

Introduction to Programming CS-111

Fall 2023

Sudoku Game

#الشعب\_الصيني\_ما\_له\_حل

Patrick Boules 23-101185

Marwan Khaled 23-101296

Youssef Ashoush 23-101049

Ahmed Yasser 23-101168

Introduction to the game:

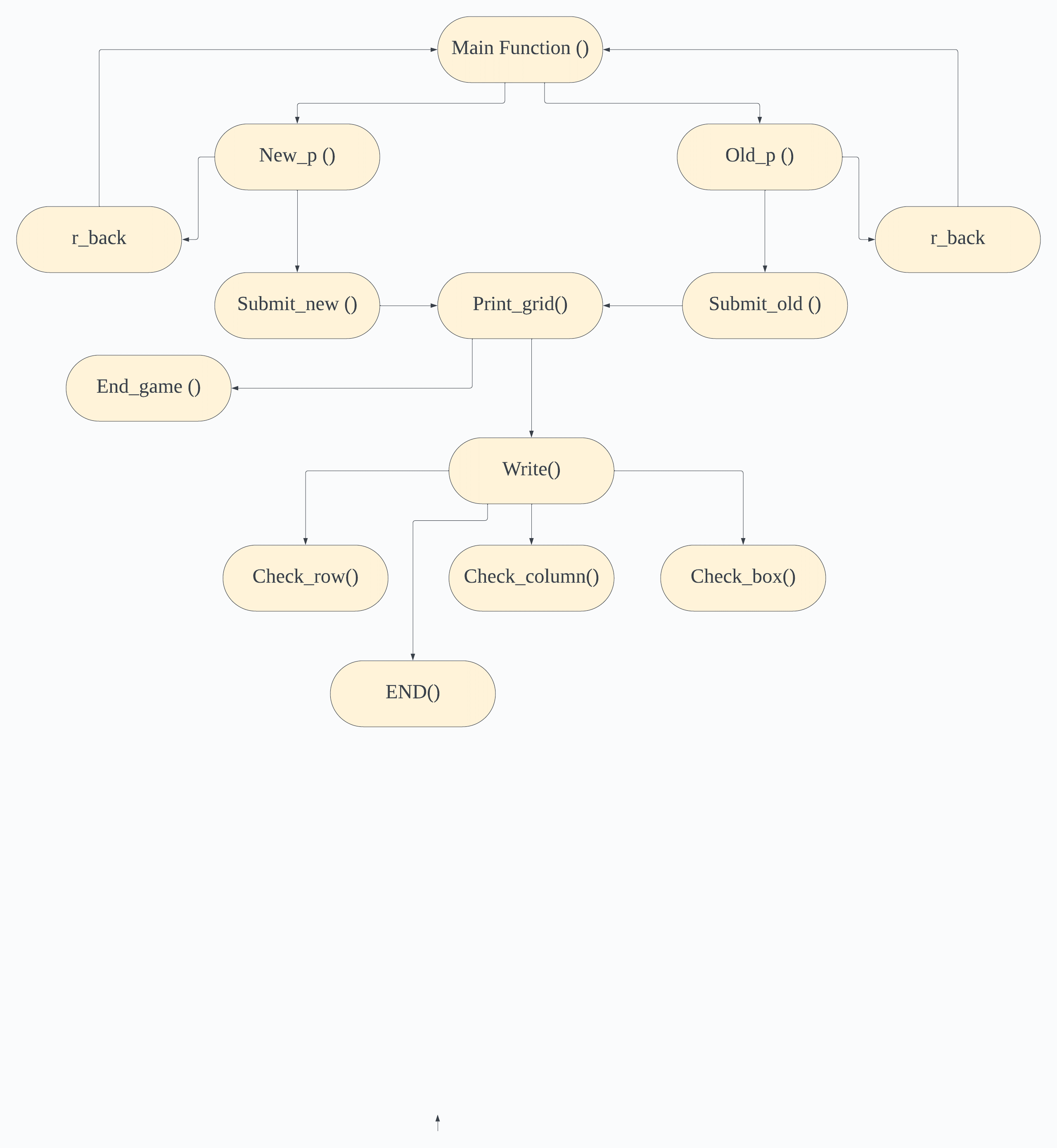
Sudoku is classical game that we all enjoyed while reading the newspaper in the morning. We decided to use the knowledge acquired during this course to recreate the classical game but using the modern techniques of python. Sudoku is game that requires you to fill all the boxes with numbers between 1 to 9, but there are some restrictions while playing. These restrictions are that a number shall not be repeated along the same row, same column, and with each 3 by 3 box.

Motivation:

After viewing the lists of games, we found that Sudoku will be the best project to test the knowledge that we gained throughout the course and push us to the next level. Although Sudoku is a game that doesn’t have multiple functions or features, yet the logic and math behind was the case that made us choose such project. Additionally, we wanted to learn how to play Sudoku.

Design:

1. Inside the Main function the program starts with displaying widgets such as the main label and the buttons of the main menu.
2. Then the program asks the user whether they are a new player or an old player.
3. Depending on the user’s choice different functions will be called to the program.
4. If the user is a new player a function that submits the player and prints the grid of the Sudoku. Else if the user is a old player a function that displays the saved progress of the player on the grid.
5. The print grid function takes a Sudoku list (whether a old or new player) and display buttons with the values of list. Some of the buttons will be disabled because these buttons are the main Sudoku list that can’t be manipulated.
6. Buttons that aren’t in the main Sudoku have a command that calls a function “write”.
7. The function calls 3 inner functions that checks if then number entered is in the row or column or box. If any of these functions returns “False” the number won’t be written in the box.
8. At the end, each time write is called it checks if there is no “zeros” left in the list. If its true and no zeros are left, the grid will be destroyed and a congratulations message will be displayed.
9. We have also included a “Quit & Save” button that enables the user to close the game and save the progress.



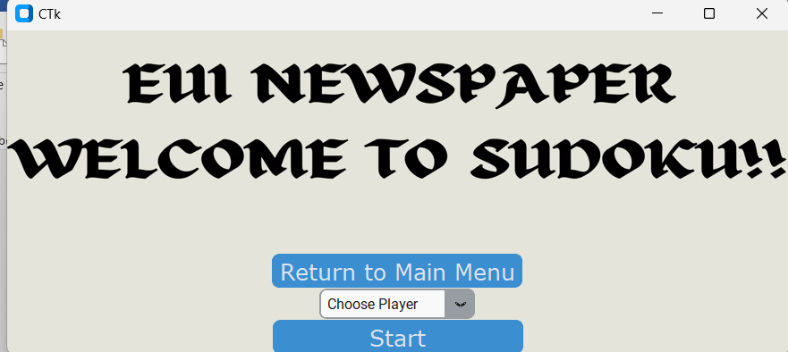
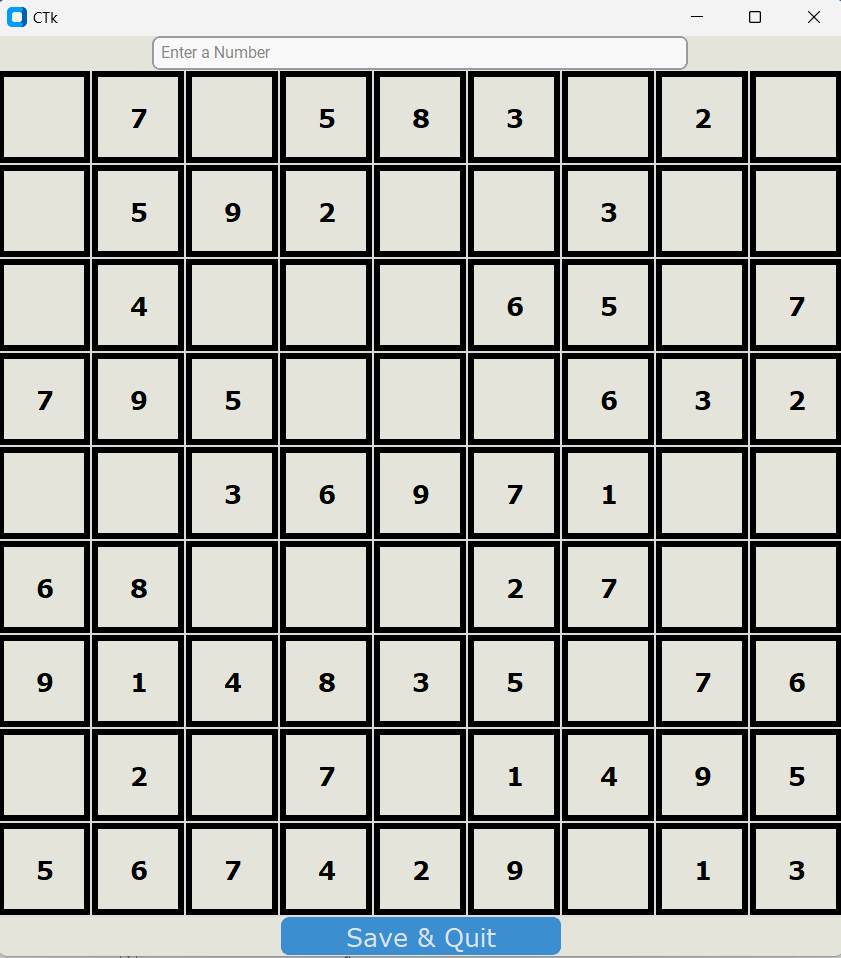
Description:

* The user is offered to choose between 2 buttons whether being a new player or an old player either way the game won’t start unless there is an input.
* If the player is new then it checks if the username is already taken or not, then assigns the sudoku list to the player. Then destroys useless widgets
* If the player is old then it loads the user’s previous progress from a file and assigns it to it.
* The print grid function compares the list assigned to the user with the main sudoku list to identify which buttons to disabled and which get the write function, then stored in a dictionary where the button is the key and the number in the sudoku list as the value.
* The write function checks each row, column, and box
* The check row functions receive two 2D lists one for the keys and one for the values. It checks if the number entered is in which rows and makes a list of the values. These values are whether the 0 or number. Then checks for the position of the buttons and returns the value corresponding to the button’s position. For example, if I click a button in the 5th row then I will get the 5th value in the results list.
* Similarly for the check column, the difference is that used list comprehension to turn the columns into rows.
* For the check boxes the same steps are done, we used 4 for loops the outer most loop iterates over 3 rows each time, the 2 outer loop iterates over 3 columns each time, the 2 inner loops are loops that access a the 2D list.
* At the end of the write function, it checks that values of the buttons dictionary don’t contain any zeros. If indeed there are no zeros left it display congratulations to the user.
* For the save progress, it takes the user’s name and append it to the Players file. Then creates a file using the username and writes in it the all the values of the dictionary in a form of a list.

Workload Distribution:

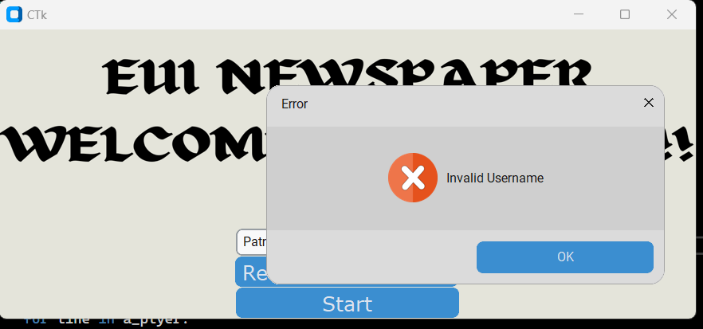
|  |  |
| --- | --- |
| Patrick | GUI presentation, functions |
| Youssef | First part of Print grid functions, Checkrow |
| Ahmed | All the files, first part of check column |
| Marwan | First part of check box, GUI widgets |

User Manual:

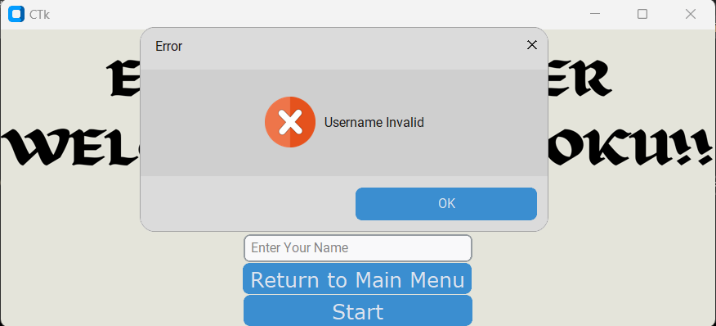
1. Click whether you are a new player or an old player.
2. If you are new enter your name.
3. If you are old choose your name.
4. Then press start, press the start after entering or choosing a name else it won’t work.
5. The start button won’t work if the new players have a similar name to an old player.
6. You can always return to the main menu
7. When the grid is printed the enter a number in the entry bar above the grid, the entry bar won’t accept any numbers above 9 or below1, it also won’t accept any strings.
8. Automatically after completing the boxes a congratulations message will be displayed.
9. The save and quit button saves your progress so you can continue later.

Test Cases and Validations:

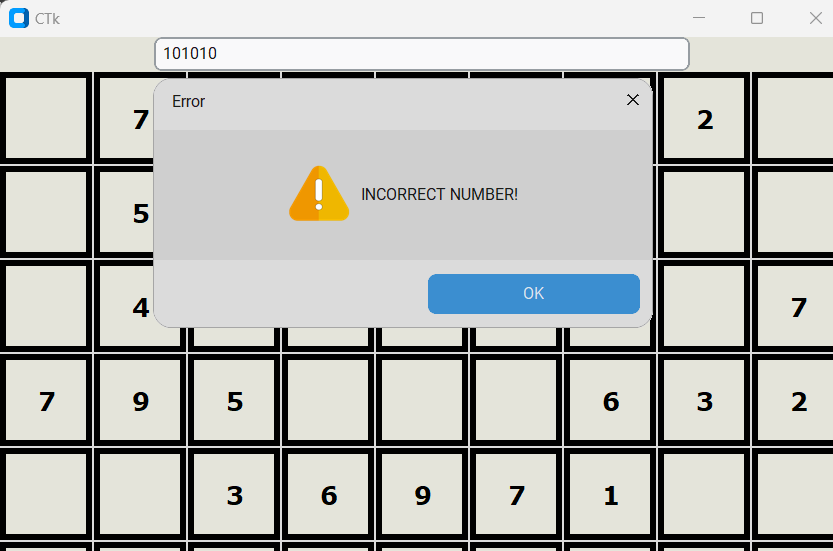
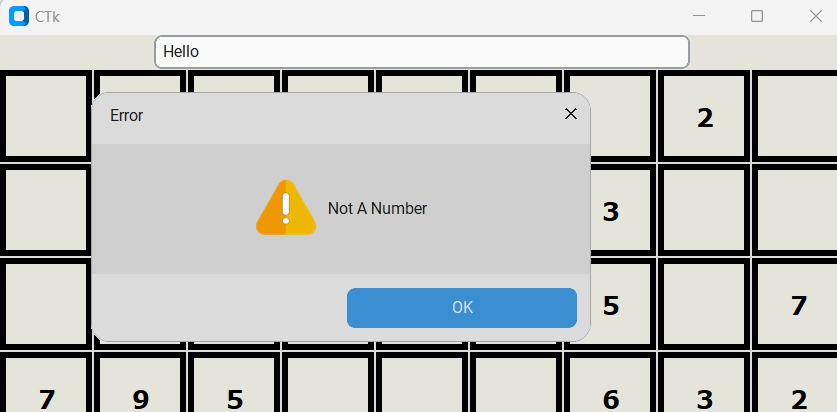
* Firstly, we made sure that the game doesn’t start unless there is a username given to the program.
* Then we made sure that no username is duplicated



* Then we made sure that name must be provided.



* We made sure that the user only enters a number and a number between 1 & 9.



Challenges and Conclusions:

* Challenges that faced us in week one was how to make each button has a unique id and easy to access. Thankfully, because of python’s data structure the dictionary, we were able to store the buttons and values in one variable easily.
* In the second week, it was challenging for us to check the dictionary because dictionaries don’t have indices, we thought it was best to check the values alone and change the button clicked alone.
* In the third week, we added the files and made sure that all the variables are read and written correctly.

Challenges and Conclusions:

* <https://youtu.be/YXPyB4XeYLA?si=jAyjUpxt_uqRorcP>

This is a 5 hour video about tkinter by FreeCodeCamp