Kozmoz

A classic sci-fi arcade game

William Söderqvist (971226-8813) & Sebastian Lihammer (000121-1655) **2019-03-04**

Objective and Requirements

The goal of our mini-project was to develop a game to run on the ChipKIT Uno32 that would be inspired by the classic arcade game *Space Invaders*, developed by Tomohiro Nishikado in 1978. In *Space Invaders*, the player controls a cannon that can move from left to right alongside the bottom of the screen and shoot at incoming waves of enemies coming from above. We wanted our project to be an “advanced” project and as such we also wanted to implement additional functions, listed below;

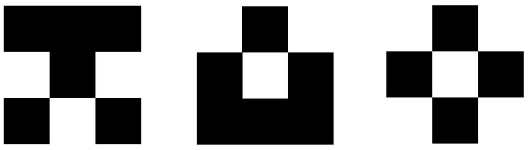
Implemented features

- ❖ Usage of the PIC32 platform.
- ❖ Have the game run on the Basic I/O shield using the OLED graphics display.
- ❖ *Space Invaders*-esque gameplay (e.g. player and enemy movement, projectiles etc.).
- ❖ Fluid movement of objects across the screen.
- ❖ Singleplayer and Multiplayer (cooperative) mode.
- ❖ High Score-system and High Score-list.
- ❖ Implementation of external components - joysticks

Solution

The project was developed on the ChipKIT Uno32 board together with the Basic I/O shield. The small display on the Basic I/O shield is used to display the game. All development was done using the MCB32 tools and all of the code was written in the C language. External components were implemented for in-game movement along the X-axis (joysticks). The game itself functions as a while loop depending on a variable *enemyHit* which once it hits a value of 30 (e.g. all enemies have been killed) signals that victory has been achieved, repeating the game.

The refresh rate (e.g. speed) of the game is dependent on an implemented delay function so that the game isn’t running in the maximum speed of the ChipKIT (in which it would be nigh unplayable). Enemy movement speed is regulated through a counter. The system of a random enemy firing back at the player was implemented through using two timers (TMR2 and TMR4), running at different speeds, to generate a more or less random number, which was then used to determine which enemy would fire at any one given point in time. This system is used for each row of enemies individually, meaning that there can be a maximum of three shots (one per row) fired at a time. The green LEDs of the ChipKIT are used to represent the health of the player (8 to 0 healthpoints), decreasing with each successful hit by an enemy.



Example graphics: enemy designs (3x3 pixels)

The screen-covering graphics used in the game (the start screen, the menu, the "game over" screen) were designed in MS Paint and then converted to a bitmap code that could be sent to the OLED display. This was done using a web service (http://www.majer.ch/lcd/adf_bitmap.php). Other graphics (the player ships, the bullets and the enemies) were implemented using a function to light up specific pixels on the screen. The name of our game, *Kozmoz*, derives from the word "kozmos", Albanian for "space".

Verification

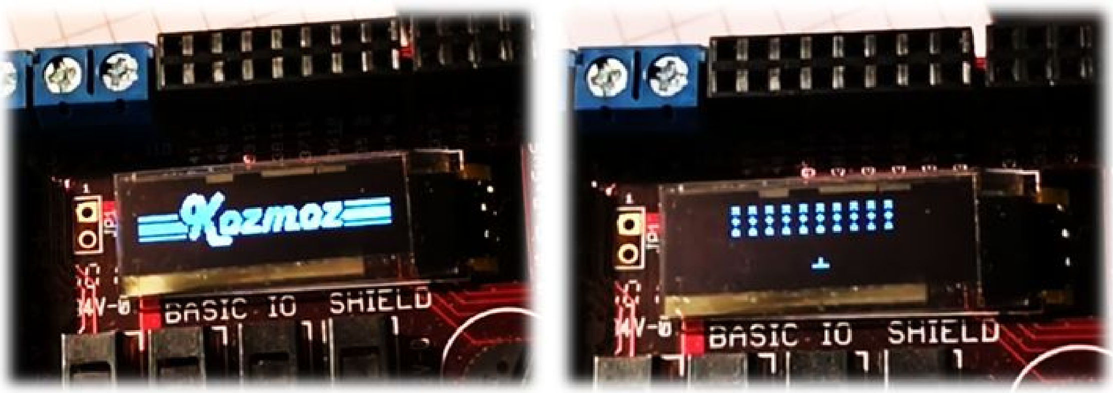
The program was verified through thorough testing, both by us and by other people. Different scenarios, including most importantly regular gameplay (and everything that entails) but also unexpected cases, such as unorthodox gameplay strategies, were tested through playing the final working game. The final and most important verification is presenting and showcasing the final, working, game at the Expo.

Contributions

We have cooperated on the design of the game and research into how we would be able to implement the functionality we wanted. The coding was physically done on only one of our computers, since we encountered difficulties in attempting to creating a file we could both edit in real-time, but we have contributed with code, ideas and solutions together.

Reflections

Through the making of our game we have significantly developed our skills in programming and in interpreting and understanding manuals. Though the majority of our work was initially based on the files used in Lab 3 of the course, we have also been forced to create new functions using variables previously unknown to us, such as *struct*. All in all, this mini-project has been a great learning experience, especially in regards to project management and C-programming. We've uploaded our full code to GitHub, where it can be read and downloaded (https://github.com/Sebbmeister/Kozmoz).



Game start screen and example gameplay