Pong Game

By Philip Salqvist (1992-04-17-1376)

2021-02-07

Objective & Requirement

The project objective is to develop a pong game on a CHIPKIT Uno32 together with an CHIPKIT Basic I/O shield. The game is going to be rendered on the display embedded in the I/O board and the interaction from users is done using the pushbuttons on the the same board. The game consists of two players represented by a pong paddle each and a ball that bounces between them as well as the top and bottom limit of the screen. Player 1 is going to be positioned on the left and Player 2 on the right side of the screen. If the ball travels past the left outer limit of the screens x coordinate, player 1 gains a point, and vice versa for player 2 concerning the right outer limit of the screen. First player to gain a defined number of points, wins the game. Due to the fact that I am currently studying at a pace of 150%, I won't attempt an advanced project. The following are the basic requirements that must be met:

- Multiplayer mode
- Clearly describe visually how to start the game
- Score visible and updated while playing
- User's can control their respective paddles on the Y-axis using push buttons
- When the ball passes a player's defended limit, the other player gains 1 point
- When a player reaches a defined number of points, the game stops
- Clearly state visually who wins on the display when game is over

If time allows for it:

- Simple AI to play against
  - If this requirement is implemented, the multiplayer requirement is no longer obligatory

Solution

The game is going to be implemented on a CHIPKIT Uno32 together with an CHIPKIT Basic I/O shield. The input is going to be generated using the shields push buttons and the output is going to be displayed on it's display. The paddles are going to be moveable by the players using two push buttons each, one representing upwards direction and one to move in the downwards direction. When a player is asked to start the game, the player will be able to do so using any of the push buttons. The same goes for restarting a new game. The built in timer will update the screen as well as set the speed of the game. Concerning the development MCB32 will be used and code will be written in C.

Verification

The project will be tested using the specific requirements stated earlier in this abstract. The tests will first and foremost focus on corner cases such as what happens when other buttons are pushed, when buttons are pushed at the same time, when the ball hits the horisontal walls of the screen as well as the vertical walls on the screen. The tests will make sure that points are given to each player correctly, that the score is updated correctly and that the game stops and displays the correct content when the game is over.

Contributions

Due to the fact that I have a lot of courses this semester, I have decided to work alone to ease the planning and scheduling of the project.

Reflections

Reflections will be added to the final draft of the extended abstract.