**TERM PROJECT: HIT THE BOX**

**CSE1142 Computer Programming II, Spring 2021 Date Submitted: May 28, 2021**

**FARUK AKDEMİR 150119012**

**ÖMERCAN SABUN 150119555**

**1-)**Problem Definition

The diversion framework comprises of 10x10 boxes. Each box distinguished by utilizing zero-based. The client will utilize the mouse to crush the boxes. When a user clicks on a box to crush, it'll too influence neighboring boxes (right, cleared out, up, and down boxes).

I can definitely say that one of the most challenging parts of the game was the cell logic and point system. Adjusting the scoring system and continuing until there were no mirrors and wooden-type boxes left, even if there were points - when it was finished, it forced us to show for the button to move to the new level.

But the hardest part of the game was the precision Save and Load Game logic. A txt for this. The file is written and the last file of the cell is written .txt, which we created a clone of the cell, after the level you stayed and the hamlets you made when you closed the game. With the continuation game, it is provided to continue a game with this written file. The style files and the design phase of the game were not difficult. We were able to adjust the animations, color palettes and text and button sizes we wanted.

Since we thought it would be fun, we hid easter eggs in our game. They're not really surprising though, because we told you their location, but one is still hidden.

We have separated each section in order to establish the basis of the game and not to get confused. We explained all the codes as detailed as we could and talked about their logic. As an extra, we set up an account and posted our promotional video on Youtube. I hope you like our game, thanks in advance.

**2-)** Implementation Details

UML DİAGRAM

|  |
| --- |
| SaveAndLoadData |
| Load\_level(level)  Save\_highScore  Load\_highScore  SaveGame  LoadGame |

|  |
| --- |
| Cell Logic |
| Score  High score  Level  Columns  Cells[rows][columns] |
| Set\_cell\_logic  Calculate point  Check finsh point  Check finsh level |

|  |
| --- |
| Main |
| stage |
| render menu  render level(lv)  start menu  start level |

|  |
| --- |
| StackPane |

|  |
| --- |
| Pane |

|  |
| --- |
| Cell |
| position  life |
| render\_cell()  animate()  HighLight\_hover()  Un\_HighLight\_hover() |

|  |
| --- |
| Grid |
| width  heigth  rows  columns  cells[rows][columns] |
| add\_cell(cell)  check no wood and mirror |

10x10 1

3-)TEST CASES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Step to test | Expected result | Test result | Result |
| 1 | 1.Check load grid data from file is correct | 1. Change level parameter in StartLevel(level) to 1 | data in file have loaded in to grid | data in file have loaded in to grid | OK |
|  |  | 2. Start game |  |  |  |
|  |  | 3. click New Game |  |  |  |
|  |  | 4. check data in file , check data in grid |  |  |  |
|  |  | 5. test another level |  |  |  |

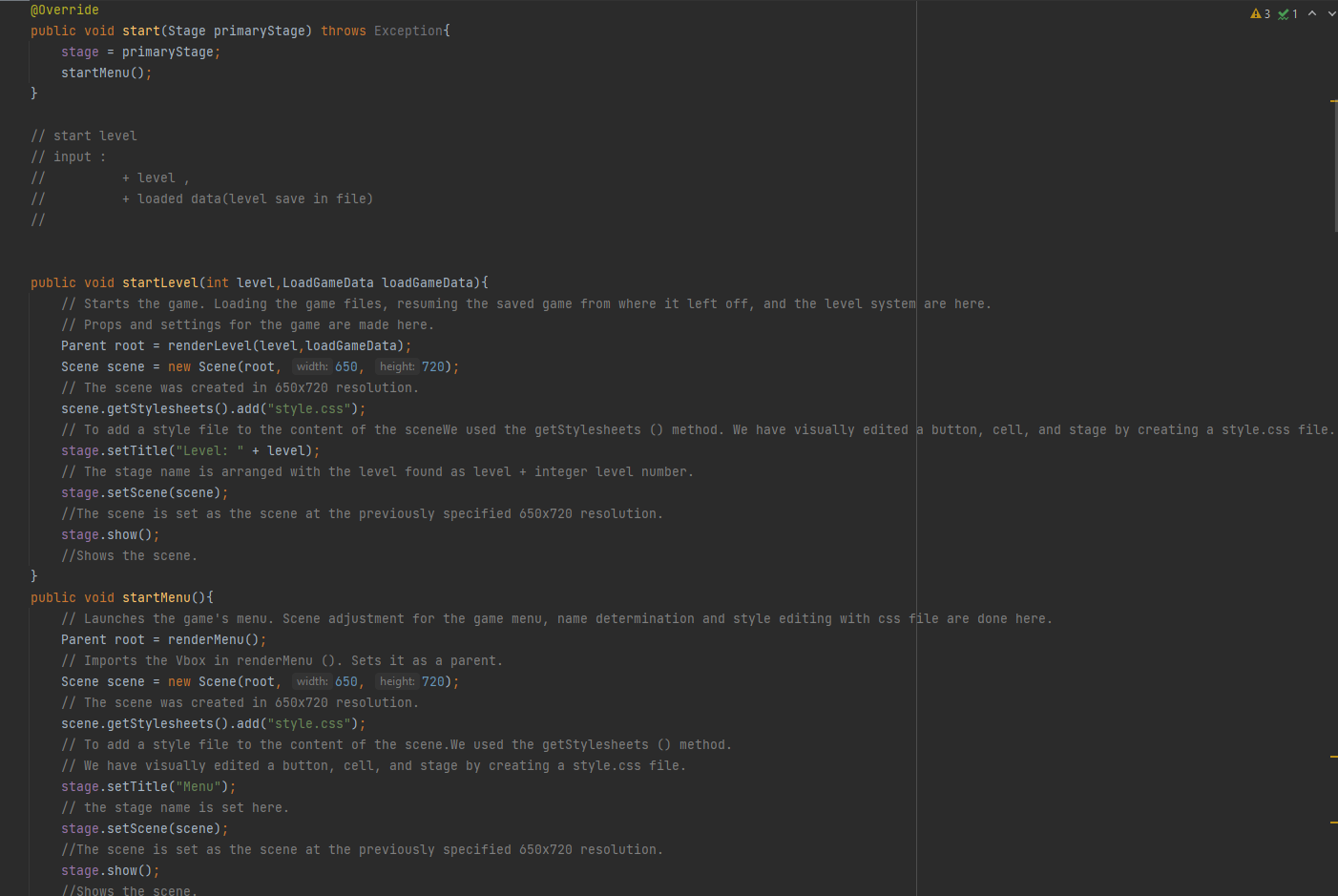
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Step to test | Expected result | Test result | Result |
| 2 | 2. check logic when clicked box is | Click wall box | Nothing happen | Nothing happen | OK |
|  |  | click empty box | Nothing happen | Nothing happen | OK |
|  |  | click mirror box have 1 mirror neightbour box | box and neighbourbox -> empty box | box and neighbourbox -> empty box | OK |
|  |  | click mirror box have 1 wood neightbour box | box -> empty, neighbourbox -> mirror | box -> empty, neighbourbox -> mirror | OK |

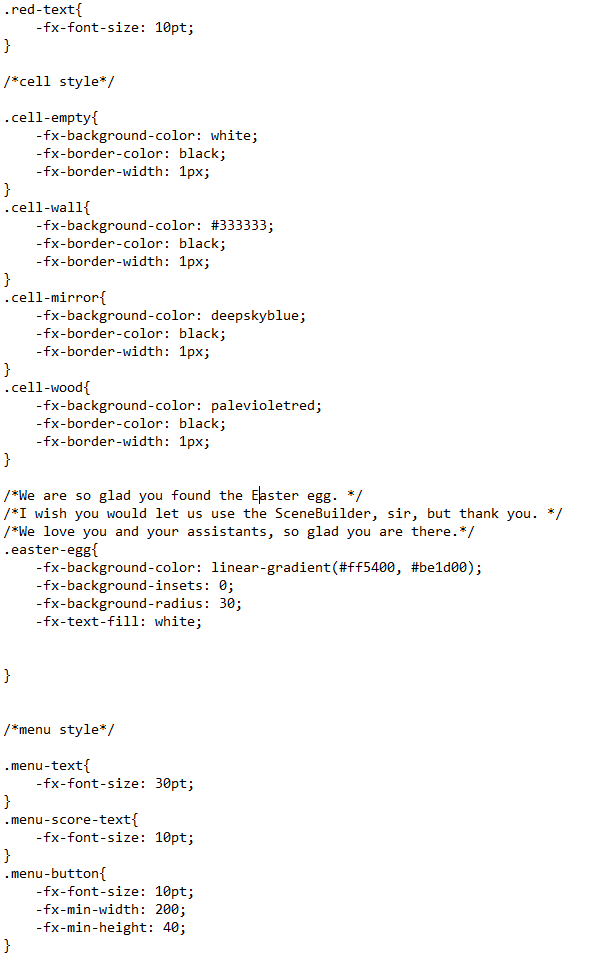
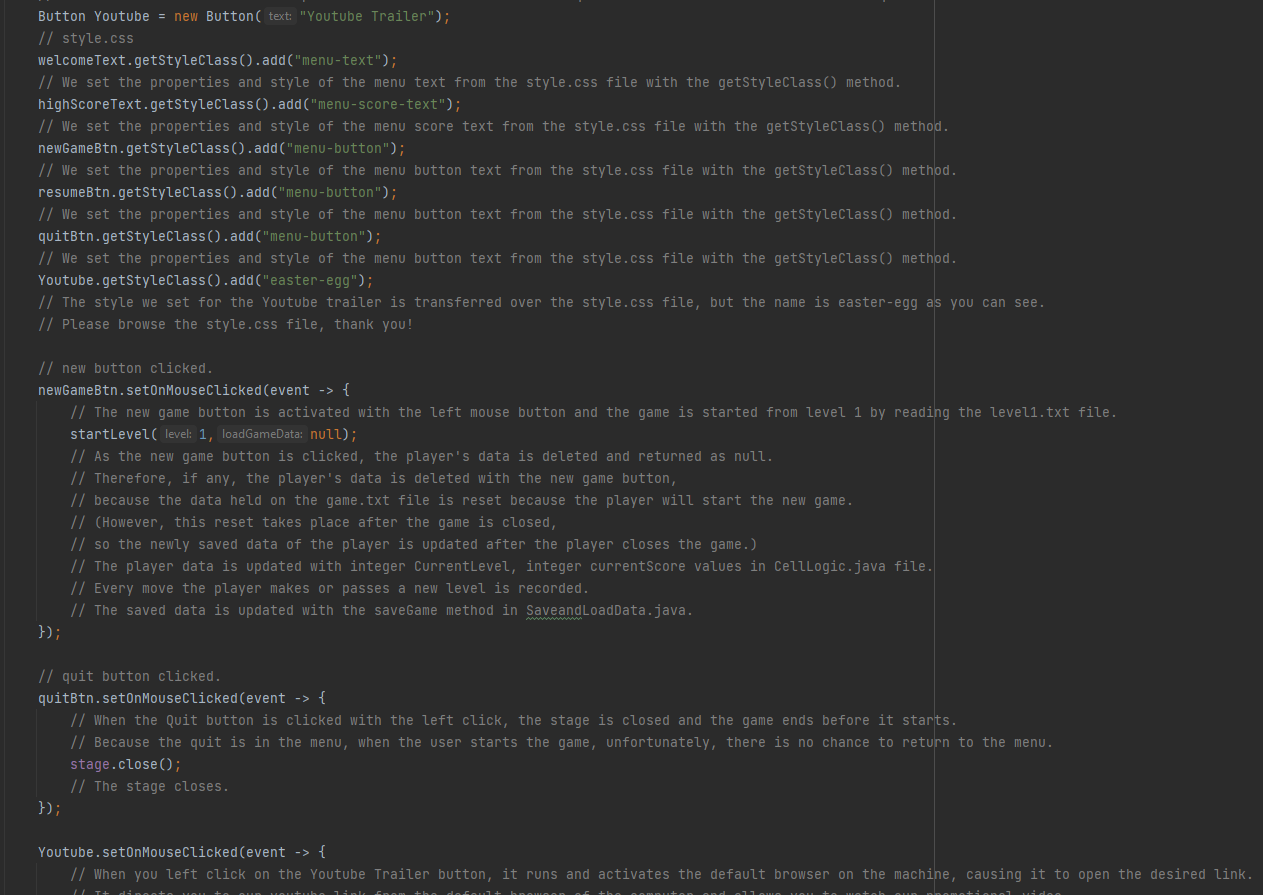
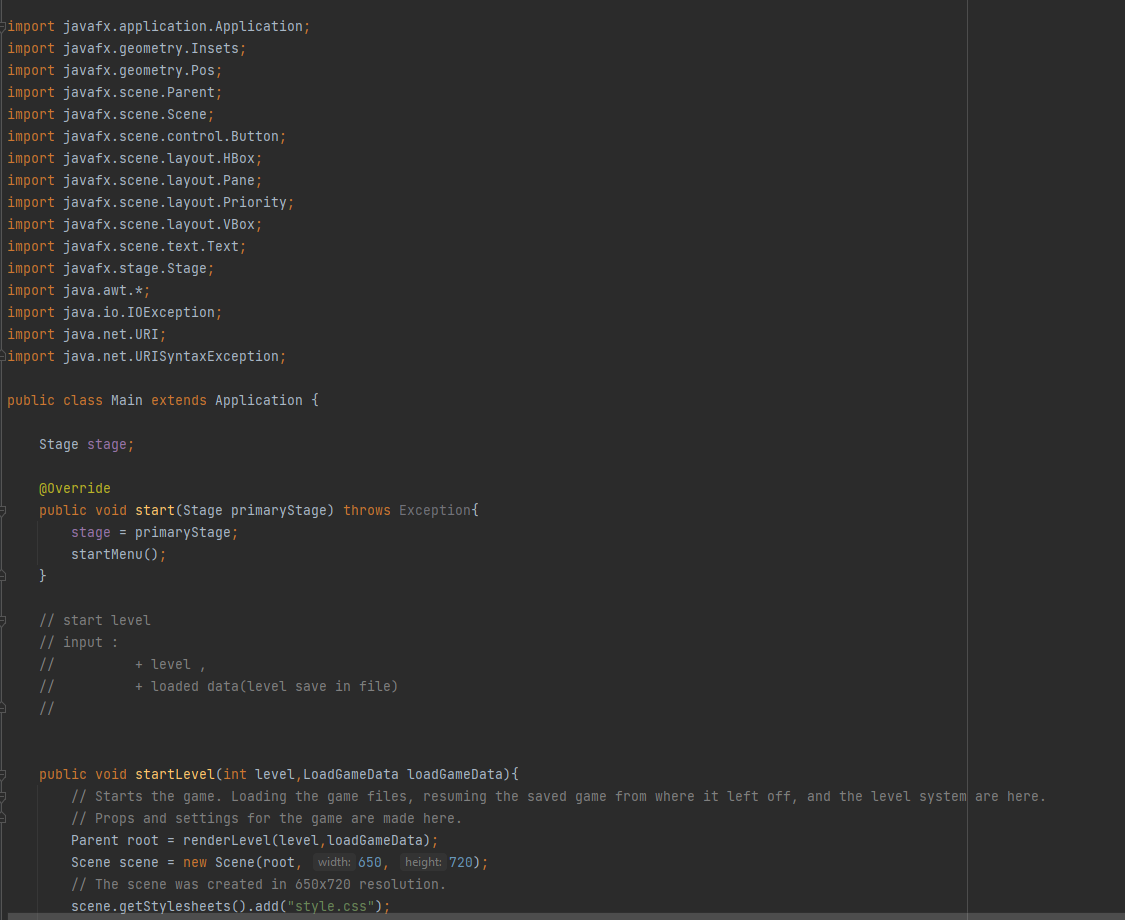
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Step to test | Expected result | Test result | Result |
| 3 | 3. Check menu button | click newgame button | show game level 1 | show game level 1 | OK |
|  |  | click resume button | show game level 1 , score, grid loaded from file game.txt | show game level 1 , score, grid loaded from file game.txt | OK |
|  |  | click quit button | window close | window not closed | BUG (Hotfixed) |

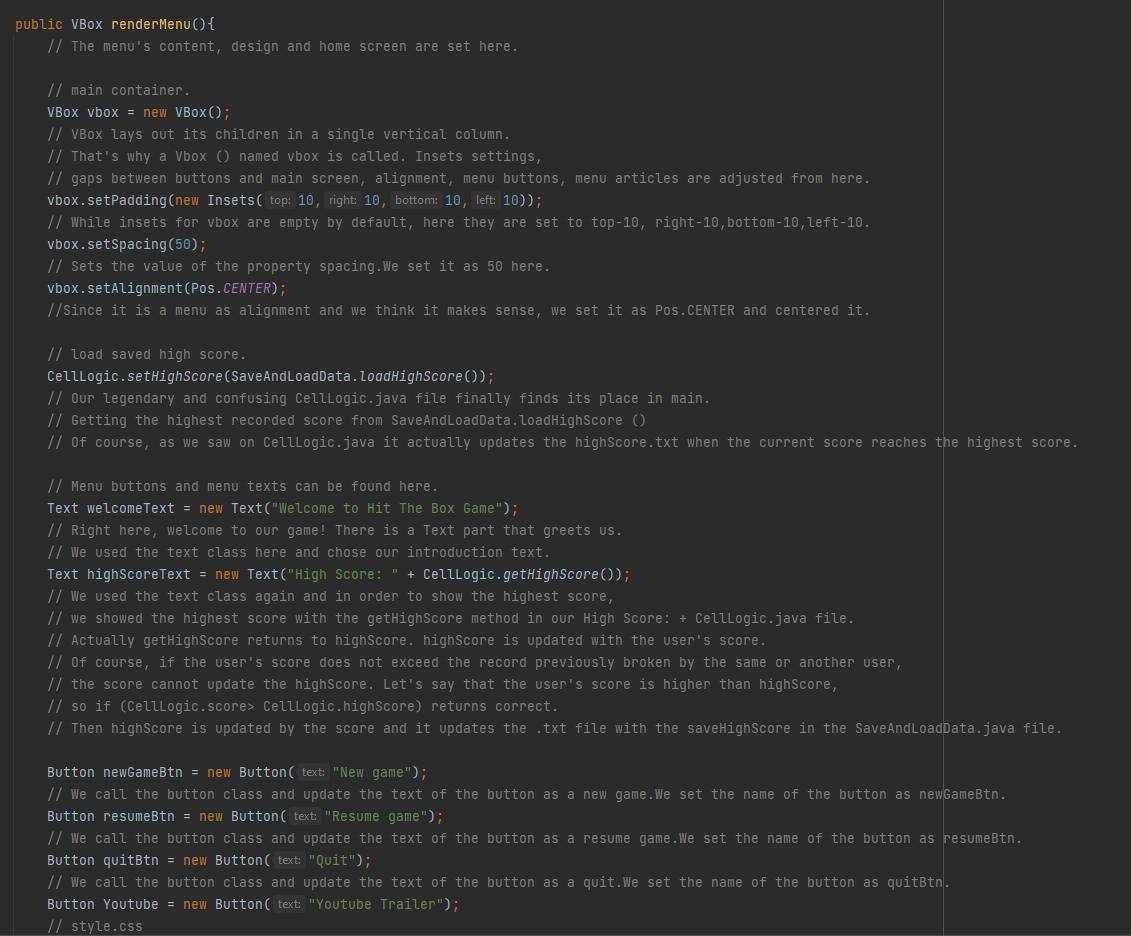
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Step to test | Expected result | Test result | Result |
| 4 | 4. Check data is saved when click X button | 1. Play new game | data in game.txt is loaded to the grid | data in game.txt is loaded to the grid | OK |
|  |  | 2. click 1 mirror cell | score in game.txt is loaded | score in game.txt is loaded |  |
|  |  | 3. click X button to close window | current level in game.txt is loaded | current level in game.txt is loaded |  |
|  |  | 4. check file game.txt |  |  |  |
|  |  | 5. Open menu , click resume game |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Step to test | Expected result | Test result | Result |
| 5 | 5. check high score is loaded | 1. play game till highscore change | high score = high score before quit game | high score = high score before quit game | OK |
|  |  | 2. quit game |  |  |  |
|  |  | 3. open game |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Step to test | Expected result | Test result | Result |
| 6 | 6. Check button nextLevel is working | 1. play new game | button nextLevel is not showing | button nextLevel is not showing | OK |
|  |  | 2. finish level 1 | button nextLevle is showing | button nextLevle is showing | OK |
|  |  | 3. clicked button | show level 2 | show level 2 | OK |
| 7 | 7.Check bottom Text is showing correctly | Click 1 cell |  |  |  |







All the photos were also used in the promotional video. Details are in the video and in the comment section of the project.