Janish Yerra - jyerra2
Akhil Kandimalla - kandmll2

4th April 2017

ECE 385
Final Project
Proposal

Pokemon FPGA
Game

Idea and Overview

We propose to carry out a low-level implementation of the popular Nintendo game "Pokemon" using the DE2-115 development board and the NIOS II CPU as well. The underlying concept of this game is linked to the fictional anime series - "Pokemon" in which the world is inhabited by several intelligent creatures called Pokemon, which share a friendly relationship with people. These Pokemon are raised by Pokemon trainers to fight and move ahead in this world while several others live in harmony with their Pokemon, going about their everyday lives.

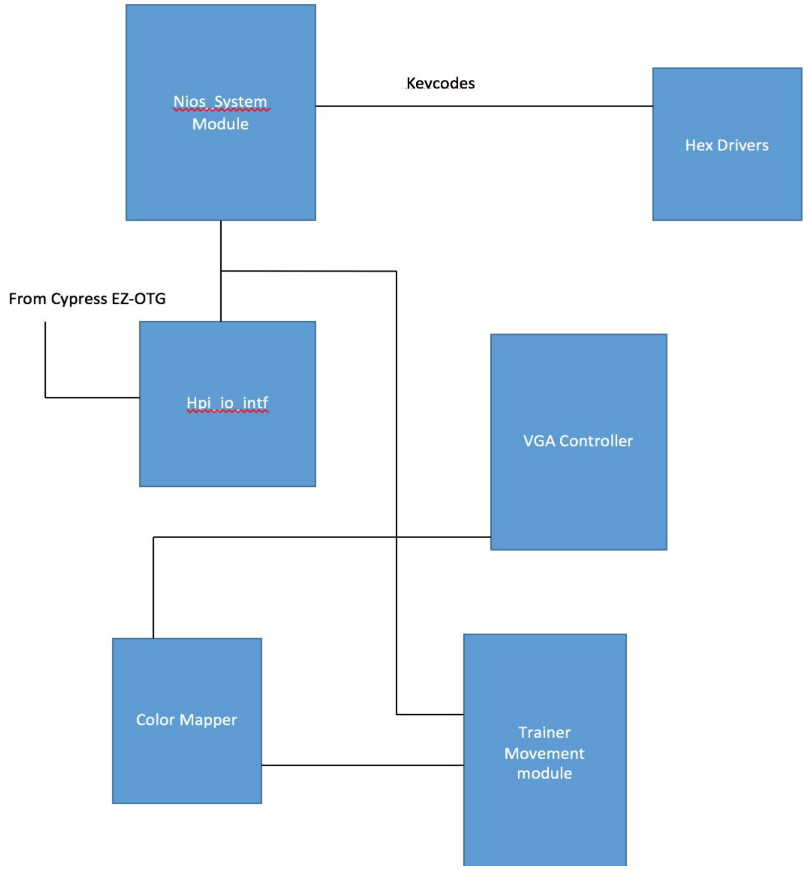
For people unfamiliar with the game, the player controls a Pokemon trainer in a small city, exploring the region with the Pokemon he/she has caught in the wild and striving to train his Pokemon to, one day, be the best in the world. There are two modes to the game: the one where we actually control the player on the map and the fight mode, in which we can control the moves our Pokemon inflict on the opponent Pokemon.

In our design, we seek to implement a basic version of this Pokemon game, into which features can be added or removed till the day of the demonstration. Since the FPGA has limited memory and we would need the trainer to move across a large area, we want to make predefined PNG to hex images which can be stored on an SD card and accessed by the FPGA. In our basic implementation, the screen will be static and the trainer will be controlled by the 'W,S,A,D' keys of a keyboard connected to the Cypress EZ-OTG Host Controller on the board. There will be the image of wild grass on one part of the static map, in which wild Pokemon live. We want the fight mode to go active randomly when the trainer walks around in the grass and the fight mode will become active which will enable us to choose attacks to fight against wild Pokemon.

We hope to build on this basic functionality and incorporate 9 maps in total such that, the screen would load a different map once the trainer steps onto the boundary of the map he/she is in currently, a corresponding stored map is loaded onto the VGA display. There is also a single Fight Screen which is constant for every fight except for the Pokemon involved in them.

Block Diagram

The Block Diagram to implement this functionality will be very similar to the block diagram that we implemented in Lab 8. We will be using an SV module to control the moving trainer. Since the NIOS II is used to interface with the Host Controller and get the keycode values, we will define a module for this in the top level file in System Verilog. We will be using the hpi_io_intf module in order to communicate with the Host with the NIOS II system which we can use to send the values to the hardware. The trainer has a particular module and movement around the screen is controlled by the keycodes. A color mapper module is used to define entire screens. The VGA controller makes use of the above two modules in order to send a display into the hardware. Hex Drivers are included as well since displaying the current keycode would make it easier to debug.



List of Features

Baseline Features:

- **Select a Pokemon at the start of the game**
- **The ability to move the trainer sprite around a static map using arrow keys**
- **Fight against another pokemon using one attack for each Pokemon**
- **Make a user interface for a basic wild Pokemon battle**
- **Incorporate basic sprites for three Pokemon**

Additional Features:

- **Each Pokemon would have multiple attacks the opposite pokemon during the battle o**
- **Dynamic view or screen shifts**
- **The Pokemon would get healed when it visits a pokemon center**
- **Additional Pokemon Sprites**
- **Sprites stored in Memory using given PNG to Hex Module**
- **(Optional) Changing sprite of trainer according to direction of movement**
- **Incorporation of audio at beginning of game**
- **Integrating trainer battles**

Expected Difficulty:

We believe the baseline version of the game would fetch us 4-5 points for difficulty. The aspects of the project that would be challenging to implement would be designing the maps and making sure that we can store it on an SD card. The final version of the project which includes all the additional features would fetch us about 6-8 points for difficulty as

it would be an arduous task to be able to implement a dynamic view of the map, and add more details and functionality to the project.

Proposed Timeline:

Week 1: Develop a static map and sprites for the trainer and a few pokemon

Week 2: Implement all the baseline features

Week 3: Implement the dynamic view/screen shifts, and added attacks for the pokemon and trainer battles.

Week 4: Changing sprite of trainer depending on the movement, incorporating audio, and all the other additional features.