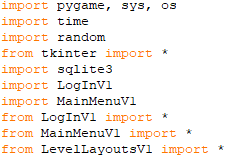
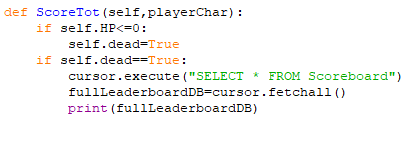
Development pt2



Made some changes to my imports and their orders so I can update the leaderboard after the boss dies.





Got it to print out the leaderboard to test that it’d give me the data needed to update the leaderboard.

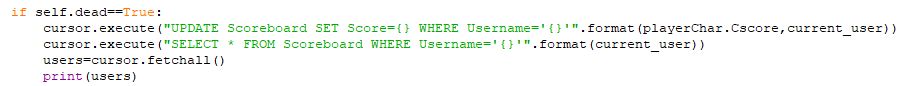


Gets the current user from the log-in code. This will display the player’s score after the boss is killed.





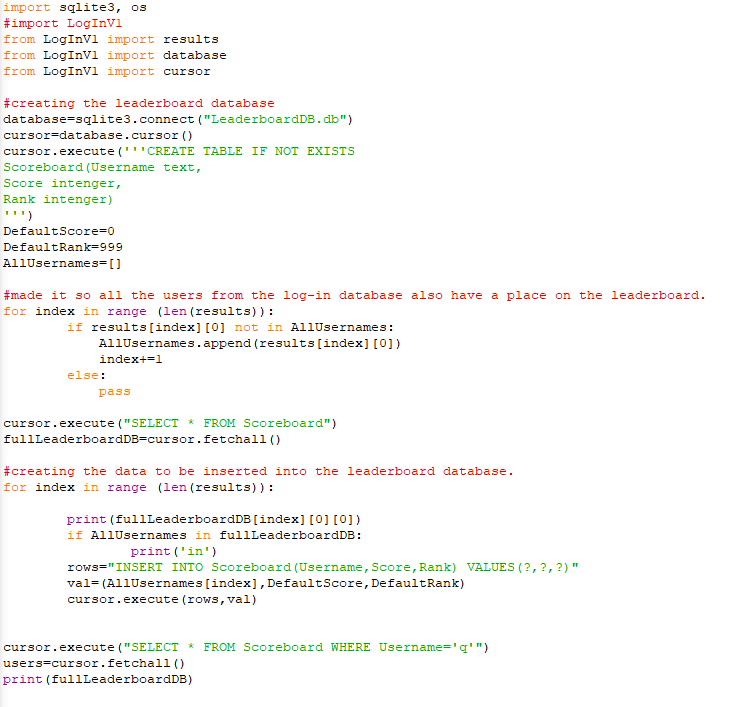
Added the extra instruction for the player in controls.



Database updates and replaces the previous score. There is an issue where this score will be overwritten back to 0 after the game closes.

After further testing my databases I’ve realized that the log-in database also doesn’t store the data it’s meant to and instead adds data to itself every time.

Got the log-in database into the same file as the leaderboard database, made it so they’re part of the same database, instead of 2 separate ones. Also, it seems that nothing will be inserted to the log-in table.



For convenience sake I put the database for the leaderboard in a separate file, I’m currently trying to get it to distinguish user that are already in the database from those who aren’t.



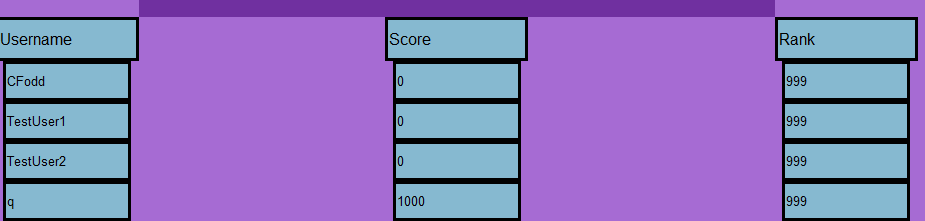
Noticed that the data in the database is wiped clean after the file stops running.





This is because I missed out these lines after I created the database. Now the data is repeatedly added to the database when the game is being ran again.



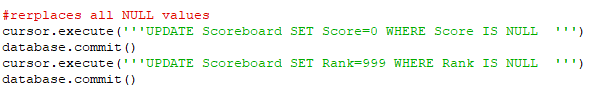


I added the code wherever the database needed to be edited or data was needed from it and database.commit and close at the end of the files. Now that I’ve done this the database now updates the player’s score. The issue, however, is that even if the score is lower then the recorded score in the database then it will still replace the score.

Data in the log-in table now stores, forgot to commit after inputting the data there.

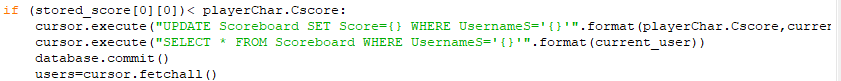


The scoreboard table now receives the entire field of data from the log-in table’s usernames. The issue with this is that the user’s score and rank are stored null.





If any values are null they are set to the default 0 and 999 instead.



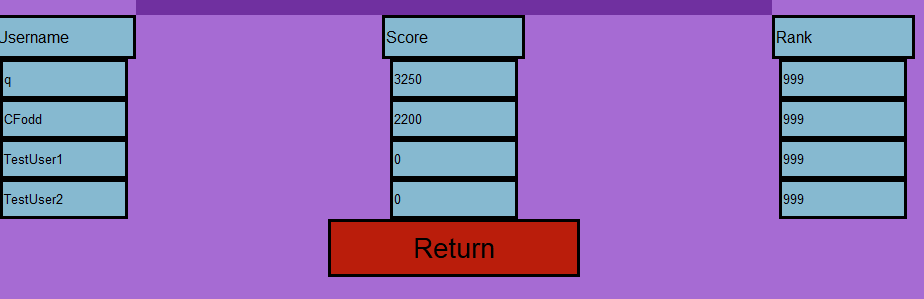




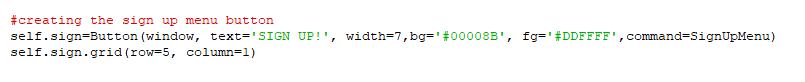


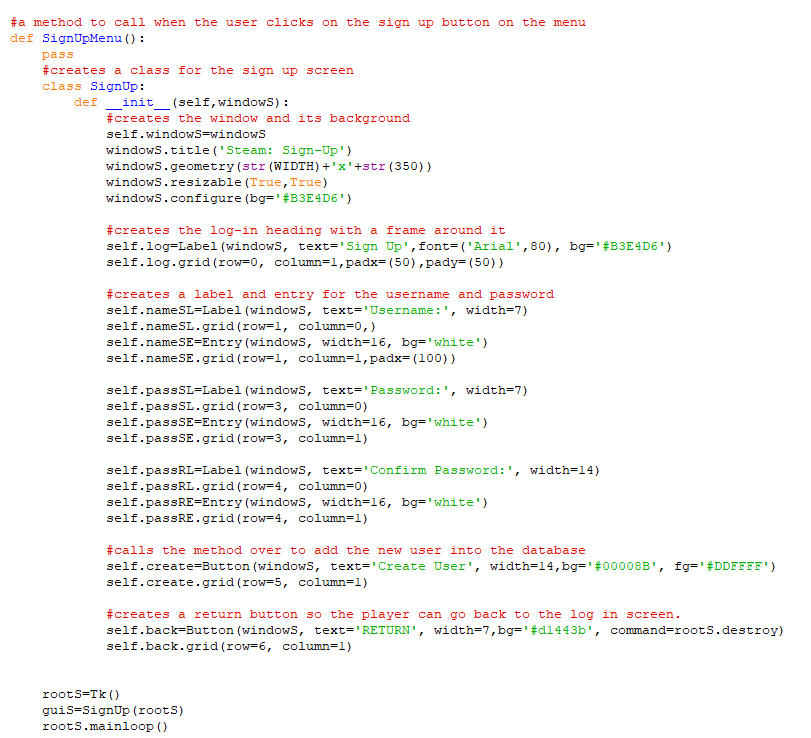
If the new score is greater than the old one then the leaderboard updates.





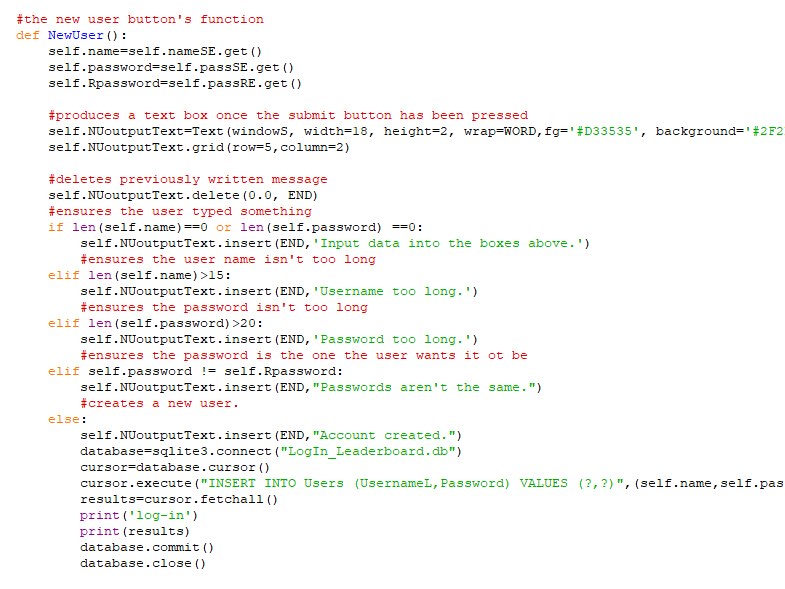
Th leaderboard is now ordered from greatest score to the lowest.







A new button on the log in screen now exists opening a sign up menu that doesn’t do anything yet.



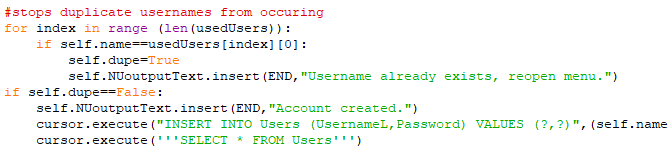
Let’s the user create a new user name. It seems like the new users won’t actually be able to log-in due to an error with checking if they’re in the database, even though they have been added to it.



I seemed to accidently delete the line selecting all from users to check the database. Now new users can access the game.

New users can log-in but they have to close the program and then reopen the game to access their account. To fix this I simply displayed the message restarting software and the game should crash when the user starts it without a log in so they’re forced to restart and now they can enter their new user account.

I still have to fix the issue of the leaderboard always inserting the log-in’s username field because it causes the leaderboard to have multiple records of the same user.



Fixed multiple username issue by creating a check for duplication.



Removed the line where the username field from the log-in table is insert into the scoreboard’s username field and instead set a default score and rank for when the user first creates their account.