

EE/CS 10b

Balance Game Functional Specification

Zachary Pestrikov

Description

The balance game will use the AVR CPU to interface with a circuit board that will have a sequence of LED's that represent a ball. As the orientation of the circuit board changes, the LED's will shift on and off based on how a ball would roll as it would be pulled around the board by gravity. The goal of the user will be to get the orientation of the board to light the last LED on each side consecutively without allowing the 'ball' to roll off the board. This means that the user will have to orient the board such that the 'ball' rolls to one edge of the board without rolling off, and then orient the board directly opposite so that the 'ball' rolls to the opposite edge of the board without rolling off either. The game will start whenever the user presses the begin button on the board. If the user is able to accomplish this task, a victory sound will play and the program will exit. Alternatively, if the 'ball' rolls off the board, then the user will lose instantly, and the program will exit. The user can also restart the game by pressing the start button while the game is running, or by just turning the board upside down.

The user will also have the ability to interact with the settings of the game, such as the weight of the ball or the size of the ball, which is represented by the number of LEDs turned on at a given moment. The game will keep track of the user's high score(the amount of time to win) while the game is running, and there will be an option to reset the high score.

Global Variables

The only global variable will be the user's high score.

Inputs

The inputs to the program will include the orientation of the board, the begin and exit buttons, as well as the adjustable settings for the weight and size of the ball.

Outputs

The outputs of the program will include the LEDs that will light up to signify the position of the ball and the audio, which will play depending on whether the user wins or loses.

User Interface

The user interface will include all 70 of the LEDs, three buttons, and a rotary encoder. The buttons will be for starting/exiting the game, changing the size of the ball(with the second button), and changing the weight of the ball(with the rotary encoder and third button). The high score can be reset by restarting the game. The final score of each game will be displayed on the 7-segment display, followed by the high score. If the user accomplishes a new high score, audio will play.

Error Handling

The only relevant error is that the board is flipped greater than 90 degrees or upside down, in which case the game will exit and play a losing noise as if the user lost the game.

Algorithms

The only relevant algorithms will be to determine the location of the 'ball' based on the data from the IMU. Every time the program receives data from the IMU, the program will compute the location of the ball, where the ball will be due to gravity, and the corresponding LED's that represent that location.

Data Structures

The only relevant data to store will be the settings and location of the 'ball', as well as the high score. It will also be necessary to store whether or not the user has already gotten to the edge for the first time during the game, and only needs to get to the edge one more time.

Limitations

The only limitation is the lack of more ways to interface with the program.

Known Bugs

There are no known bugs at this point.

Special Notes

There are no special notes at this point.