PROJECT REPORT-Milestone 3

GROUP:

LONGCIRCUIT

GROUP MEMBERS:

ASHBAH FAISAL 08271

MANAL HASAN 08438

M. NAASEH SAAJID 08085

RIDA FATIMA 08463

PROJECT NAME:

COLOR HUNTER

ABSTRACT:

The COLOR HUNTER implemented on the Basys 3 FPGA is an engaging and challenging platform where players navigate through various levels represented by maps, aiming to reach designated blue boxes while avoiding hazardous red boxes. The player, depicted as a green box, moves in a distinctive manner, traversing the entire length of the map with a single key press.

The game introduces strategic elements, requiring players to plan each move meticulously, as they cannot navigate step by step. Movement restrictions are imposed by pink boxes, halting the player's progression when encountered. The intricate design encourages players to calculate their moves strategically, enhancing the cognitive challenge.

The input system relies on buttons, where each key press triggers a predefined movement of the green box. The output is visually displayed on the Basys 3 FPGA, with the player aiming to successfully reach all blue boxes to advance to the next level. Failure to do so, such as landing on a red box, results in the termination of the game, signified by a sad face display.

The flow diagram illustrates the sequential progression of the game, detailing how player inputs interact with the game logic to determine the movement of the green box. The incorporation of pink and red boxes introduces an element of risk and consequence, fostering a dynamic and engaging gaming experience. Overall, the blue box explorer on Basys 3 FPGA offers a unique and strategic gaming challenge, combining precise movement planning with an immersive visual interface.

INDEX:

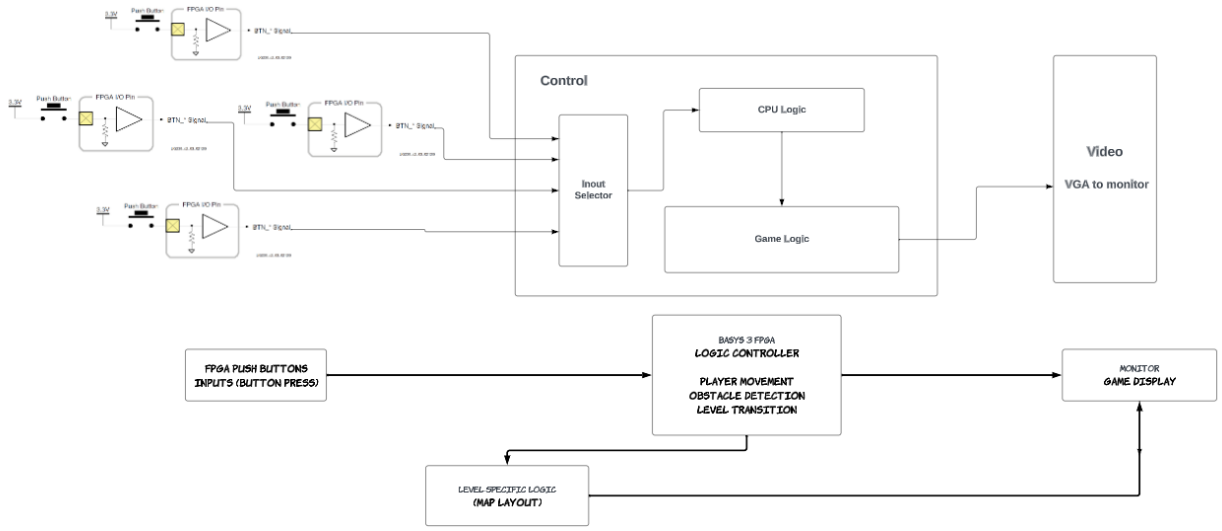
PROJECT INTRODUCTION:

In the color hunter, we have several levels with different maps. The goal of the game is to avoid red color boxes and navigate through the level maps to reach the blue boxes, currently via buttons on the basys 3 fpga.

The game has several restrictions:

- Player movement is represented through a green colored box
- Player goal is to get to every blue box on the map, while avoiding the red boxes
- The movements are set up in a way that when a player presses the left key, the green box shifts all the way to the left end of the map, not just one step at a time. This movement is the same for all the other movements.
- There are pink boxes that will restrict the movement of the player. Let's say if a player presses the down button and, on the way, down to the very end, if it lands on a pink box, the player stops right there.
- If the player ends up on a red box, the game ends with a sad face display.
- There are 8 movements, UP, DOWN, LEFT, RIGHT, and 4 diagonals (UP-LEFT, UP-RIGHT, DOWN-LEFT, DOWN-RIGHT)

This game does not give players the liberty to move one step at a time, therefore they'll have to calculate each and every move of theirs, or else they'll be stuck.



In this detailed block diagram:

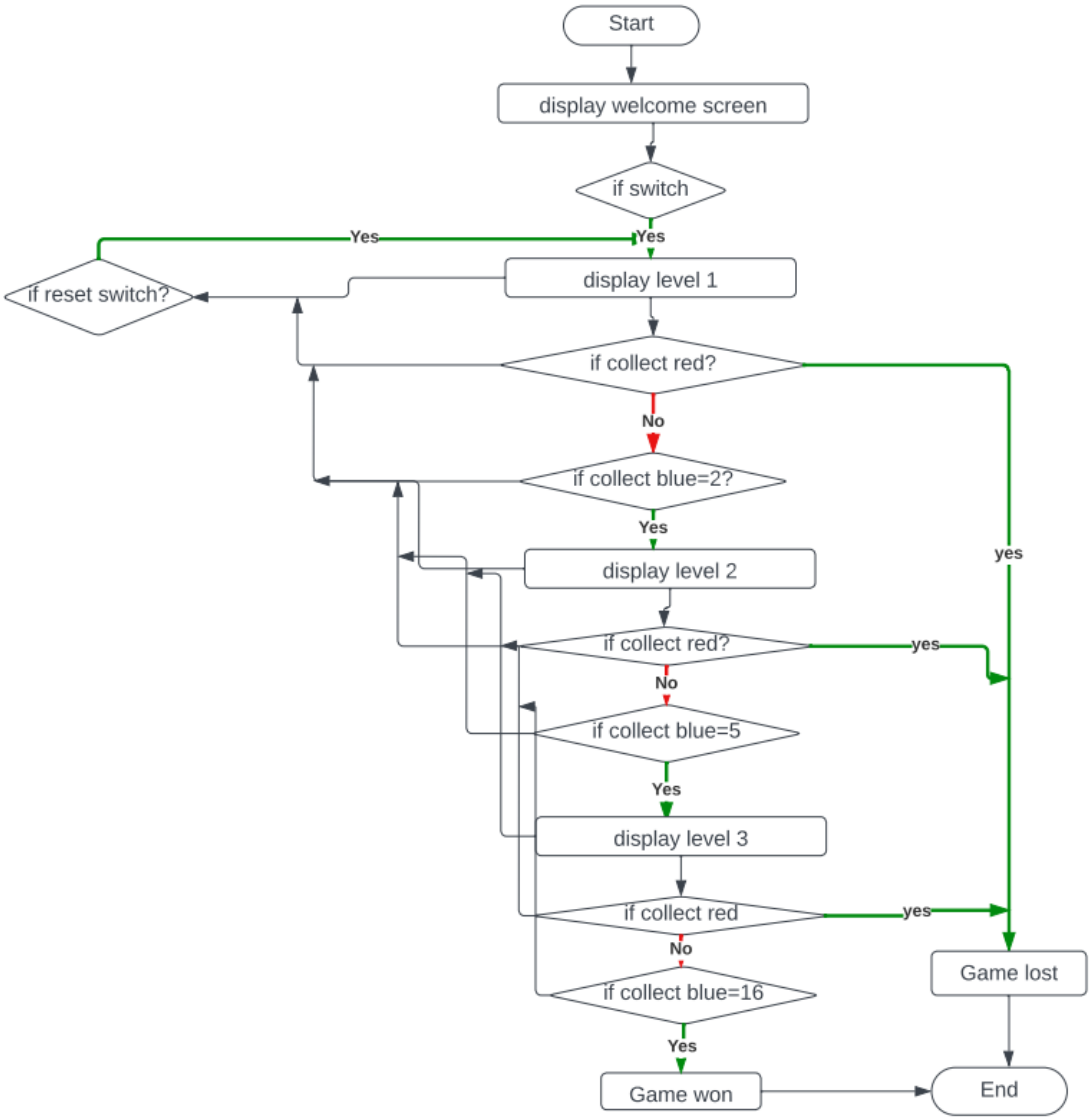
- The "FPGA Push Buttons" component represents the player inputs, specifically button presses.
- The "Basys 3 FPGA" component includes a Logic Controller responsible for handling various aspects of the game, such as player movement, obstacle detection, and level transitions.
- The "Computer Screen" component handles the visual display of the game environment, including the game display with relevant graphics and feedback.
- The "Game Levels and Logic Control" component represents the overall game control, including the logic for different levels and level-specific elements like map layouts.

Arrows indicate the flow of information:

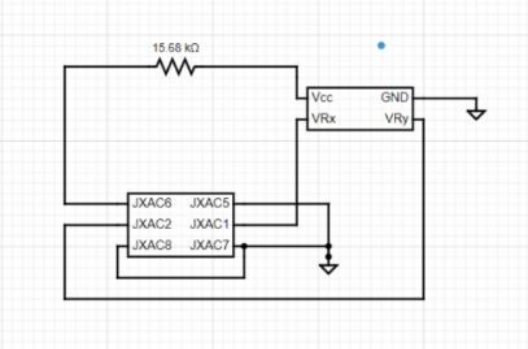
- Player inputs from the joystick influence the "Basys 3 FPGA."
- The "Basys 3 FPGA" processes this information through its Logic Controller and communicates with the "Computer Screen" for visual updates.
- The "Level-Specific Logic" within the "Game Levels and Logic Control" component influences the overall game logic, creating a dynamic and engaging gameplay experience.

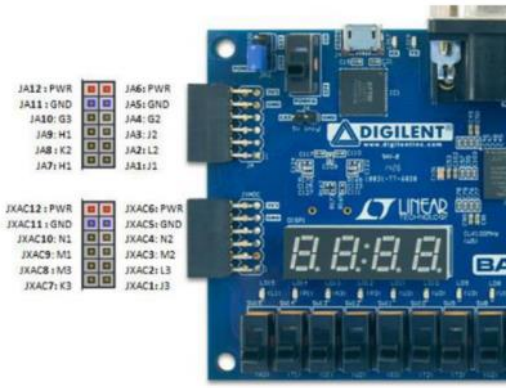
This detailed block diagram provides a more granular view of the interactions and components within the game system

USER FLOW DIAGRAM:


Start
display welcome screen
if switch
Yes
Yes
if reset switch?
display level 1
if collect red?
No
if collect blue=2?
Yes
display level 2
yes
if collect red?
yes
No
if collect blue=5
Yes
display level 3
if collect red
yes
No
Game lost
if collect blue=16
Yes
Game won
End

INPUT PERIPHERAL:





The input peripheral integrated into the Basys 3 FPGA board for movement utilizes a joystick as the primary control device. The joystick provides a versatile and intuitive interface for navigating the sprite in different directions.

To enable upward movement, the joystick can be tilted or pushed forward. The sprite ascends continuously until it reaches either the top boundary or encounters a grey block. If the sprite encounters a pink block during its upward movement, it halts in that position.

For leftward motion, the joystick can be tilted or pushed to the left. Similar to upward movement, the sprite moves continuously to the left until it reaches the end of the row or encounters a grey block. A pink block serves as a stopping point for leftward movement.

To facilitate rightward movement, the joystick is tilted or pushed to the right. The sprite moves continuously to the right until it reaches the end of the row or encounters a grey block. Once again, a pink block acts as a stopping point for rightward movement.

The downward movement of the sprite is controlled by tilting or pushing the joystick downward. As with other directions, the sprite moves downward continuously until it reaches the last block or encounters a grey block. A pink block functions as a stopping point for downward movement.

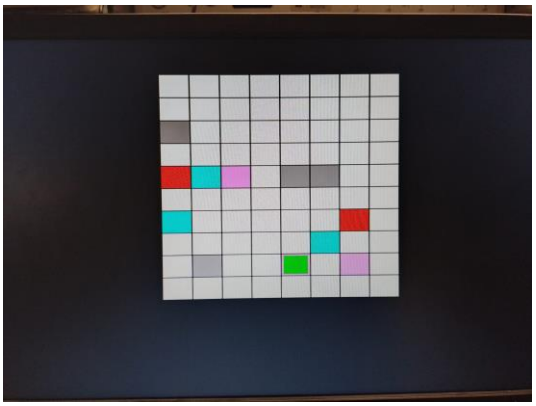
Additionally, the sprite has the capability to move diagonally when the joystick is pushed in a direction between the four main ones. This allows for dynamic and flexible control, providing a seamless and engaging user experience for sprite navigation on the Basys 3 FPGA board.

Switch V17 is used as reset.
Switch W13 is used as start, when the switch is off title screen is displayed, once the switch is turned on it transitions to game.

OUTPUT PERIPHERAL:

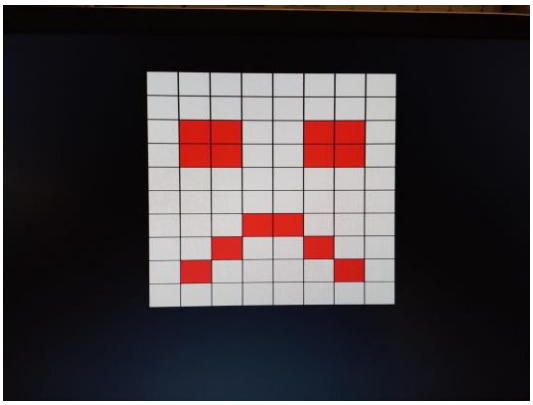
The output of our game is on screen, using vga controller and pixel_gen modules. Following are the screens of the game:

Level 1

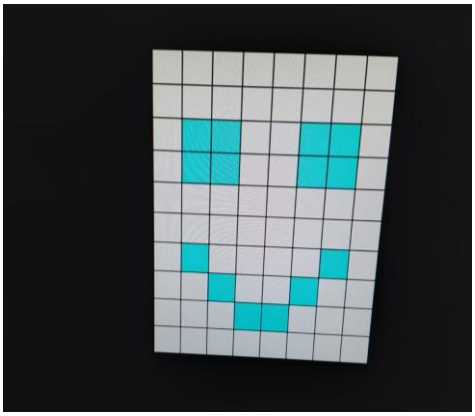


Level 2

level 3



Lose screen



Win screen

REFERENCES:

https://github.com/AdrianFPGA/basys3

https://github.com/FPGADude/Digital-Design/tree/main/FPGA%20Projects/VGA%20Projects/Pong%20pt2

https://github.com/BerkMandiracioglu/DiamondExplorer

https://github.com/muhammadali74/Basys3-Joystick-Interfacing