IS1200
Tegar Björkdahl 20031014-6758
Henrietta Gidehag 20030612-4348
2024-02-23

Pong: THE Game

Final abstract

Objective and Requirements

In this project, an old arcade-like game of Pong from 1972 was created. In the game, a ping pong ball moves back and forth across the screen, and each player attempts to return the ball to the opponent using their own respective paddle. Failure to return the ball, results in the opposing player gaining a point, and the game continues. If a player accumulates 10 points, they win the game, and the score resets.
The intricacies of the project did not exceed the complexities of a advanced project and thus the game implemented the following basic requirements:

- Simulating the game on an OLED graphical display.
- Allowing the ball to move in both the X and Y directions.
- Enabling the paddles to move solely in the Y direction.
- Ensuring that the ball bounces naturally on the paddles, as well as on the top and bottom sides of the screen.
- Implementing functionality to detect and display when a player wins.

Solution

The group created a Pong game specifically for the ChipKIT Uno32 board in conjunction with the Basic I/O shield. Player control is made through the utilization of the 4 buttons on the ChipKIT. Each player is then able to use two buttons each to maneuver their respective paddles, allowing for upward and downward movements, shown on the graphical display. The entire development process was carried out using MCB32tools, and the programming language employed for coding was C.

Verification

The techniques used in this project aimed to assess the game's performance to confirm its functionality. Through systematic testing of various in-game scenarios, the goal to eliminate potential edge cases in the code, ultimately improved the overall gaming experience. For instance, examining situations where the ball hits the corner of the screen was one such scenario. Another example was when a player continued to hold down a button after a player won. This resulted in the game resetting and starting rapidly. Before both players might have been ready. After a function in the code was done, the group tested it in the terminal to find out if it did what it was supposed to do and if it had any errors that could be modified. Different parameter values

were also tested to see what impact it had on the overall experience. Such as, size of play area, paddles, and scoreboard.

Contribution

During the programming process, the tasks were not partitioned among team members with regard to bigger cases, for instance displaying all elements on the board and ensuring good interaction of the ball with its surroundings. The work primarily used collaborative efforts, wherein the code could be discussed back and forth iteratively.

Instead the coding was divided between the members in functions and/or methods, so the members could easily help each other out if some part of the code did not work the way it should. This approach helped the group members get a better understanding of all the parts of the code. A majority of the time, both group members sat with the same problem and worked together to find a possible implementation of code that would solve the issue.

Reflections

In summary, the project went well throughout almost the whole process. At the beginning it was somewhat difficult to get started due to the lack of instructions that was handed to us. To start from scratch with something that neither of the group members had any experience in, resulted in a slow start. To understand and use the display on the chip kit was challenging but once we got the hang of it, the project ran smoothly. Due to the fact that we began the project promptly after lab three was finished, we did not feel the need to stress the process and had, in fact, time to spare in the end.