

CSEE 4840

Embedded System Design

Jumpers

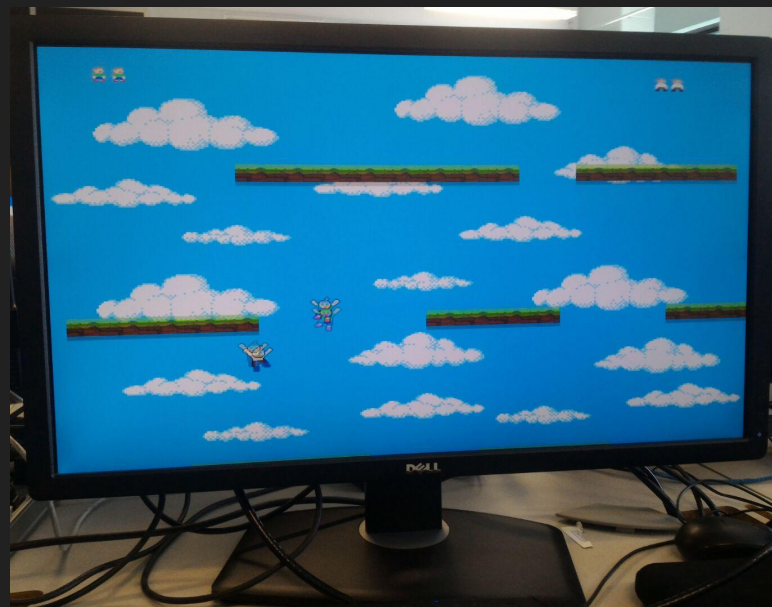
Bernardo de Almeida Abreu bd2440

Henrique Pizzol Grando hp2409

Lucas Ikenaga Barros li2176

Tomas Mantelato tm2779

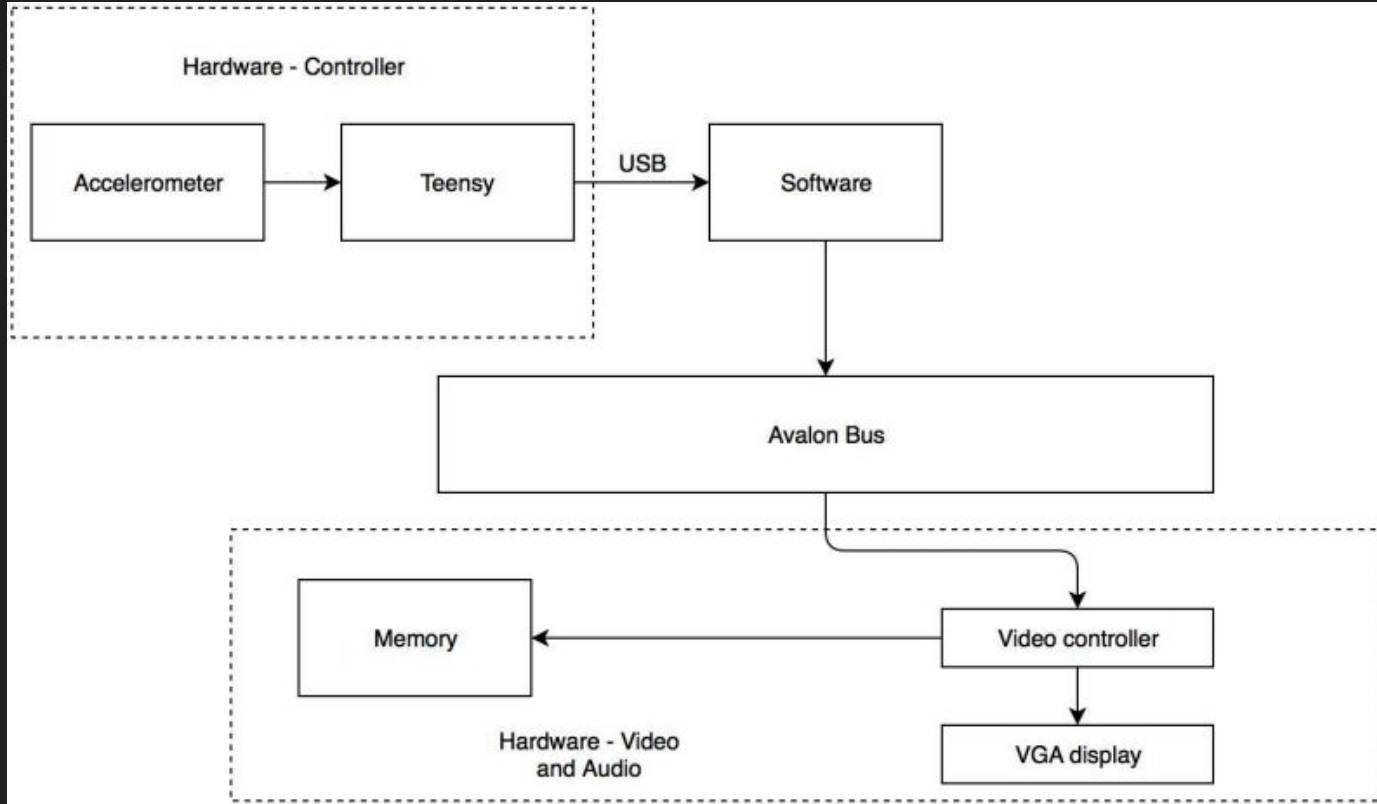
Professor Stephen Edwards



Overview and Objectives

- The objective of this project was to develop a game controlled by an accelerometer, in a FPGA board.
- The game is composed by two players that need to jump over platforms in order to stay alive.
- If a player reaches the bottom of the screen he loses the game
- The accelerometer will define the horizontal velocity in which the player moves around.
- The number of lives that a character has is pictured by the number of icons representing himself at the top of the screen

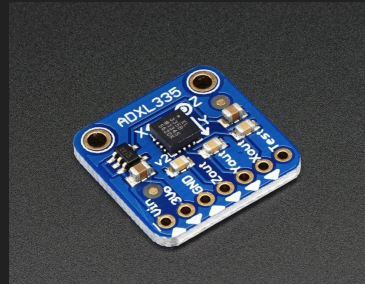
Design Diagram



Hardware Peripherals

Accelerometer:

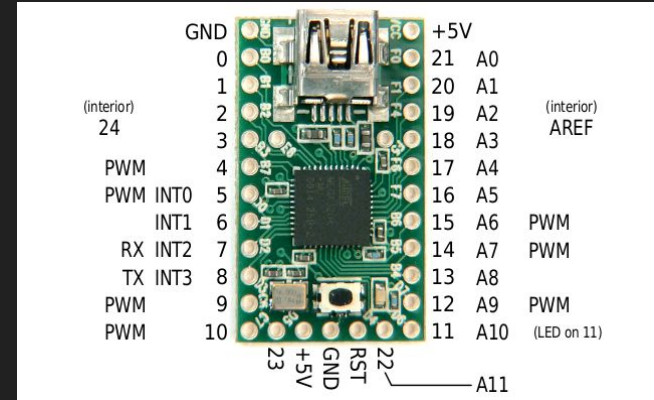
- ADXL335 (3 axis accelerometer)
- Only one of the axis controls the horizontal displacement of the character
- Connected to the Teensy 2.0 through analog pins



Hardware Peripherals

Teensy 2.0 (ATmega32u4 USB development board)

- Reads data from accelerometer and sends to the FPGA via USB
- Arduino compatible.

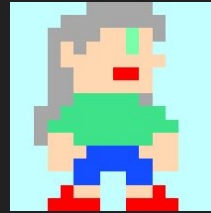


Why we used it? The teensy is a plug and play device that shows up as keyboard when connected. We could use the driver from Lab 2.

Hardware

Sprites

- The sprites created for this game are divided in icon sprites and the gameplay sprites
- The icon sprites have a size of 16x16 whereas all the other sprites have a size of 32x32
- There is a color palette of 19 colors
- The sprites can be mirrored through the hardware



Hardware

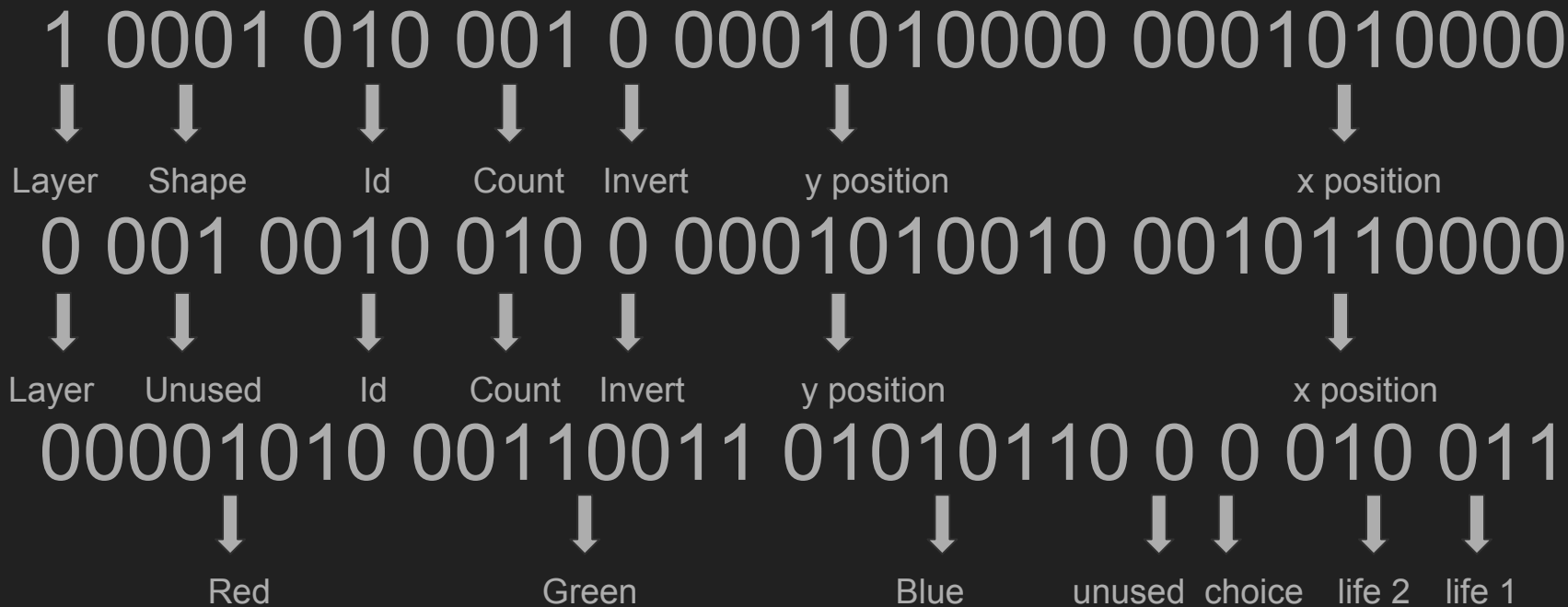
Background

- The background has a size of 320x240 and the hardware increases its size by a factor of 2
- The color palette of the background has 4 colors



Hardware

The instructions were used in the following ways



Software

The software can be categorized in three different parts:

1. Main game logic
 - Implemented by a user C program
 - Randomly generated grounds
 - Collision Handling
 - Gravity Simulation
 - Input received from user through an accelerometer to move the characters

Software

2. Communication with our hardware

- ioctl calls
- send sprites and their position

3. Communication with peripherals

- Uses USB library
- Accelerometer treated as keyboard
- Treats information received from peripheral