29/10/2023

CBL ASSIGNMENT

1. BASICS OF THE GAME

The code consists of 4 different files; Main.java, GamePlay.java, UserInput.java and MapGen.java

A. Main.java

In main.java the code is started and GamePlay.java and MapGen.java are called. The window is also made in main.java.

B. MapGen.java

In MapGen.java a boolean matrix is created establishing & handling the visibility/existance of each brick. At the start of a new game all bricks will be created according to the specified perameters for the amount of rows & columns of bricks. The boolean value for each brick will be set to "true" so that they can be drawn & checked for collision with the ball.

C. GamePlay.java

GamePlay.java handles the whole gameplay loop. Upon starting a game, a new instance of MapGen is created. Subsequently the paddle & ball's positions are reset to default and the ability to start another game is disabled until the current game is finished.
An event loop timer is started so that every tick it will increase the position of the ball by the amount specified as the speed, which is split up into vertical speed and horizontal speed. It then checks for collision with the paddle & every brick by checking if their shapes intersect. It also checks for collisions with the window border by comparing position of the ball.

If the ball's position is detected in the bottom 5% of the window (= below the paddle) then the game over condition is triggered. If no blocks are visible anymore, then the win condition is triggered.

D. UserInput.java

UserInput.java handles the input of the user. It determines what buttons are pushed and what the correct follow-up action is. This action is executed by calling a method from another class.

2. USER EXPERIENCE

The first topic of choice is the user experience. The most important part of a good user experience for every game is of course a game that works the way it is supposed to, and does not give any error messages. Making this part of the user experience optimal was the number 1 priority. The first two things to ensure this is a starting screen, so the user can decide when they are ready to start playing, the second thing was an ending screen, displaying the outcome of the game: you win/lose.

After a while, we started to notice that we probably would not have a lot of time left to dedicate to the user experience after the whole game was finished. That is why, during the development, multiple people have been asked to try out the game and give their input on what things they would like to see. The decisions of how the development was furthered were mainly based on these answers. The questions asked were for example, what is more important to you, this or that, or, which colors do you prefer etc. These rounds of questions were asked to at least five people, who all varied in age, interest, technological fluency etc. Because of this approach the game is quite user-oriented, for the time spent on it was always spent on developing what was necessary for a good user experience.

3. VERSION CONTROL

The second topic of choice was version control. This was done with the help of GitHub. However, all the programming work was done in a VScode live share environment as we always were working on this at the same time. However, we did keep up with the version control on this GitHub, so that we would be able to revert to a previous version of the code in case of major issues.

Note that due to a mistake, the new versions were pushed to the master branch, rather than the main branch. However, this did not hinder the results, and thus, not much time was spent on trying to fix it.

During the project, there was one instance where we worked separately on the project. However, we neglected to communicate this with each other, which resulted in major merge conflicts in the repository. Instead of attempting to salvage the repository, we created a new one. We have learned to always be sure to check if any changes have been made to the repository, before committing & pushing your own changes.

Repository: https://github.com/maxmax187/brickbreaker-2IP90/activity