MS3: User Manual
Arushi Aggarwal (aa2555), Grace Wei (gtw25), Petros Georgiou (pag238)

Below is a detailed set of instructions to install any packages to play our MS3 of Mastermind, and the general steps to properly use our program. We'll first go over how to install graphics onto our computer; then, we'll explain the rules and instructions of playing our version of Mastermind.

1. Installing OCaml Graphics

For the project, we used OCaml Graphics to create the GUI that allows the user to interact with the program.
To install Ocaml Graphics,

1. In the terminal, run [`opam update`] to update existing package repositories, then run [`opam upgrade`] to apply any updates.
2. Run the command [`opam install graphics`].
   a. Some systems don't have terminals that support graphical user interfaces; specifically, VSCode and the default Mac OS terminal do not. This command may prompt you to install a terminal/display server that supports graphics if you do not have one (on mac, this will likely be xQuartz – if an error pops up during this process, you may need to manually download xQuartz here). On Windows, the default terminal may suffice (unfortunately none of the authors have a windows computer to test this), or possible alternatives are puTTY or VcXSrv.

2. Running the Program

The program must be accessed through the installed display server in order for the GUI to open. Open a terminal on the software. As an example, on XQuartz, open a terminal by opening the app from the Applications folder. It might not pop up at first; you may need to open it as such:
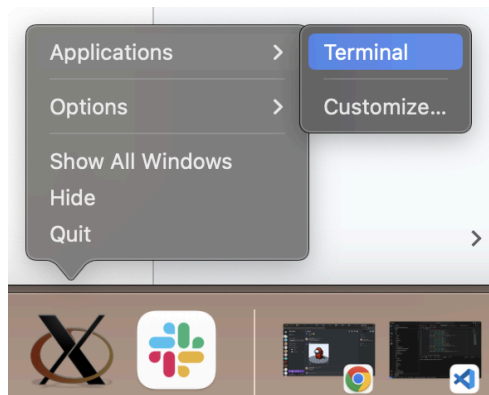

This new terminal behaves exactly like the standard terminal. For the sake of consistency and ease of use, we recommend running all programs here, including any dune commands.

This project is a Dune project; to run the program, run the commands [`dune build`] then [`dune exec bin/main.exe`].

3. Instructions for the Game

We'll explain how to run the game here. Note that this GUI is key based, and not based on any mouse events. This means all actions must be performed via typing on the keyboard.

After you've run [`dune exec bin/main.exe`] a screen will pop up with really pretty followers which is the title screen of the game. Note that at any point you can press 'q' on any of the screens to quit the game (on the top right of all the screens).
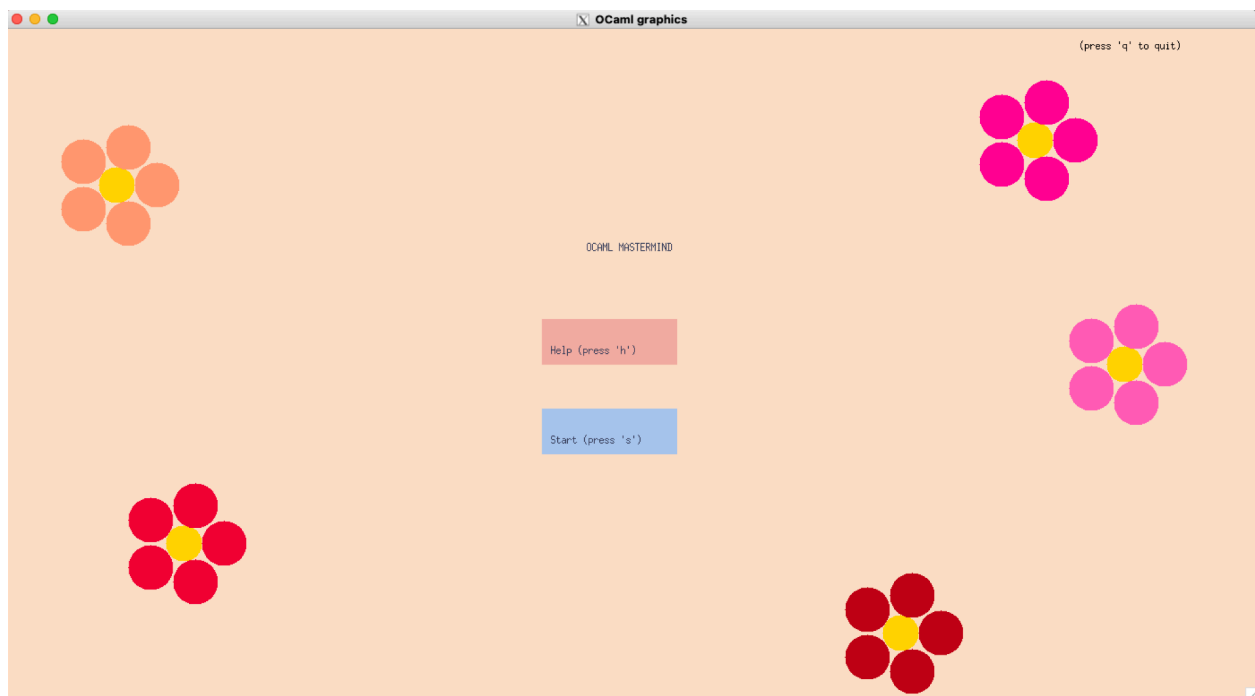

At this point you have two options which is either 'h' which will bring you to a help screen that explains the rules of the game. You can go back to the home screen by pressing 's'.

Back to Home (press 's')

Rules of OCaml Mastermind

1. Press keys to advance through the game. The keys shown on the buttons correspond to keys that need to be pressed.

2. The player will decide in advance whether they want to play as the codemaker or the codebreaker.

3. If they are the codemaker, the computer becomes the codebreaker, and vice versa.

4. The codemaker chooses a pattern of four code pegs to be the answer.

5. The codebreaker tries to guess the pattern, in both order and color, within eight to twelve turns.

6. Each guess is made by typing a sequence of four numbers.

7. No duplicates are allowed in neither the answer nor the guesses.

8. Once placed, the codemaker provides feedback by entering feedback (using the digits r, w, n in that order), which will show up as pegs in the small holes of the row with the guess.

9. A colored key peg is placed for each code peg from the guess which is correct in both color and position.

10. A white key peg indicates a code peg that belongs in the solution, but is incorrectly positioned.

11. Once feedback is provided, another guess is made; guesses and feedback continue to alternate until either the codebreaker guesses correctly, or all rows on the decoding board are full.

12. If the codebreaker guessed the code, they win, otherwise the codemaker wins.

Once you go back to the main screen, you start playing the game by pressing 's'. At this point you'll be prompted to be either player 1 or player 2; by pressing '1', the user assumes the role of the role of the code maker where the user will make the code and the computer will guess. In player 2 the computer will make the code and the user will guess.

Select your player:
(Player 1 means you start as the codemaker)

Player 1 (press '1')   Player 2 (press '2')
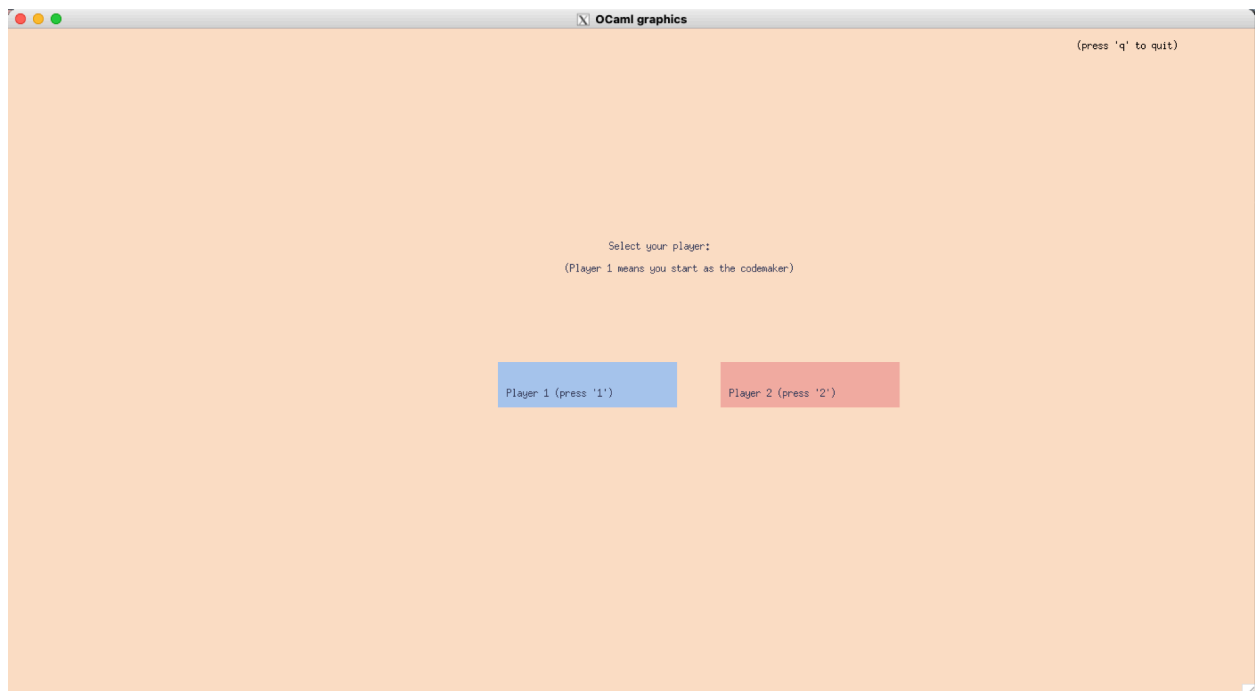
Before we go into the differences of the player 1 vs player 2 screen and paths, a screen shows up to announce what computer algorithm you will be playing against and you can press 'p' to continue or if you're scared you'll lose you can press 'q' :)
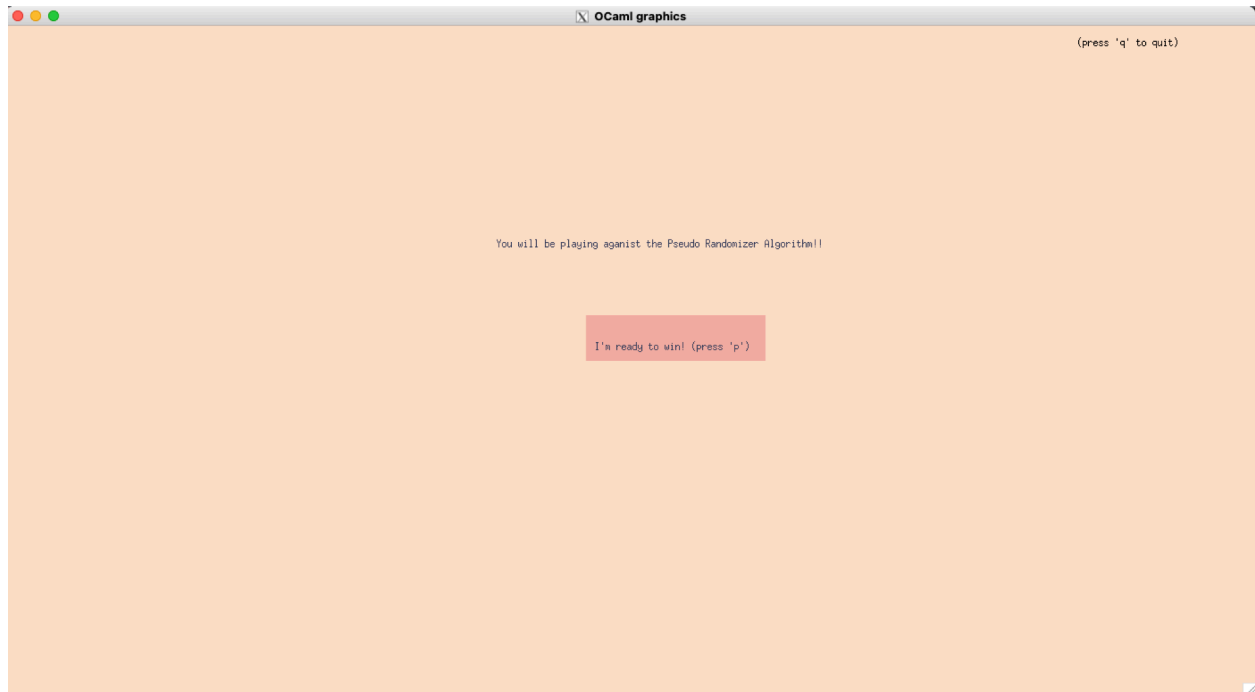
Player 1 Gameplay:

If you selected player 1, you will be prompted to enter a 4 digit code. The code entering process has validation checks such that it can only be numbered from 1 to 6 inclusive, 4 digits and no duplicates. This is an interactive feature you can just start typing you code in and as soon as you hit 4 digits you will be automatically taken to the next screen.

Enter your code (4 digits, no duplicates, digits 1 to 6 only):
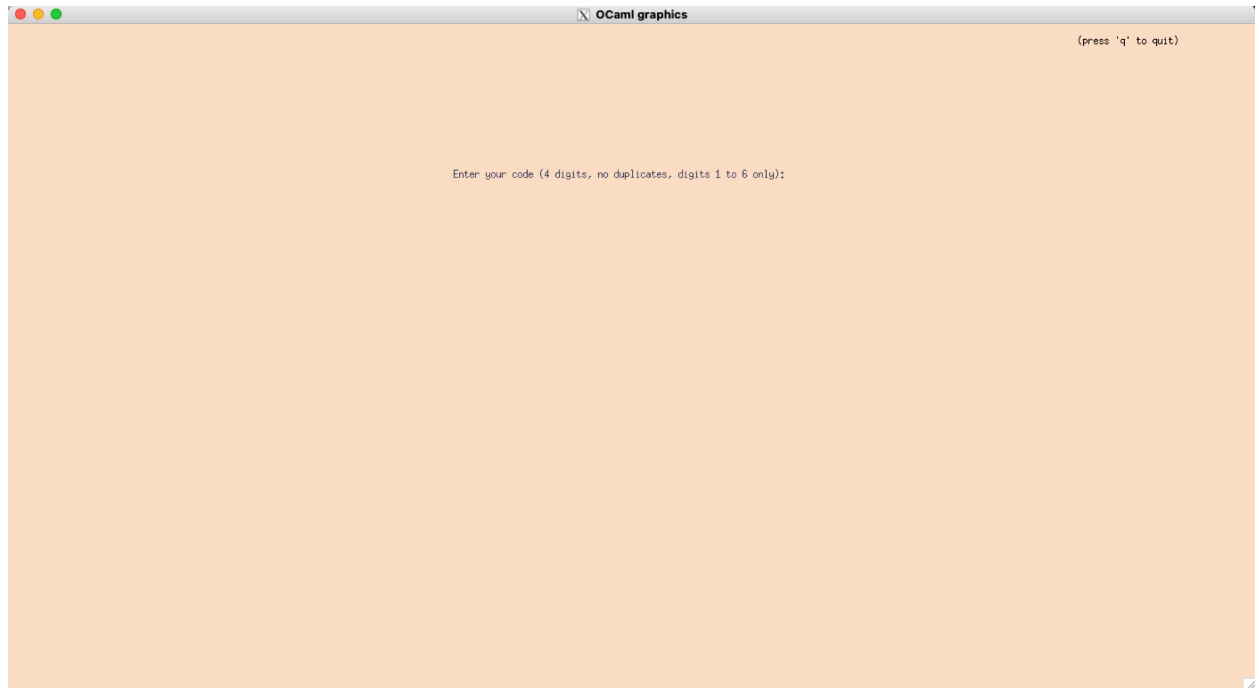
From here, the computer automatically inputs the first guess based on the algorithm. At this point you have to give feedback in the following format where

• 'r' indicates a Red pin, where a color is in the correct spot with respect to the answer.

• 'w' indicates a White pin, where a color is not in the correct spot, but in the answer.

• 'n' indicates a Null pin, where a color isn't in the answer.

We always establish that the feedback must be given in order of Red, then White, then Null and inputted as lowercase letters as shown above.

For instance, consider the code is (1, 2, 3, 4) and the computer guesses (4, 2, 5, 6). Then the pin feedback would be r, w, n, n. Notice that Red appears in the first spot, although that corresponds to the second index. If the user does not following these rules of properly writing feedback, we warn them that it was incorrect and match it to the correct feedback. We output the feedback and the game continues on to which the computer has 12 tries to get the right feedback.
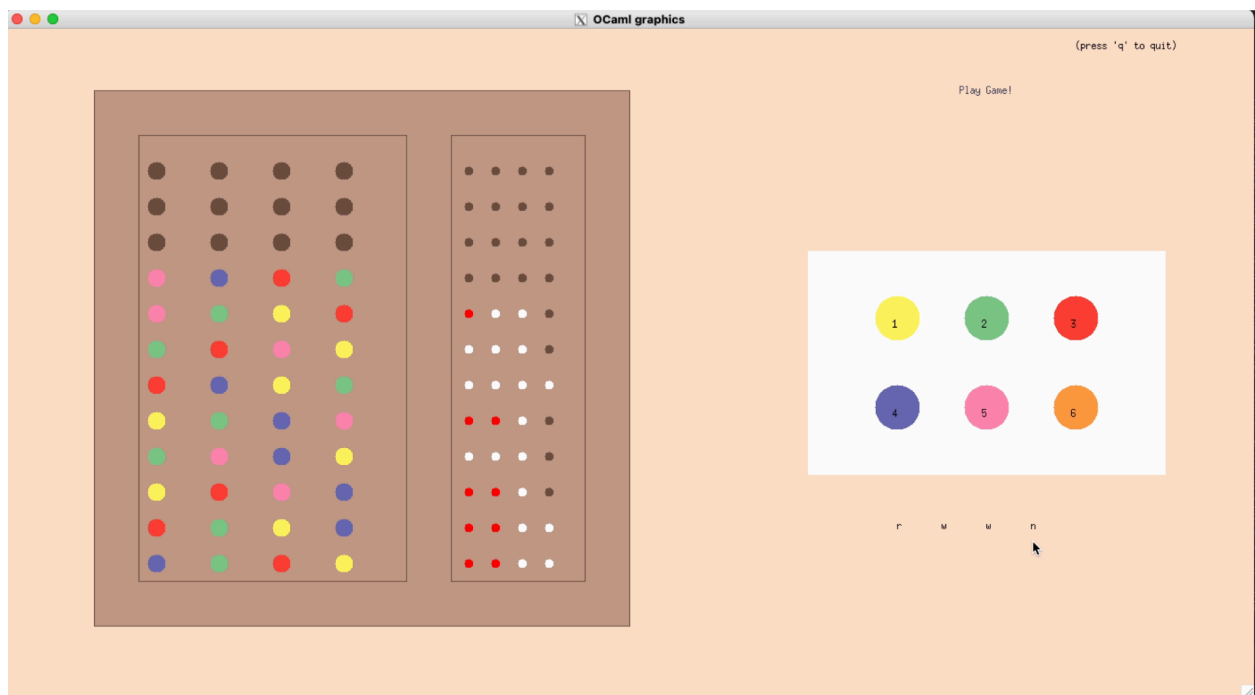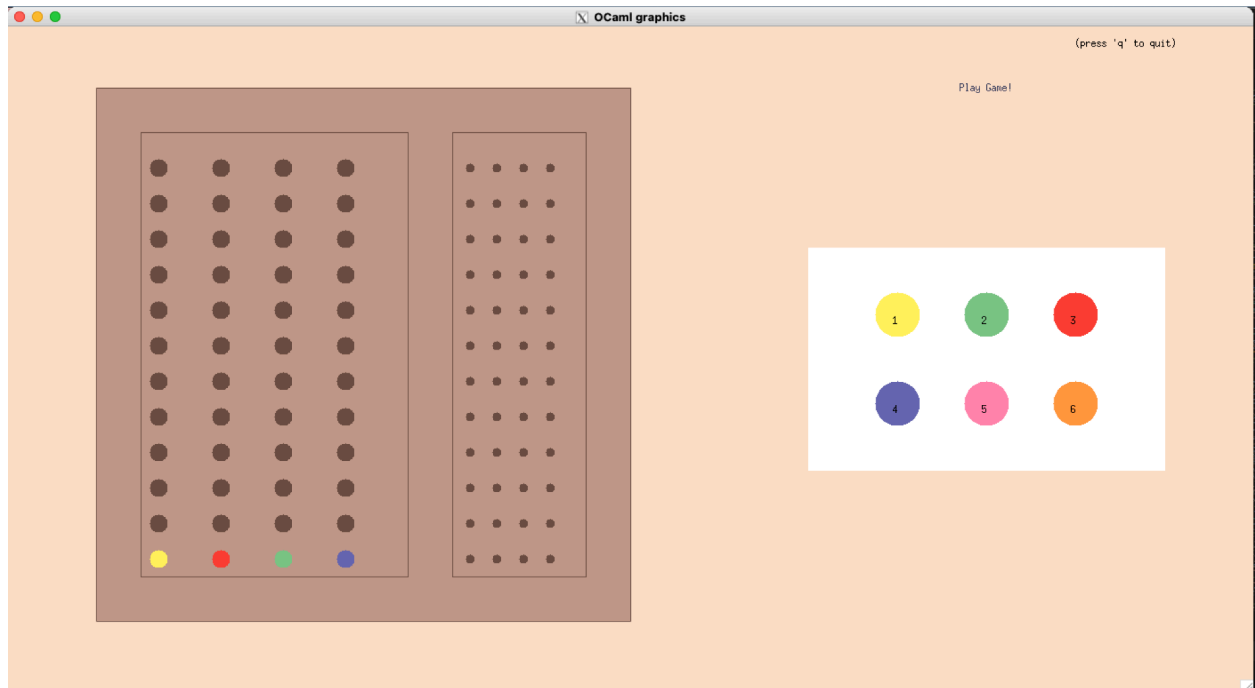
This is an example of how the game runs with an example input feedback (you don't have to press enter or anything it takes in the feedback automatically). The game ends when the board is full (the computer has taken its 12 tries) and we have a winner. Before going into the win screen, we will move onto the other play which is when the player guesses the code.

Player 2 Gameplay:

If player 2 was selected, the algorithm screen will also show. Press 'p' to indicate to play. From here, you will not be prompted to enter a 4-digit code. The computer randomly generates a code. The same rules apply to the player to the computer. They must guess codes based on the feedback to get closer results to the answer.

The player has 12 tries. Guesses should be inputted based on the corresponding numbers, so type in 4 digits to submit a guess (you do not need to . Typed numbers will appear on the screen under the input board.
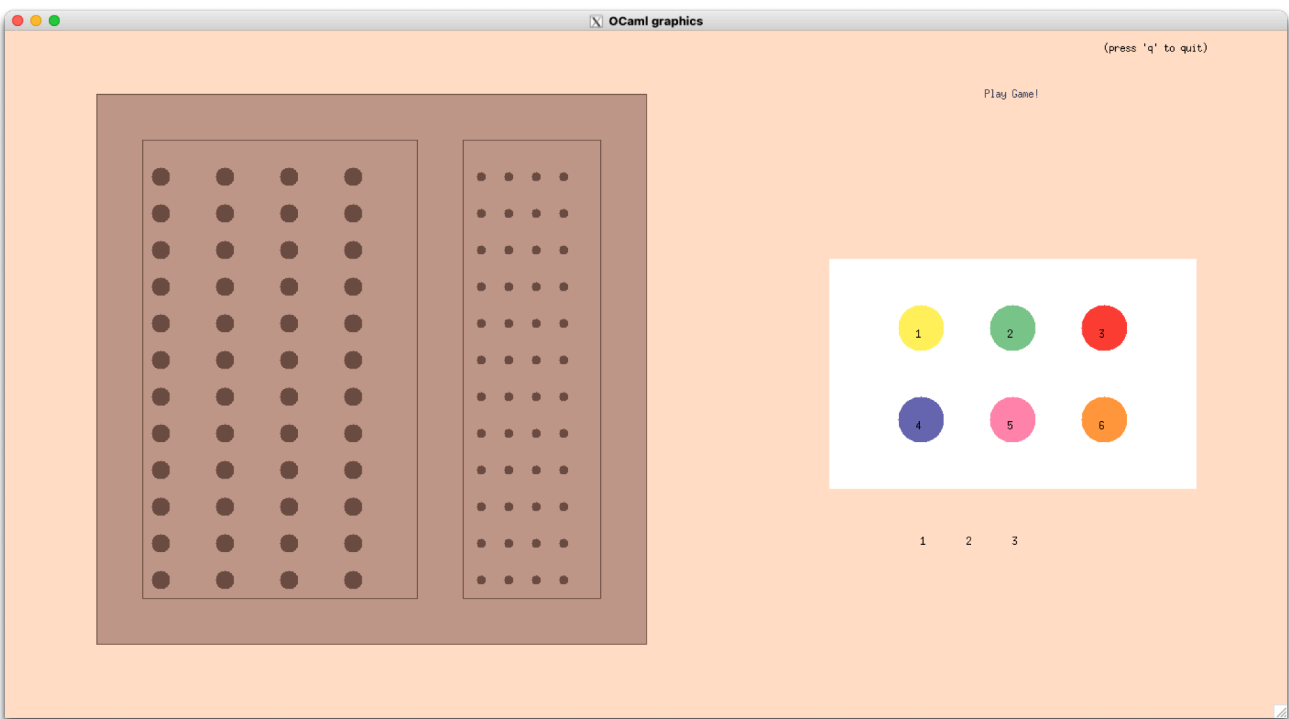

After inputting a guess, the board will populate with the guess and the feedback calculated by the computer. For example, if the user inputted [1 2 3 4] as the first guess:
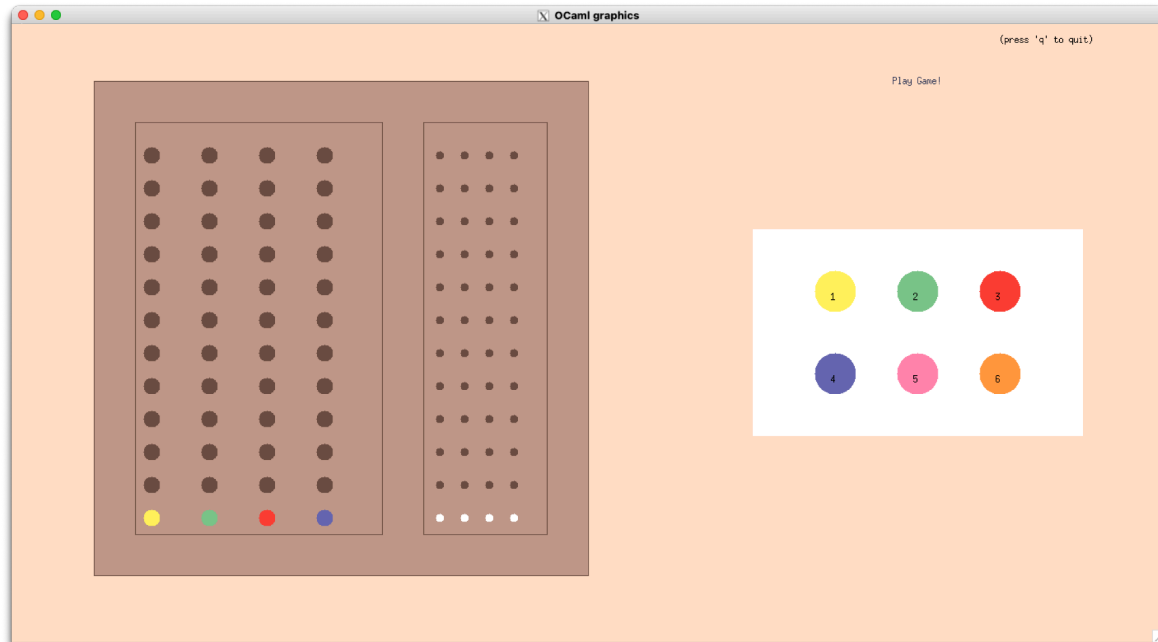
Since the feedback is all white, we know that the four digits are all correct, but none are in the right spot. The game ends when the user guesses correctly or runs out of tries. There are also small motivational messages that may appear after guessing.

Ending and Rerunning

After the win conditions have been met from either of the two plays above, it will print a win screen which shows who won, the computer or the player and if you want to play another game you can press 'm' to go back to the menu or 'q' to quit.
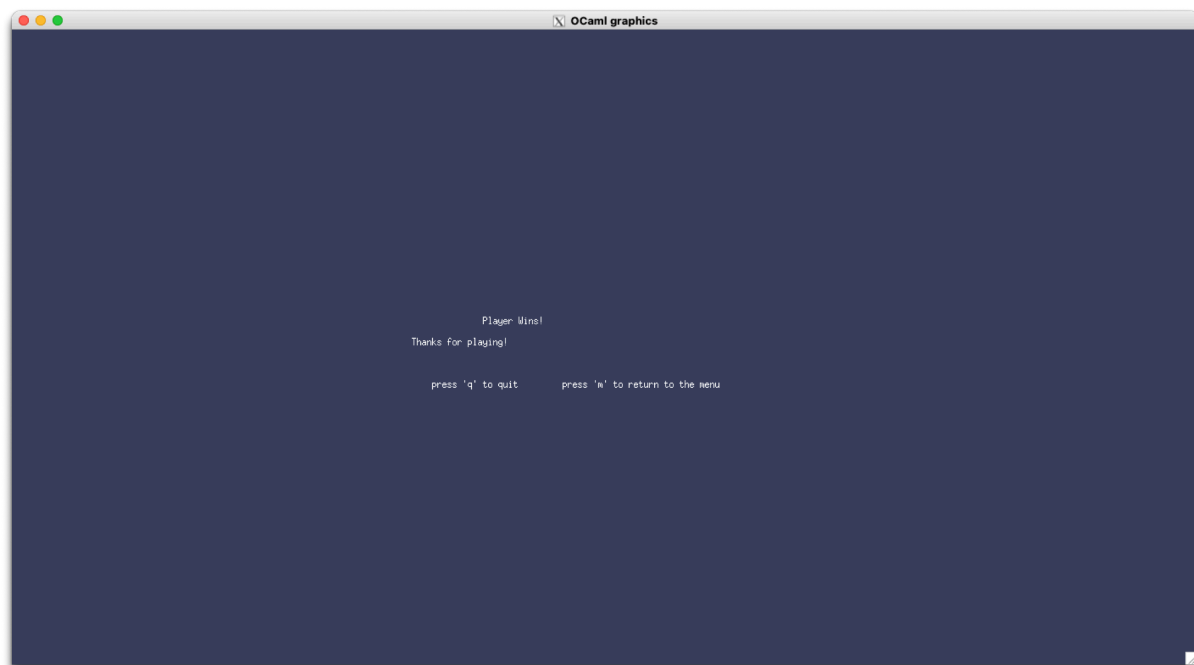
An important rule on typing: anytime where the user inputs a key not previously prompted, the screen will not do anything. On the players side, while making a code or breaking a code, adding duplicates will be prompted with a red warning that it cannot be done. Also, due to the refresh rate of OCaml Graphics and our program, overloading the system by rapidly spamming guesses or feedback may cause unintended behavior. "The two most powerful warriors are patience and time" - Leo Tolstoy, *War and Peace*.

I hope you enjoyed our game and have fun! :)