SNEAKY PIXEL

A GAME ABOUT A PIXEL THAT DRIVES A SPACESHIP WHICH IS MADE FROM PIXELS

MADE BY LAVDIM IMERI AND ROBERT ALM

THE ESCAPE FROM THE PIXEL EMPIRE

In a galaxy far, far, away. The Evil ASCII Empire, (after the great Pixel Rebellion, decides to strike back, and in one decisive victory to destroy the Pixel Rebellion to bits and bytes. The left of the pixel army are forced to flee in order to escape, the Imperial ASCII Fleet.

At the same time. A lone heroic pixel., known as Sneaky – 2 – Pixel – 2 flies through a wormhole in a quest to unlock the secrets of the RGB in the lost city of Lorem Ipsum.

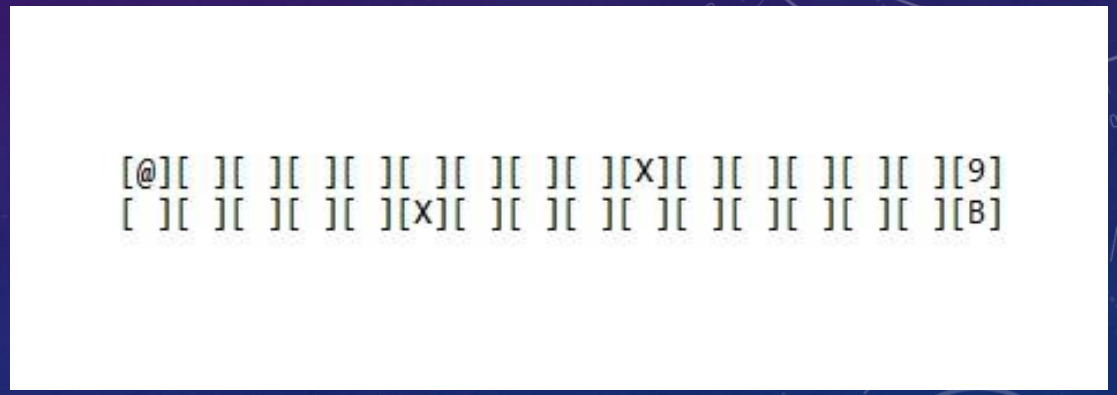
This game, follows Sneaky 2 Pixel 2 on its journey through the wormhole.

THE GAME IS MEANT TO BE LIKE FLAPPY BIRD

- Different pipes that spawn up and down, had to be made with different objects, and to run through a fairly complicated algorithm that 32k of memory probably would not be so happy to handle.
- The graphics also had to be complex as well, otherwise, the game would look like dots up and down, and few dots floating in the middle.
- The worst part was that we would have to make the game designed through individual pixels and that means a lot of work, that probably would not be able to be handled from Arduino.
- So, we had to design something simpler…

SO WE CAME WITH THAT

- The data for the stage of the game is just an array that holds integers from 1 to 2.
- The content of that array is rotated by 1 and copied to one other array that helps with the representation of the stage the specific moment.
- The position 0 of the second array is the step, which contains as value a specific integer.



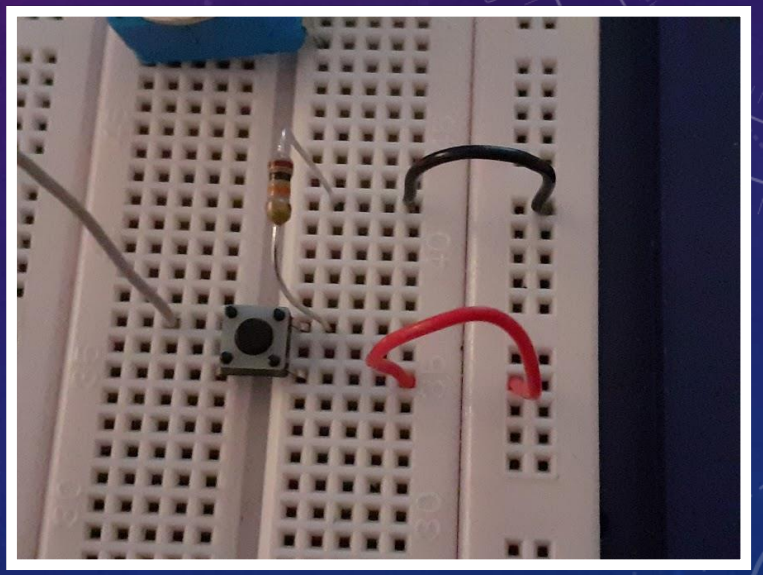
ACCELERATION

- The game has a nice acceleration model. The stage does not go one step on every loop, but it has a counter, (we call it a ticker), that increases on every loop, and we have a variable which is called delayer. When the spaceship accelerates, the delayer is decreased, when the spaceship is slowing down, the delayer increases again.
- Every time the ticker is equal the delayer, the spaceship moves by one, and ticker returns to zero.
- This little trick apart of giving the feature of speed to the game, unbinds the up/Down movement from the speed and the spaceship can move one up or down one every loop, regardless of the speed.

```
//updates the frame and gives the tempo
if(ticker < delayer){
  ticker++;
} else {
  if(stepInStage < 14){
  stepInStage++;
 } else {
  stepInStage = 0;
 }
 ticker = 0;
}
```

“INSERT COIN” AND “GAME OVER”

- The best use for the button and for the “attachinterrupt()” method was not a feature inside the game.
- Old arcade games used to ask for coins, in order to allow the player to play the game.
- When the player presses the button, he inserts a coin and he gains a life.
- When the player loses all his lives, the game turns to a “Game Over” screen and asks from the player to insert coin in order to play again.



JOYSTICK

- As expected, the spaceship is controlled by a joystick
- We didn’t have special cables for the joystick, and we didn’t want to bend the pins, so we inserted the joystick vertically to the board and it is very unstable.
- That means that this game is not for hardcore gamers
- That affects also the Y-axis of the joystick as we had to reverse the readings to make it work vertical, (up is down and down is up). The x-axis is not effected by the configuration.

THE SCREEN, (LCD)

- Our biggest challenge was the screen.
- We had problem with the first monitor (OLED) all the time.
- After many tries and replacing almost everything in the project we found out that the problem was in the dashboard which we replaced with a new one.
- We used eventually the LCD screen to implement the game.

THANK YOU!

- Creating that “masterpiece” it was a journey, (a bumpy one).

Lavdim Imeri

Robert Alm