Snake game

By Joakim Sundman (020406-1139) and Julius Larsson (901028-4470) 2022-02-07

Objective and Requirements:
We are gonna make the classic snake game on the ChipKIT Uno32 board together with the Basic I/O shield using the programming language C and MCB32tools. The goal of the game is to score as high as possible or until the screen is full of the snake and the snake can't move anymore. The player loses if the snake hits a wall or itself. It is a basic project.
The requirements that must be met is:

- Display game on build-in OLED graphical display.
- Snake moves across the screen in X and Y directions.
- Snake chases apples, which, when eaten, increases the length of the snake.
- Snake can collide with itself and walls.
- Detect when snake collides, and display the number of apples eaten (score).

Solution:
Our intention is to develop our project on the ChipKIT Uno32 board together with the Basic I/O shield. We will use the small display on the I/O shield as the screen for the game. We are gonna try to use the push buttons as input for the snake or if we have time maybe use the sliders as well for controlling the snake. We are gonna work using the programming language C and MCB32tools provided from the labs.

Verification:
To verify the program we are gonna test the program with a lot of different tests. We are gonna do the testing in phases, where we have gotten a part working as intended and test different worst case scenarios for example when the snake hits itself by the player or giving too much input or when the snake is getting into corners with nowhere to go. We are also gonna do these tests with the apples for example with spawning and where the apples can spawn. In the final abstract we are gonna have all the test cases listed with more information on what is tested and why and if any crossover with the test happens.

Contributions:
We are gonna divide the work into two parts, one is gonna be making the snake work and being able to move around on the map. The other part is the map and apples spawning in the map then combining those two parts to have the final game. Joakim will work on the snake and Julius will work on the map with the apples. We are both gonna work on the general program logic and the development of the graphical display. The final report will have how we divided up the project more specifically and in more detail.

Reflections:
Reflections are gonna be made in the final abstract when we have done the project and we will discuss how it went and what happened.