Space Escapers

Christopher Mckerracher
1000617718

Jacob Reinertson
1000362109

Game Rules:

The game is a simple clone of the game “Space Invaders”

The idea of the game is to kill the Mothership UFO (M) before any of the UFOS(A) escape into the earth’s atmosphere.

You control a spaceship(B) that shoots lasers to kill these UFOS. Every time you kill a UFO, the game updates your score. If you gameover, the game displays a “Game over” message, and resets your score. If you win, you get a “You win” message, the game restarts, and you maintain your score. You keep playing the game to see how high of a score you can get before it “game over”s.

Code:

The code is written entirely in assembly.

The code at the start inits all the array and memory values, so that every memory value is the same at a game reset(come game over, or game win).

The main loop, called LOOP, handles keyboard inputs, and updates the user's lasers, or location on screen.

We have a timer that, at an interrupt, changes context between LOOP and the "game loop". In our interrupt handler, we have a context switcher to switch these two contexts.

The Game Loop and its related subroutines handle updating UFO's location, drawing the updated player location on screen, updating laser location, checking for a collision between laser and UFOs, checking for game win, and checking for game lose.

WEEK 1:

In week 1, we worked getting display to the screen via VGA and the game update cycle working. For VGA display, this entailed figuring out the peripheral logic for VGA display, including writing to and reading from the port. We decided on using the VGA port's ASCII display mode for our project, since it conveniently divided the screen into into a 60x80 grid, which we would have done with pixels anyways. The update cycle was coded by creating a timer interrupt that occurs on a fixed interval, allowing us to have a game logic update happen at a consistent period.

-Jacob handled:

-timer initialization

-drawing things on the vga

-Chris handled:

-memory allocation for the items drawn on screen

-interrupt code for the timer

WEEK 2:

In week 2, we began to implement the components of the game related directly to the player. This included controlling the player character, firing lasers, and having those lasers collide. For player control, we read in inputs from the PS/2 port, discarding the “key release” codes that are associated with a key press. We then applied the corresponding value to the player position based on these inputs. We then added lasers, which are lines that are defined to move down the screen. We gave those lasers the ability to collide with objects, setting a flag for scoring purposes if they did.

-Jacob handled:

- -keyboard code
- -UFO logic code

-Chris handled:

- -Laser code
- -collision logic with laser and UFO

WEEK 3:

In week 3 we added enemies, win/lose states, audio, and scoring. The enemies are rendered to the screen from an array, and move up the screen while strafing to the left and the right. When a player's laser collides with an enemy. the score is incremented up by 1. If the UFO that the player hits is the mothership, a win is registered, and if any UFO reaches the same Y plane as the player, it is game over. In terms of audio, we added sound effects to the player firing a laser and hitting a UFO.

-Jacob handled:

- -Audio code
- -Score keeping
- -Game over score reset logic, and game reset logic
- -Game win score keeping logic, and game reset logic

-Chris handled:

- -UFO movement logic code
- -Logic for game over
- -Logic for game win