

Mars Lander - A simple lunar lander game

By Edvard All (911019-0437) and David Ljunggren (940101-4734)

2019-12-09

1 Objective and Requirements

We wanted to make a game with gameplay similar to that of the classic vector game based on carefully controlling a lunar lander in order to land safely on a surface.

Collision If the spacecraft reaches the surface at less than a given speed, the player has successfully landed. In multiplayer mode, a rudimentary routine handles collision between players.

Thruster Fuel Fuel is a scarce resource that if depleted renders the player unable to use thrusters.

Randomized Stages Terrain is generated randomly.

2P-mode Two players are able to simultaneously play the game in a best of five. Whoever lands with the most amount of fuel left is the winner.

High score list The highest ranked single player mode players are saved in a list.

Increasing difficulty As a solo player progresses through stages, fuel and (programmatically ensured) platforms get more scarce.

2 Solution

We developed the project on the ChipKIT Uno32 board with the Basic I/O shield only. We made use of the small display on the Basic I/O shield to display the game state, and the LED lights to display fuel (as OLED display space was a scarce). Slide switches were employed for menu navigation and the push buttons for in-game controls. Specifically, in single player BTN3

and 4 are used for right and left thruster (both being upwards thrust), respectively, whereas in multiplayer also BTN1 and BTN2 are used by the other player. We used one of the built-in timer interrupts to take care of in-game speed. We linked the, in the FAQ provided for, standard library to implement pseudo-random generation of terrain. We settled on seeding with the aforementioned timer as this seemed satisfactory for our use-case. All the development was done using the MCB32tools and all code is written in C.

3 Verification

Bugs were discovered by playtesting, throughout the project. Possible edge-cases, mostly focusing on drawing pixels on the screen, were reflected upon and iteratively considered by testing them throughout the implementation of features.

4 Contributions

The workload of this project was not divided up formally. Both participants were doing a bit of everything. Game design was done through conversation, and implementation through pair programming. Most work was done with both group members being present. Some features, such as the high-score list and pause feature, were implemented by Edvard All.

5 Reflections

If more time was given for the project, a larger external display would most likely have resulted in a more enjoyable game. Due to the small OLED display on the I/O Shield some features deemed hard to implement, such as consistent fuel display in multiplayer mode. More input options would also have made for a better experience. A third button per player would have been more optimal, which would mean a total of six buttons instead of four.

In a conventional Lunar Lander game, the game controls allow for the ship to rotate. This lets the player land the spacecraft on planes not perpendicular to the spacecraft. Rotation was however deemed too difficult to implement, and it's not clear how this would be represented on the screen. This made us reconsider the actual aim of the game. If we would have kept the ground as a flat surface and instead introduced obstacles, we feared the game would end up like a dull Tetris hybrid.