MainMenu.java
This class is where everything is ran out of, and contains the music aspects
Main.java
This class represents the Coin that the Player can pick up!
Coin.java
This class handles what screens are drawn and mouse interaction
DrawingSurface.java
This class is the LevelMenu and users can choose which level to pick from here
LevelMenu.java
This interface instructs the program on how to switch screens
ScreenSwitcher.java
This class represents the ShopMenu and players can buy perks from here
ShopMenu.java
This class represents everything in a level that the player can interact with
Obstacle.java
This class represents the Door in the level

Door.java
This class represents the first Level of the project, and contains all its components
LevelOne.java
This class represents the 2 Level of the project, and contains all its components

LevelTwo.java
This class represents the 3 Level of the project, and contains all its components

LevelThree.java
This class represents the 4 Level of the project, and contains all its components

LevelFour.java
This class represents the 5 Level of the project, and contains all its components

LevelFive.java
This class represents the 6 Level of the project, and contains all its components

LevelSix.java
This class represents the 7 Level of the project, and contains all its components

LevelSeven.java
This class represents the Walls in the levels in our game

Wall.java
This class represents the coin that gives 10 coins in some levels
PowerCoin.java
This class represents the speed boost coin that gives our user a speedboost
SpeedBoost.java
This class represents a Screen in our class and handles most of the common aspects of each level
Screen.java

Credits:

- JaySound Library for Sound, Audio downloaded from Envato Elements.

Vihaan: Javadoc Classes & Methods, Jar-File, Finalized & Came up with many parts of UML (added class connections, new classes, potential & existing fields & methods), Created packages & classes, did basic LevelMenu with buttons), added ShopMenu & button, made it so that esc key goes to main menu, intersection with door makes user go back to main menu, found images for player, saw, spike, made basic LevelTwo, added Music that changes depending on the screen, sound effects for death/door, ideas for powerups/obstacles, came up with the methods concerning rotation in classes, which included multiple methods in Screen() where the object's coordinates would be updated correctly. (such as finding the topLeft corner). Also made sure the rotation worked

seamlessly in every class. Ensured that Spikes and Door will always be facing the right way up by playing with the Image fields. Designed Screen class with common methods for all levels, saving significant time. Came up with level design for levels 5, 6, 7, polished and added elements to most levels.

Kevin: Javadoc Classes & Methods (& added javadoc file and jar file), contributed to UML, Added & Formatted buttons to MainMenu with text, made sure they worked & route to the correct menu by making sure DrawingSurface & Screen work as intended. Made the Entire level One, added all the obstacles (walls, spikes, etc..), players can move around, interact with spikes and saws, restart the level, etc.., powerups. Worked on layout and level design for the majority of levels. Worked on player-obstacle interactions like death, resetting level, gravity, and standing on top of objects without falling or passing through. Play tested every level to make sure a solution was possible for each level.