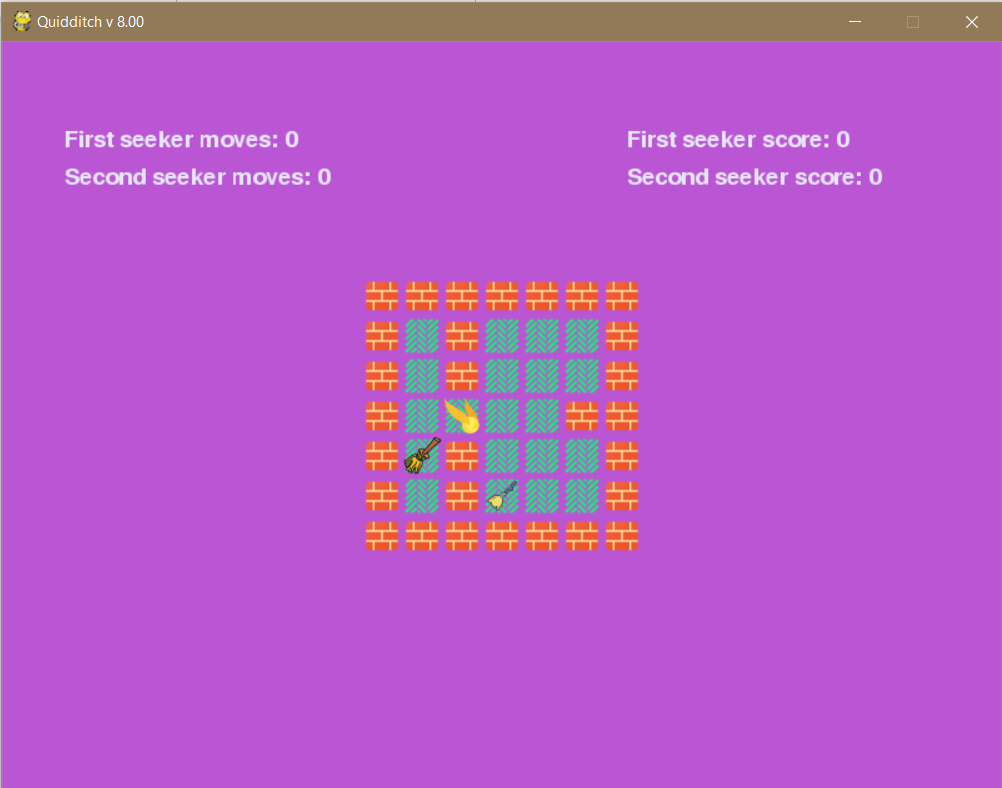
**Test Plan and Log**

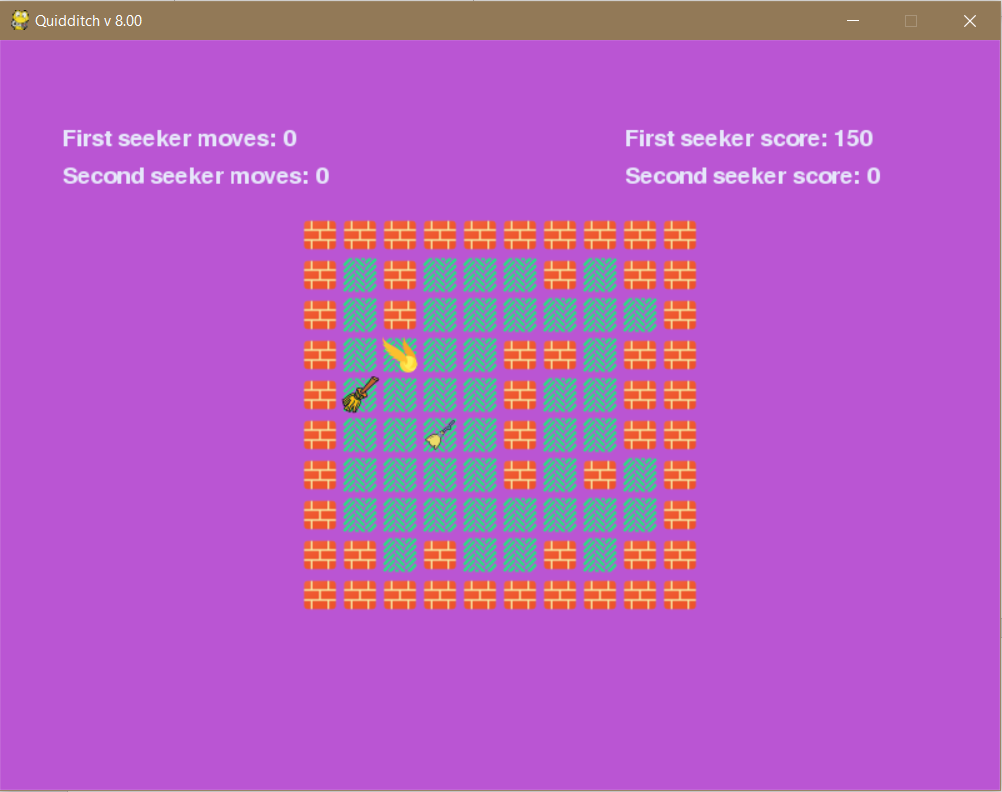
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Action** | | **What should happen** | **Actual** | **Pass/ Fail** | **Remediation** |
| 1.Run the top level | The field with Snitch and both seekers are displayed. Score and moves are printed as well | |  | Pass |  |
| 2.Press the *Up arrow* key | Seeker1 moves up,  First seeker moves = 1 | |  | Pass |  |
| 3. Press the *Down arrow* key | Seeker1 moves down,  First seeker moves = 2 | |  | Pass |  |
| 4. Press the *Right arrow* key | Seeker1 do not move, as there is a wall,  First seeker moves = 3 | |  | Pass |  |
| 5.Press the *Up arrow* key and *Right* *arrow* key Catch snitch on the first level (there is no *End Game screen* yet) | Go to level2 | | Doesn’t go to the second level when snitch is caught | Fail | Add if statement to *for event in pygame.event.get():* checking whether Snitch is 0. If yes, than go to level2 |
| 6.Catch snitch on the first level (there is no *End Game screen* yet) | Go to level2 | | Goes to the level2, resets the moves to 0. | Pass |  |
| 7. Press the *Up arrow* key and *Right arrow* key | Seeker1 moves up (moves = 4) and right, catches snitch and game switches to level2,  Moves = 0,  Seeker1 score = 150 | |  | Pass |  |
| 8. Press *Right arrow* key and *Left arrow* key | Seeker1 moves right and left, moves = 2 | |  | Pass |  |
| 9. Start game, Press *d* key | Seeker2 moves right,  Moves = 1 | |  | Pass |  |
| 10. Press *a* key | Seeker2 moves left,  Moves = 2 | |  | Pass |  |
| 11. Press *w* key | Seeker2 moves up,  Moves = 3 | |  | Pass |  |
| 12. Press *s* key | Seeker2 moves down,  Moves = 4 | |  | Pass |  |
| 13. Catch Snitch on the second level (there is no *End Game screen* yet) | Snitch is caught | | Cannot catch Snitch, every time seeker tries to, both seekers Moves go to 0 and both seekers go to the initial position for level 2 | Fail | Add if statement that catches snitch if level == 2 |
| 14.  Catch Snitch on the second level (there is no *End Game screen* yet) | Snitch is caught | |  | Pass |  |
| 15. Follow Snitch on level2 | Snitch is moving | | Snitch moves, but sometimes duplicates, sometimes ‘eats’ seekers or walls. Cannot catch Snitch | Fail | Modify moveSnitch() function adding 2 new variables – row and col to store the random number for row and column there, so that it won’t randomize several times inside the function. |
| 16. Follow Snitch on level2 | Snitch is moving | |  | Pass |  |
| 17. Catch the Snitch on level2 | End screen  *No counter at the moment* | |  | Pass |  |
| 18.Seeker1 catches the Snitch on level1 | Go to level2, seeker1 score = 150 | |  | Pass |  |
| 19.Seeker1 catches Snitch on level2 | End game screen,  seeker1 score = 300,  seeker1 – the winner | |  | Pass |  |
| 20.Seeker2 catches the Snitch on level1 | Go to level2, seeker2 score = 150 | |  | Pass |  |
| 21.Seeker2 catches the Snitch on level2 | End game screen,  seeker2 score = 300,  seeker2 – the winner | |  | Pass |  |
| 22.Seeker1 catches the Snitch on level1, seeker2 – on level 2 | End screen, seeker1 – 150 points,  Seeker2 – 150 points,  1:1 – no winner | | Both seeker scores do not count on level2 | Fail | Add scoreIncrease variable to seekers movement functions. And return scoreIncrease += 150 if Snitch is caught. Then in main() function, when the movement methods are called, add the return (scoreIncrease) to the score1 or score2 variable |
| 23.Seeker1 catches Snitch on level1, seeker2 – on level 2 | End screen, seeker1 – 150 points,  Seeker2 – 150 points,  1:1 – no winner | |  | Pass |  |
| 24. The opposite: Seeker2 catches Snitch on level1, seeker1 – on level 2 | End screen, seeker1 – 150 points,  Seeker2 – 150 points,  1:1 – no winner | |  | Pass |  |
| 25.Restart the game from the End Screen | Whole game restarts  when *space* key is pressed. | | Nothing happens when *space* key is pressed | Fail | Call the main() method from the EndScreen if *space* key is pressed. Set all the variables to the default |
| 26. Restart the game from the End Screen | Whole game restarts  when *space* key is pressed. | | Game restarts, but to the second level, Snitch duplicates. | Fail | Add return statement in the EndScreen While loop to exit the function |
| 27. Restart the game from the End Screen | Whole game restarts  when *space* key is pressed. | |  | Pass |  |

**Test strategy**

Every time new features to the program were added, I have tested them. First of all, Field, Seeker and Snitch classes were created and tested. Then I started to work on my Main level. After adding new functionalities, I tested if they work correctly, documented if needed and only after that proceeded to the next step. This game was developed in prototypes. In my first prototype, both seekers were able to move and catch the Snitch and in the final one - the whole game was fully working. I have documented the errors that occurred in different prototypes of the game. The fully working prototype is shown in Test Plan and Log in the last 2 cases. Afterwards, I changed my design a little bit: the score and moves are on the top of the screen now and the End Screen looks different too.

Now the first level looks like that:

The second level:



And the End Screen:

