Appraisal_Yang Liuxin(19206207)

liuxin.yang

December 2020

1 Design and Cohesion

Overall, I consider my design and cohesion is excellent. Each class is used for sensible responsibilities.

1.1 Packages

I divide my code into two packages. One is called 'Code', which contains the majority of the functionalities and key features. Another is called 'Presentation', which contains codes to do present work.

1.2 Classes

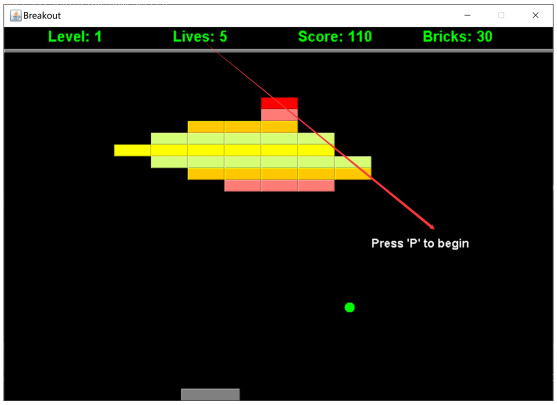
Fourteen classes are used in total. The Game class is where the game runs by calling the run() method. The run() method keeps running until certain requirements are not satisfied. The updateGame() is inside the run() method. Moreover, the updataGame() method contains many other methods to update the game each turn through the loop. The Bonus class is related to implementing four special bonuses, excluding the multiBall function which is implemented in the Ball class. The Ball class do the work of hitting, bouncing, moving and multiBall. The main work of the Brick class is to initialize bricks according to different levels. Moreover, the Brick class also contains relationship with bonuses as the type of Brick is an indicator whether this brick contains a special bonuses or just no bonus. The Paddle class can do movePaddle function according to the keyboardListener. The paddle class also does part of the work of widenPaddle because a checkTime() method is customized for narrowing the width of the paddle when time expires. The playerListener class does the work of user interaction as it connects the game with users by the keyBoard listener.

The window class The GameScreen class contains a paintComponent method and does most of the drawing work in the game interface. The MainMenu class contains a run() method to do some of the interaction work and interface transitional work. The Menu class draws everything in the menu. the AboutScreen and ScoreScreen class draw texts in the 'help' and 'Breakout Hall of Fame' interface respectively. The ScoreKeeper class maintains a score array and whenever higher scores come in, it will replace the lowest value in the array. The PersistantScoreKepper reads and writes things in the score.txt.

2 Input

Overall, I consider my code in input as excellent. My code adopts the **keyboard listener** and user types can be implemented by this listener.

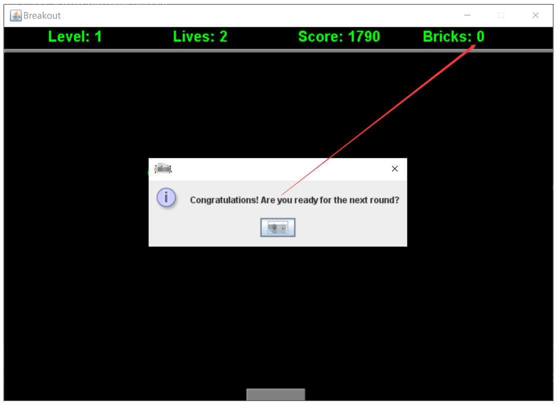
You can enter 'pause' at any time as long as your still have life.



Level: 1 Lives: 5 Score: 110 Bricks: 30

Press 'P' to begin

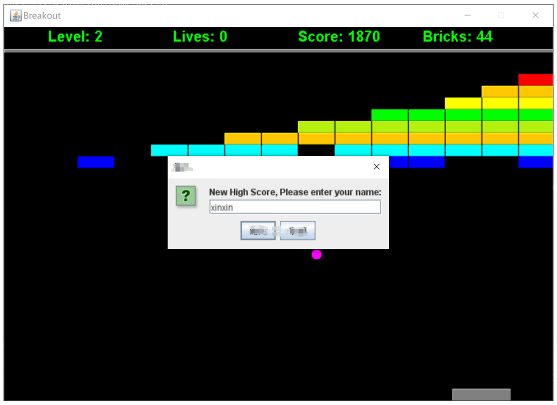
When you complete the current level, you can choose when to enter the next round at will.



Level: 1 Lives: 2 Score: 1790 Bricks: 0

Congratulations! Are you ready for the next round?

When you achieve new high scores, you can enter you name for the hall of frame to remember you.



New High Score, Please enter your name:

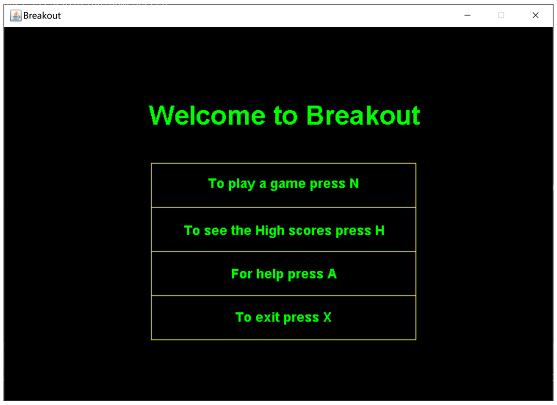
xinxin

3 Display

Overall, I consider my code as excellent in displaying objects. Each object is drawn according to the accurate coordinates.

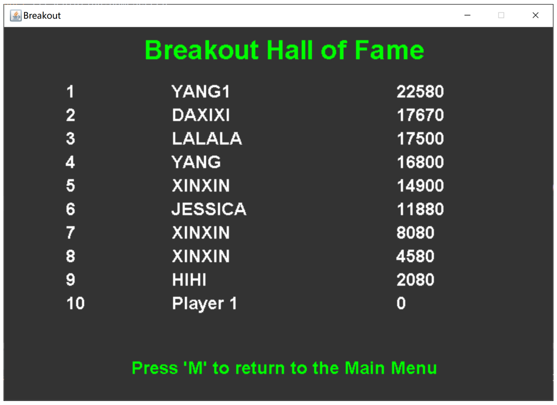
4 Menu

Overall, I consider my code as excellent in displaying menu. Before gameplay starts, a main menu is shown at first which allows you choose path, such as playing game, hall of fame, an info showing the controls for the game or just let you exit.



5 High Scores

Overall, I consider my code as excellent in storing high scores. Previous high scores are loaded from a file called scores.txt and any new high scores are saved in the file when a game is completed.



Breakout Hall of Fame

1	YANG1	22580
2	DAXIXI	17670
3	LALALA	17500
4	YANG	16800
5	XINXIN	14900
6	JESSICA	11880
7	XINXIN	8080
8	XINXIN	4580
9	HIHI	2080
10	Player 1	0

Press 'M' to return to the Main Menu

6 Bouncing

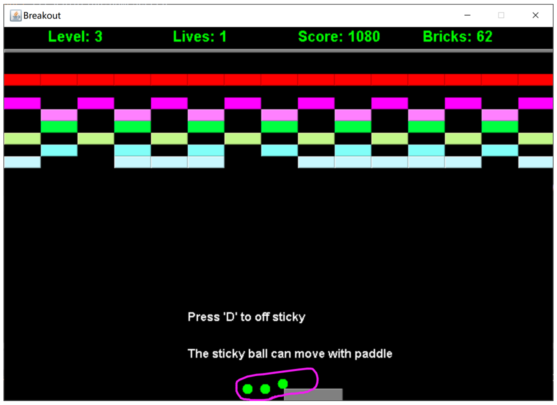
Overall, I consider my code as good in bouncing, not very satisfying to me. Bouncing the right wall does not work exactly the coordinate owing to bar problems. I do not know the exact x coordinate the right border so I can just make guesses about it. Additionally, since my ball is conceived as a Rectangle to check whether hit conveniently, almost everything works well except when the ball hits the side point of the paddle. It may make it seemingly has not hit the paddle but actually it has. However, in order to mitigate this problem, I change my the y coordinate of the paddle Rectangle a little bit larger. That is why I consider I have found out a solution to deal with visual mistake, though I do not know whether this solution is preferred by my professor.

7 Bonuses

Overall, I consider my code as excellent in implementing bonuses. There are five kinds bonuses in my game. Four of five bonuses are with special effect and the other is common bonus without special effects and they are drawn in cyan and magenta respectively.

7.1 Multi-ball

This bonus can be implemented when this bonus is collected when the game is running. However, I set the maximum number of multi balls to prevent some latency problems. Each ball will be separated into three balls.



Level: 3 Lives: 1 Score: 1080 Bricks: 62

Press 'D' to off sticky

The sticky ball can move with paddle

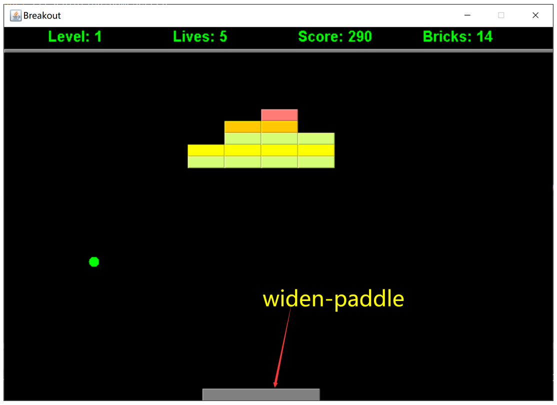
7.2 Widen-paddle

When collecting the widen-paddle bonus, the width of the paddle gets larger, stretching from the right end point further. I find it a pity that I did not implement it to stretch from the middle from to both sides. That is where I still need improvement.

For checking time, I wrote a checkTime() method myself rather than using the Timer, which gives me more control and freedom to customize my own function.

Once the time expires, the width of the paddle decreases from the right side to the middle until it gets back to the initial width.

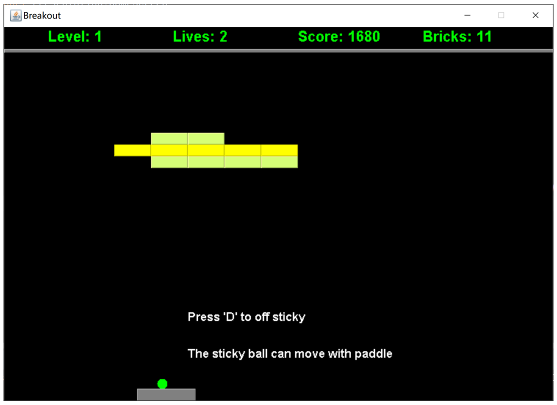
Moreover, if the paddle collects the widen-paddle bonus and sticky-paddle at the same time. If the time for widen-paddle bonus expires, the paddle does not leave the ball outside. Instead, the ball will move with the paddle with the same speed as its width decreases.



Level: 1 Lives: 5 Score: 290 Bricks: 14

7.3 Sticky-paddle

When collecting the sticky-paddle bonus and a ball hits the paddle, this particular ball is then glued to the paddle. The glued ball can move with the paddle and all other balls remain still at their locations. Only when users type ’D’, all of the balls can continue to move.



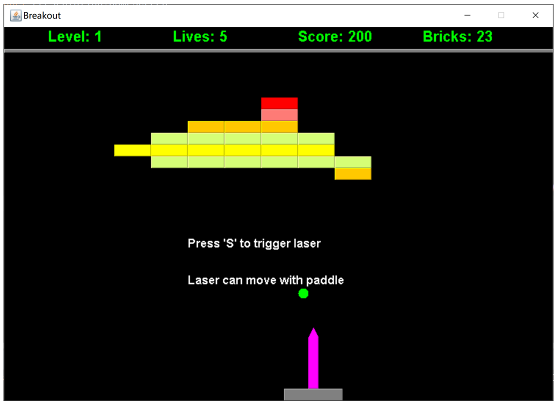
Level: 1 Lives: 2 Score: 1680 Bricks: 11

Press 'D' to off sticky

The sticky ball can move with paddle

7.4 Laser

After collecting the laser bonus, the laser will be shown in the screen. It can move with the paddle and is not triggered until users type ’S’ to trigger it. Moreover, any listener in advance will be ignored. In other words, if users enter ’S’ before collecting a laser bonus, this listener will be ignored. This is where I think I take into account.



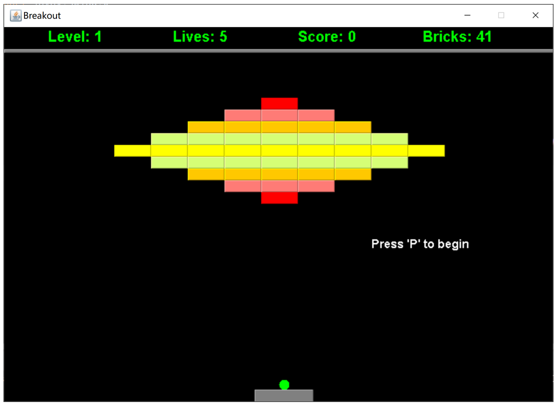
Level: 1 Lives: 5 Score: 200 Bricks: 23

Press 'S' to trigger laser

Laser can move with paddle

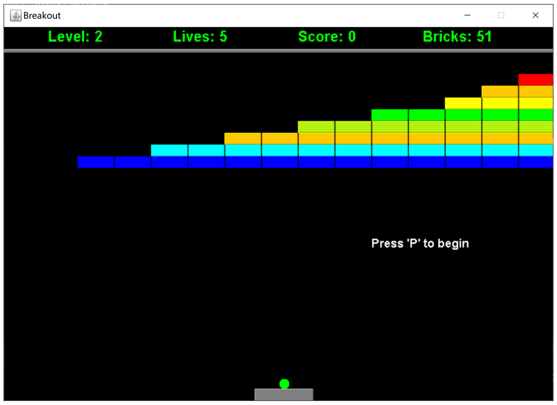
8 Levels

Overall, I consider my code as excellent in initializing levels. 5 levels are implemented and the difference between them is their layouts.



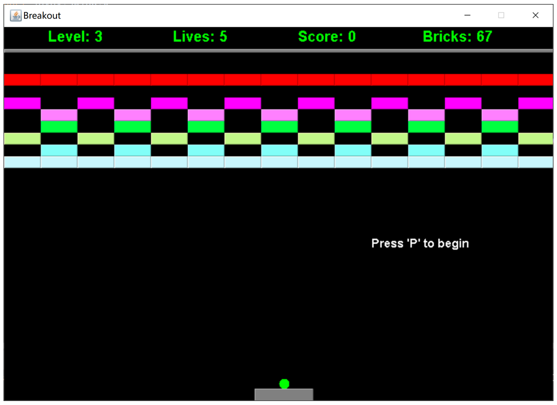
Level: 1 Lives: 5 Score: 0 Bricks: 41

Press 'P' to begin



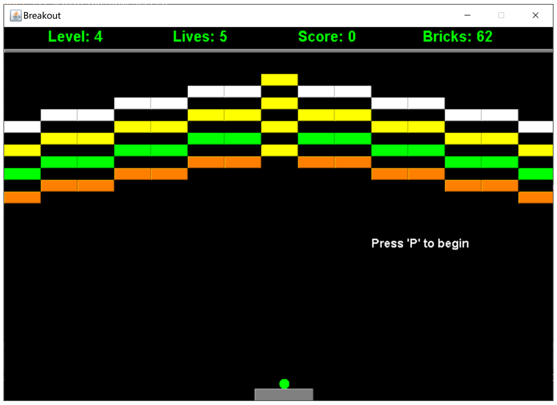
Level: 2 Lives: 5 Score: 0 Bricks: 51

Press 'P' to begin



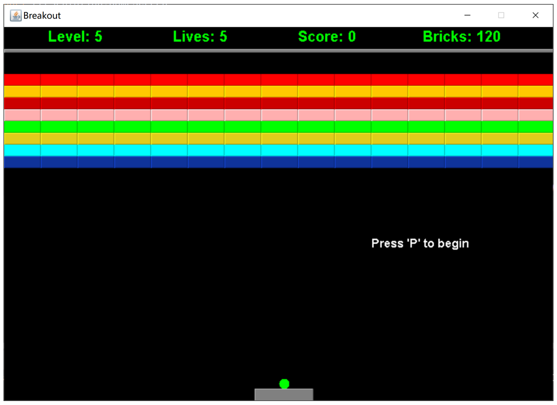
Level: 3 Lives: 5 Score: 0 Bricks: 67

Press 'P' to begin



Level: 4 Lives: 5 Score: 0 Bricks: 62

Press 'P' to begin



Level: 5 Lives: 5 Score: 0 Bricks: 120

Press 'P' to begin

9 Appraisal

Overall, I consider my code as excellent in writing my appraisal. Detailed description are included in this pdf. I showed where I still need improvement and highlighted where I met and exceeded the project specifications.

10 Video-Quality

Overall, I consider my code as excellent in writing my appraisal. My video clearly shows the execution of my code, staring the game and navigating the menu. Additionally, my video shows the bricks being destroyed, bouncing on walls/bricks and paddle (change direction).

11 Appendix

I used IDEA to write my codes.