Kian Ashrafganjouei

Mr. Morrison

ICS4U

June 19, 2020

CPT Report

**Code Features:**

Object Oriented Program -- Used throughout the program.

Abstract classes -- Used in GameState class.

Sorting (Bubble Sort) -- Used in LevelState class (148-164).

ArrayLists -- Used in LevelState class (148-164).

**Work Log**

June 9, 2020:

* Project inception.
* Created the title and a brief description of the game.

June 10, 2020:

* Coded and commented on the Main class (Game) of the program. This class contained the code which created the JFrame.
* Coded and commented on the GamePanel class of the program. This class contained the code which implemented the JPanel.

June 11, 2020:

* Coded and commented on the GameState class of the program. This abstract class contains the methods used in the game state manager.
* Coded the GameStateManager class. This class creates stacks for the different states of the program. Moreover, the first state (MenueState) is called.

June 12, 2020:

* Commented the GameStateManger class.
* Created the MenueState class. This class holds the code for the menu screen. Allowed the user to choose between Start, Instructions, and Exit.
* Implemented the Game Title (Spike Ball) and added the PAC costume font.

June 13, 2020:

* Commented the MenueState.
* Coded and Commented the LevelState. This code controls everything that occurs in the game.
* Created the class for Physics, Player, Platform, Ball, and BallHitBox. These classes are all extensions or compositions of the LevelState.
* Drew the background for the game screen, and created all the platform positions(ground platforms + mid air platforms).
* Created a scores arraylist and wrote the code for the end game (GAME OVER screen: Your Score & Your highscore).
* Wrote code that allowed for the bubble sorting of the arraylist which contained all the scores.
* Worked on the Platform class, which extends the Rectangle class and is a composition of the LevelState class. *NOTE: The rectangle class is a built-in java class*. This class draws the platforms.

June 14, 2020:

* Fixed errors regarding the creation of platforms.
* Fixed errors regarding the scores arraylist and bubble sort.
* Commented the Platform class and commented on the LevelState class.

June 15, 2020:

* Putted code in the Player class that allowed for player movement(arrow keys).
* Created the instructions tab for the MenueState and filled it with the relevant info.
* Fixed some errors for the player movement.
* Created a max jumping and falling speed for the player.
* Worked on the collision detection of the player with the platforms.
* Fixed errors with player collision with the platforms.
* Commented the Player class.

June 16th, 2020:

* Added code in the Ball class that allowed for ball movement.
* Added the feature that made the ball bounce on the edges of the screen.
* Added the variable size and speed feature of the ball.
* Added code for BallHitBox that created the box around the ball for player detection.
* Fixed the variable size and speed feature of the ball hit box.
* Added the collision detection between the ball and the player.
* Made a reset function that rests the game and changes all the variables accordingly.

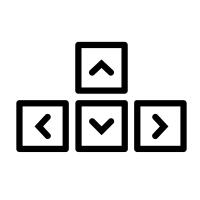
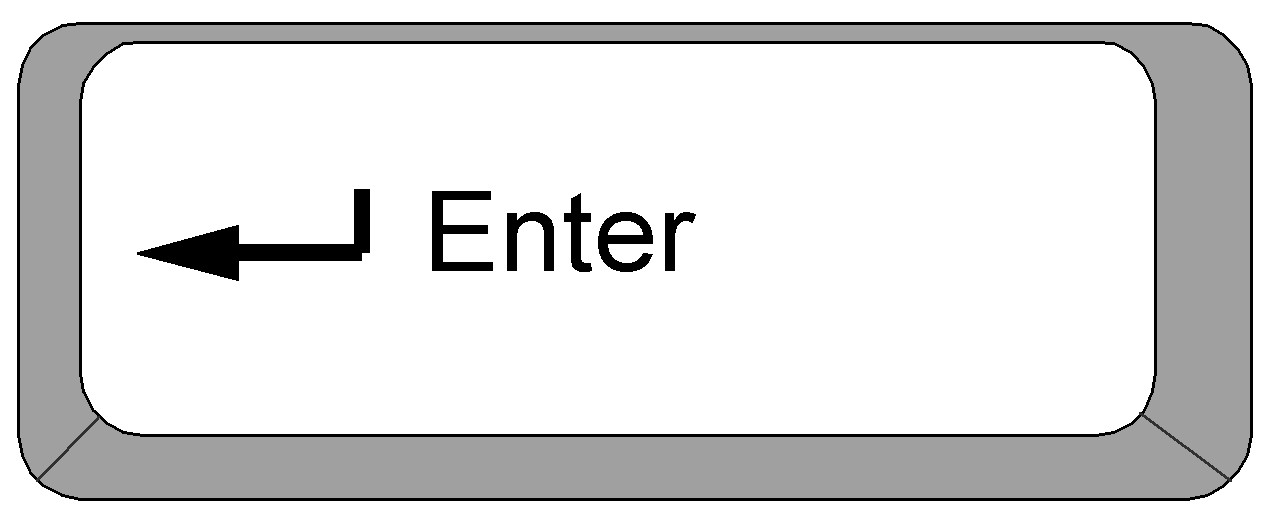
June 17th, 2020:

* Added a score counter to the top right corner.
* Got some feedback from different users.
* Attempted to break the game.

**User Manual**

*CONTROLS*

UP Arrow Key -- Jump

LEFT Arrow Key -- Move Left

RIGHT Arrow Key -- Move Right

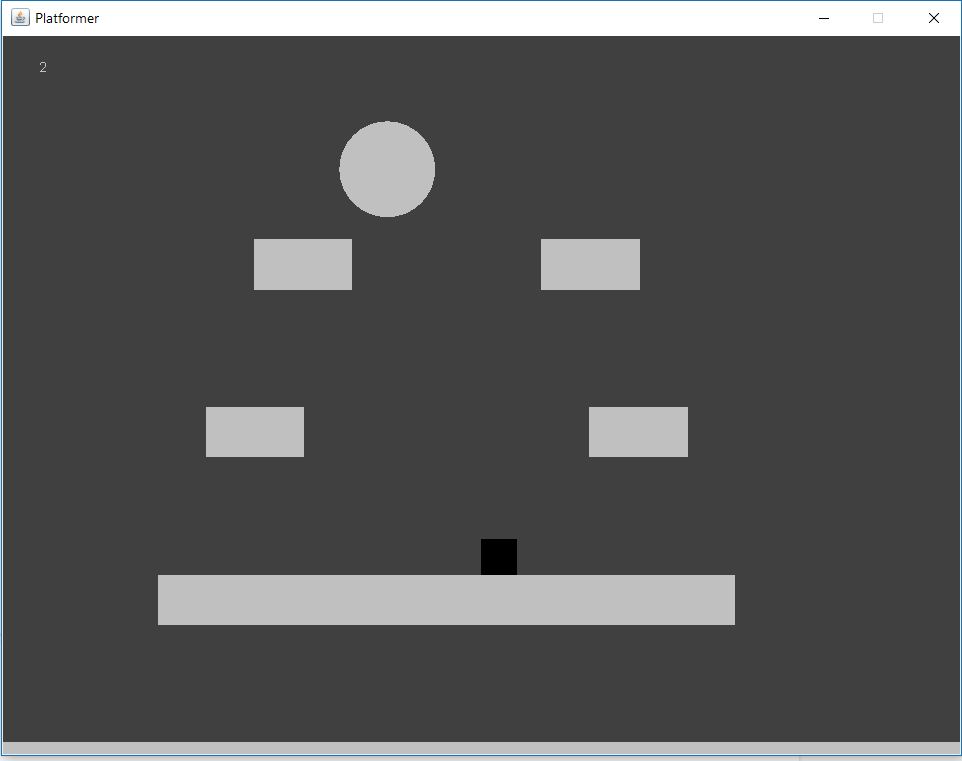
ENTER Key -- Restart Game

*HOW TO PLAY*

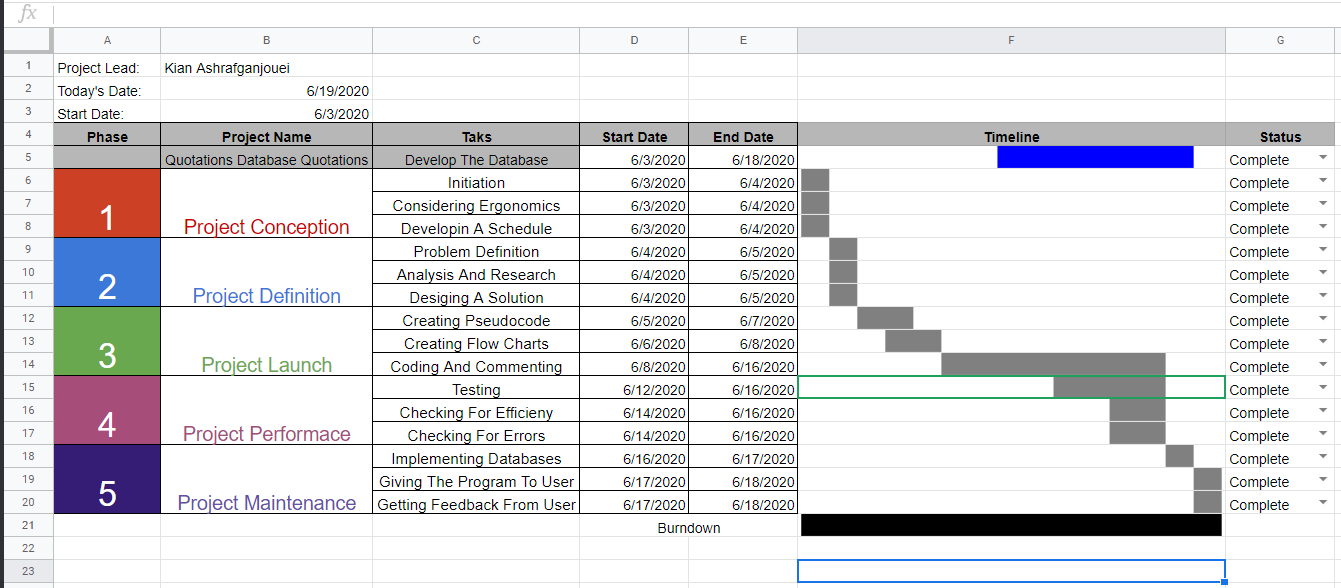
* Control the black square to evade the spike ball. If the ball hits the square, it is game over.
* Your score is shown on the top left.
* The score increases by 1 every time the ball bounces.
* As time goes on, the spike ball gets bigger and faster.

*TIPS & TRICKS*

* Use the white platforms to your advantage and get into better positions.
* Do not jump constantly as your jumps will become weaker.
* Try to predict the motion of the ball, that's the best way to get a good score.



**Gantt Chart:**



**Test Log:**

*TEST 1*

Tester Name: Ilya Shahrestani

Date Of Test: June 17, 2020

Game Rating (1-Bad, 5-Good): 4.5

QUESTIONS:

1. Did you encounter any errors during your playthrough?

* Small error, when you die and keep holding the left or right arrow key, you keep moving in that direction. There are no game-breaking bugs.

1. What did you most enjoy about the game?

* The fast-paced gameplay was very intriguing. The fact that the ball’s speed and size increase is very interesting.

1. What would you improve about the game?

* I would like to see more maps included in the game. Moreover, the maps should contain more feature as the one in the game is very plane.

1. Anything else you would like to add?

* The player movement and control is simple, smooth, and enjoyable.

*TEST 2*

Tester Name: Keyvan Ashrafganjouei

Date Of Test: June 18, 2020

Game Rating (1-Bad, 5-Good): 4

QUESTIONS:

1. Did you encounter any errors during your playthrough?

* No errors encountered

1. What did you most enjoy about the game?

* Trying to avoid the ball was a unique and good feature of this game.

1. What would you improve about the game?

* You should give the player the option to change the difficulty of the game.

1. Anything else you would like to add?

* Some music would be a great addition to the game.

**Error Log:**

*Error 1*

Error Description: The error is no points found error created with the polyformic array that creates the platform objects.

Date Of Error: June 14, 2020

Cause: Because the Physics class is a composition of the Platform class, it needs to run through every single platform object and determine if it is being contacted by the player. However, if a certain number of platforms are declared and not all of them are filled, the Physics class will have no points to check and thus the program will crash at that instance.

Fix: Ensure that all of the platform objects that are being declared are given locations values.

*Error 2*

Error Description: The arraylist which contains the scores was not being sorted properly with the bubble sort.

Date Of Error: June 14, 2020

Cause: The code was copied from one of the posited templates. However, the original code was intended for arrays not arraylists thus some minor adjustments had to be made. After the adjustments, the code did not sort the arraylist properly.

Fix: The inner for loop condition was missing a (-1), after the change the code sorted the arraylist properly.

*Error 3*

Error Description: When the player is on a platform, they are not able to move left or right.

Date Of Error: June 15, 2020

Cause: Because the collision detection between the platforms and player occurs a stage later than player movement, the square may get stuck inside the platform and restrict the left or right movement.

Fix: Ensure to adjust player location if a platform collision is detected.

*Error 4*

Error Description: After copy and pasting the code that was used to detect the collision between player and the platform to use to detect collision between ball and player, the program crashed.

Date Of Error: June 16, 2020

Cause: The method in the Physics class that detects collision is only able to work with two square objects. Having one ball and one square will crash the program.

Fix: For this reason, another class was made named BallHitBox which created a square with the same characteristics as the ball(location, size, speed). This square was used for collision detection.

*Error 5*

Error Description: When the match ends, the program does not stop running, the player loses control over the square and information regarding the match is displayed. However, the ball does not stop moving and will continue to get faster and bigger. This is a style choice. But, because the ball is still bouncing, the score in the end screen will continue to increase, moreover, the top right score counter will continue to increase.

Date Of Error: June 17, 2020

Cause: In the end game screen the program still runs the ball movement and if the ball bounces the program will increase the score accordingly.

Fix: Put a lock-system that will allow that will capture the score at the moment of contact with the ball and display that information. This will not prevent the increase to the score, however the program will still run as intended. Lastly, a mask layer was put in place in the end game screen to make the top right score counter invisible.

*Error 6*

Error Description: After implementing the restart function, i have noticed that the momentum of the ball and player carries over to the next session. For instance, if the player is jumping and they press the ENTER key, in the new session they will continue to jump. This will however cause the player and the ball to collide on certain occasions, which will result in an endgame screen early one.

Date Of Error: June 17, 2020

Cause: After hitting the ENTER key, a function resets the ball size, speed and location. The function also resets player location. However, the ball’s momentum (y increase, x increase, y decrease, x decrease) are not changed.

Fix: Adjusted the player and ball spawn location to avoid collision depending on previous momentum status.