

Basic IO shield resourced Space game

By Murat Eksi (970218-4897) and Justin Arieltan (990401-2359)

2018-02-07

- **Objective and Requirements:** The purpose of this project is to make a space game that is inspired by "Space Impact" on the old Nokia platforms and optimize it well enough to run as smooth as possible on the Uno32 embedded system. We are aiming for an advanced project.

The main *must* requirements for the game are as follows:

- a. The game must support high score lists and different difficulty levels.
- b. The orientation of the same type of objects such as enemy ships or the projectiles must be implemented.
- c. The player must be able to control the main ship using the buttons for X-axis and shooting, while using the potentiometer for Y-axis.
- d. The sprites of the game must be able to fluidly move throughout the screen, pixel by pixel.
- e. Hitbox for the objects must be implemented
- f. Particular things such as the health of player must be outputted in some way using the basic IO shield (led, screen and etc..)

Things to be implemented when the time is available:

- a. Boss levels depending on the high score
 - b. More sophisticated AI for the enemy ships
 - c. Power-ups for the player such as shield or a new weapon
- **Solution:** We are going to develop our game for Uno32 and the basic IO shield of it. We will use the display on the basic IO shield to output the game for the player to interact with. The buttons on the basic IO shield will be used for X-axis movement and shooting while the potentiometer will be used for Y-axis movement. We are thinking about outputting the health and the high score to the player using the led system on the basic IO shield and also intending to implement the pause mode and other possible modes using the Switch slides on the basic IO shield. The difficulty will be a dynamic factor that depends on the current score which can result in more ships or game speed changes as we are thinking about having a rogue style of a game. We have implemented our system of pixel drawing for the objects in order print them to the screen as smooth as possible. Refresh rate of the game and the speed of it mostly depends on a delay function we have implemented. We also had a problem with the random number generator of the standard library in C, so we implemented our own random number generator for our project using the included timer in the system. Most of our project is written using the C, while some of it is Assembly for interruptions and it is also compiled using MCB32tools.

- **Verification:** We are planning to verify our code and logic by having edge cases and by basically trying to break the game as we are playing it in a more bug hunting fashion. The game will be served to our friends as we are playing it to hear about their opinions and to also catch the bugs that we might have not come across with, more like a beta tester basically. We are planning to have an edge case scenario tests to play around with.
- **Contributions:** We are planning to work individually at first to gather up as much information we can and then try to find the best ways of implementations for our project together by discussing about what have found so far. We intend to tackle most of it together by focusing at one thing at a time. We will be explaining the final division of our project work in the final report.
- **Reflections:** Even though it is the beginning of our project, we believe that we already learned a lot by just researching and playing around with the Uno32. We will be reflecting on what happened in the project with more details in the final report.