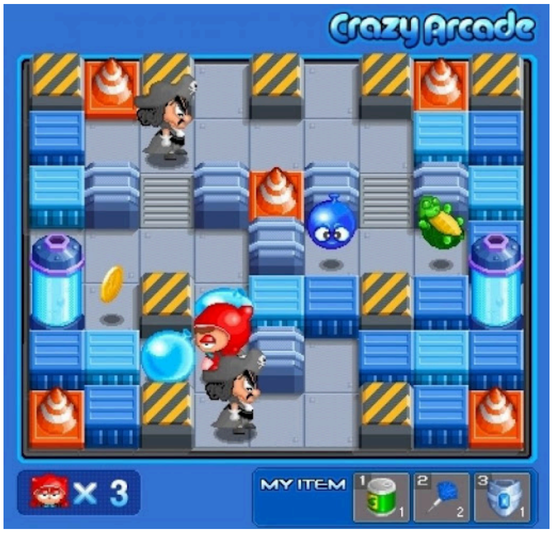
Crazy Arcade Proposal

Team Member:

Yuqi Lin (yl5334), Yelin Mao (ym3000), Hongkuan Yu (hy2819)

Project Overview:

The goal is to develop a "Crazy Arcade"-style game (shown below) on an FPGA board, leveraging SystemVerilog for hardware implementation and C for software components. The game features two players competing in real-time, navigating through a maze, placing bombs to destroy obstacles and the other player, and collecting power-ups (still considering). User input will be controlled through keyboard inputs W A S D, ↑←↓→ for movements, and Spacebar and Enter for placing bombs. Bombs will explode after a certain time.



Implementation:

Hardware Design (FPGA & SystemVerilog)

- Keyboard Input: Interface the FPGA with keyboard input capture player actions. Use SystemVerilog to process these inputs, translating them into directional commands for the players in the game.
- VGA Display Controller: Develop a VGA controller in SystemVerilog capable of rendering the game's graphics. This includes the maze layout, characters, bombs, (power-ups).

Hardware-Software Communication

- Dynamically update the game environment based on player actions and game events, requiring real-time communication between the FPGA and the C program.

Software Development (C)

- Game Logic: Implement the core game logic in C, including player movement based on keyboard input, bomb timing and explosions, collision detection, and (power-up mechanics).
- Graphics Processing: Calculate the positions of characters, platforms, and other sprites based on the game logic. Generate drawing commands to send to the FPGA for rendering on the VGA display.
- Game Rules: Enforce game rules in the C program, determining when a player has won or lost, and updating the game state accordingly.

Milestones:

- Graphics Rendering: Verify that the VGA controller correctly renders the game graphics based on commands from the C program.
- Input Processing: Test the keyboard input processing to ensure player movements are accurately captured and translated into game actions.
- Gameplay: Conduct thorough gameplay testing to ensure that the game rules are correctly applied, and the game provides a fun and challenging experience for two players.
- Optimizing and Testing