Doodle Jump (game)

By:

Ruxandra-Stefania Tudose (20030413-0206) and Oleksandra Bilovus (20040704-T003)

2023-03-02

1. Objective and Requirements

Our project consists of a game called Doodle Jump. Having said that, the main character of the project is Doodle. The goal of the game is to catch as many coins as possible by using Doodle's skills, namely the fact that it can jump and move to the right. The game has three different levels, which differ in the positioning of the coins, while the LEDs light up in the binary system the player's score (the number of coins the person has caught).

First and foremost, Doodle Jump requires only one player. As far as the **hardware components** are concerned, one ChipKIT Uno32 is needed that has to be connected with one ChipKIT Basic I/O. On the ChipKIT, we use the OLED graphical display, the LEDs and the buttons. Then, as far as **software** is concerned, MCB32 tools, C programming language, Assembly and Makefile have been used. Our game has a circular structure and that is to say, in order to restart the game, the player doesn't have to unplug the ChipKIT.

2. Solution

Our solution can be analyzed from two different perspectives. The core idea of the software we implemented is that there are two functions that find themselves in forever loops and therefore, we managed to obtain the circular structure of the game. Apart from this, there are other several functions we built which have their own unique purpose. From the hardware's perspective, the following have been implemented:

a) Pressing on BTN2 starts the game.

b) Pressing on BTN3 moves Doodle to the right and BTN4 makes it jump (we, therefore, obtain movement on both the Y-axis and the X-axis).

c) Levels are not chosen, they just automatically start after the previous one has been finished (no buttons need to be pressed).

e) When level 2 is reached, pressing on BTN4 apart from jumping, also brings the coins closer to Doodle: the coins that are up will come down and vice versa.

f) Level 2 is the last level and therefore, the game ends.

g) Pressing on BTN2 will restart the game without unplugging the ChipKIT.

3. **Verification**

The strategy we adopted in order to make sure that the game works correctly was that we always made one change at a time and immediately tested it on the ChipKIT. After we finished, we split it into different key features (i.e jump) and tried them one by one. In the end, we played the game from the beginning up until the end.

4. **Contributions**

Since we use the OLED graphical display, the development of the game started with common efforts in researching how the pixels are distributed and how we should go through them when updating the display. The most important conclusion was that we always work with a simulated two dimensional array. We update everything in a separate one dimensional array, which stores the columns which are made of eight pixels grouped together.

Then, simultaneously Oleksandra drew on paper the graphics for the game and Ruxandra built the logical template of the game where we then inserted the functions we both did.

5. **Reflections**

We are both extremely excited that we have managed to build from scratch our own vision of Doodle Jump. The most challenging part of the project was time management as we had to meet certain deadlines and at the same time researching about the OLED graphical display. On the other hand, since it has required a lot of effort, the moment we figured out how the pixels are distributed was by far, the most rewarding moment.