# ECE 385 Final Project Proposal

## “Game Crazy Arcade”

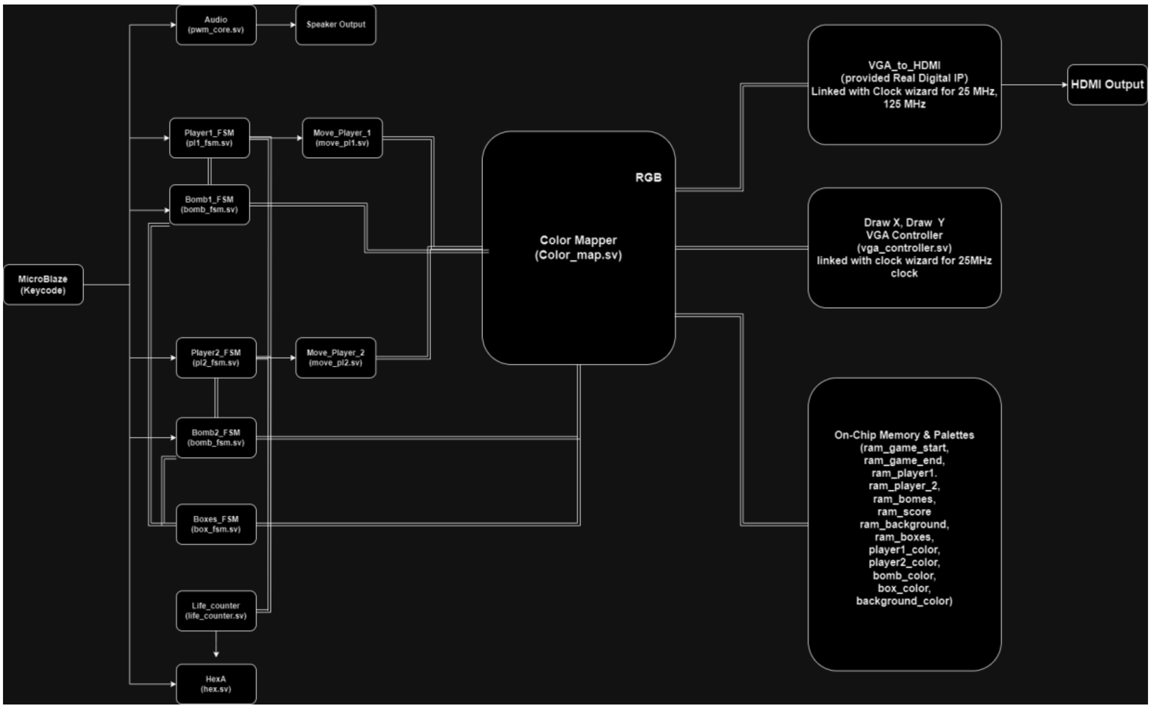
Yanghonghui Chen yc47
Yuxuan Li yuxuan43

## Part I Idea and Overview:

We propose to design a game following Crazy Arcade’s style base on FPGA. It’s a multiplayer game that supports two users playing against each other while using the same USB keyboard to provide input control signals. Player 1 should use W, A. S, D to make movement and space to place bomb, while the second player use up, left, down and right array keys to control movement and 0 to place the bomb. The rule of the game is simple, the game is launched with a standby mode, and it starts on player’s press on Enter key, initializing the game’s map consists of destroyable boxes and undestroyable stone blocks. Each player is initialized with ability to move and plant bomb, and they’re placed diagonally on different side of screen. The goal for each player is to use the bomb to destroy boxes so that they can get closer to the other one and blow the other up within close distance. We also plan to implement a score counter that can show winning and losing data of each player on the Hex displayer.

The game is FPGA-board based, and it will be built with both C code and SystemVerilog modules. Our overall structure is similar to Lab 6.2, while C codes provides the data transformation between USB keyboard and FPGA board, and .sv modules to implement game’s main function. To display the game on screen properly, we plan to use HDMI cable that connects to external monitor to provide support. The SystemVerilog essential components used including system bus that connects modules, Microblaze CPU and the MAX3421E USB host chipset, VGA Video Support and Keyboard, and on-chip memory.

## Part II Block Diagram:



## Part III List of Features:

- Baseline set of features:
    1. Background

       The background image of the game. There should be two different backgrounds, one for standby mode and one for play mode. Backgrounds should be stored in BRAM. The standby mode background is just a transitional image, while the play mode background contains the information of initial state and position of Crates, blocks and players. If one player press Enter in standby mode, the background should switch to play mode and start the game.
    2. Players

       The characters controlled by users. There should be two different players in the game, controlled by different keycodes. Player can move around within free space but can't go through crates and blocks. Player can use Space/ 0 to place a bomb at its current location. Each player has a hitbox and three lives, if a player doesn't run out of the range of an explosion, then he will lose one life. If one player run out of life, then the game is over, and his opponent will win.
    3. Bombs

       Bomb is the core component of our game. Bomb will explode in 2 seconds and raise an explosion zone that destroy all destroyable items within it (crates and player). Bomb is the only method that player can use to influence game state. Bomb has a cooldown period, within which player can't place a new bomb.

    4. Crates/Blocks
       Crates and Blocks are items within in the map. Crates can be destroyed by bomb's explosion while block can't. They can constraint the movement of players.
    5. Timer
       The timer is the constraint for game state, forbidding the case that both two players play negatively, and no one play against each other so that game can't end normally. We set Timer should count how much time has passed since the game started, and when it reaches 120 seconds it should send a compulsory ending signal so that the game should terminate.

- Additional features for extra difficulty:
  1. Life Tracker
     The tracker helps users track the real-time state of the game by showing remaining life for each player. We propose to show the value of these trackers on Hex displayers. If the time is terminated by timer, player with more remaining life on tracker should win.
  2. Audio (Background Music)
     We plan to use buzzer on FPGA chip to play a background music. This music should be played continuously during game mode, and terminate as the game reach an end.
  3. Random Bonus in Crates
     This feature is to enhance fun and randomness of the game. Bonus could be distributed randomly in crates. While player destroy a crate that contain bonus, he can receive one of the following bonuses randomly: one extra life, extend range of explosion and shorten cooldown time of bomb.

## Part IV Expected Difficulty:

Our expectation of the difficulty score is within the range of 6-7. Our design is real-time multiplayer game, including basic animation of player moving and exploding. It supports accurate gameplay and real-time life score keeping, while user can interact with items in map using USB keyboard. Moreover, we propose adding randomness and background audio into the game.

## Part V Proposed Timeline:

Week 1: set up the background, initialize and set up BRAM for use. Implement FSM and USB signal support for players so that they can move controlled by keyboard.

Week 2 (Checkpoint): Implement FSM for crates and bombs, make sure player can place bomb and crates can interact with bomb normally.

Week 3: Random bonus logics and the use of Hex displayer.

Week 4: Extra functionalities including audio.