

CSEE 4840 Embedded System Project Proposal

GridBrawl Game

Anyongyong Zhao(az2932)/Sitao Zhang(sz3419)/Siyao Yu(sy3342)

Spring 2025

I. Introduction

Our team plans to design a two-player, tile-based, competitive game implemented on the DE1-SoC FPGA platform. The game takes place on a closed map filled with hard walls and soft walls. Two players compete on the map by running across tiles to capture territory. Players can also place bombs to destroy soft walls, create paths, and attack the opponent. If a player is hit by a bomb explosion, the player dies, and all tiles previously captured by that player become empty again. The first player who captures 50 tiles wins the game.



Core Mechanics

When a player moves onto an empty tile, the tile immediately changes to that player's color, and the player's score increases by 1.

If a player moves onto an opponent's tile, the tile will change to the current player's color. The current player's score increases by 1, and the opponent's score decreases by 1.

II. System Architecture

The GridBrawl system uses a hardware–software co-design approach on the DE1-SoC platform, where the HPS handles core game logic and state management, while the FPGA handles VGA graphics output, audio playback, and polling input from custom game controllers connected through GPIO.

III. Implementation Modules

3.1 Software

- **Game Logic**

Implements the main rules of the game, including player movement, bomb placement, tile coloring, and win condition checking.

- **Collision Detection**

Checks interactions between players, walls, and bombs to ensure valid movement and correct game behavior.

- **Graphics Rendering**

Updates the framebuffer to display the grid-based map and dynamic objects such as players, bombs, and explosion effects.

- **Score Management**

Tracks the number of tiles captured by each player and updates the score during gameplay.

- **Input Processing**

Receives player input data from the controller interface and converts it into game actions.

- **Audio Control**

Triggers background music and sound effects based on game events such as bomb placement, explosions, and tile capturing.

3.2 Hardware

- **FPGA Platform**

Provides hardware support for VGA output and connects the display hardware with the processor system.

- **VGA Controller**

Generates VGA timing signals and continuously reads pixel data from the framebuffer to drive the VGA monitor.

- **VGA Display**

Displays the game map, player positions, and other game elements in real time.

- **Audio Output System**

The system will use the onboard audio CODEC for audio output. The FPGA will support audio playback for background music and simple sound effects such as bomb placement, explosions, and tile capturing.

- **Custom Gamepad Interface**

Two external game controllers will be connected through the GPIO interface. A custom FPGA module will read the button states and forward the input signals to the processor for game control.

IV. Display and Graphics Design

- **Resolution and Frame Rate**

The game will use a standard VGA resolution of 640×480 pixels and will target a refresh rate of 60 frames per second. This resolution is supported by the DE1-SoC platform and provides enough screen space for a clear grid-based game display.

- **Tile-Based Map Display**

The game map will be displayed as a tile-based grid. Each tile represents one game cell and may correspond to an empty tile, a soft wall, a hard wall, or a tile captured by one of the players. This aligns with the game's grid-based structure and simplifies screen updates.

- **Tile Size and Grid Size**

We plan to use a fixed tile size to ensure the map is displayed clearly on the screen. One possible design is a 20×15 grid, with each tile drawn as a 32×32 -pixel block. This would use 640×480 pixels exactly and fits the VGA screen well.

- **Framebuffer Update**

The screen image will be stored in a framebuffer. The game program running on the HPS will update the framebuffer whenever the game state changes, such as player movement, tile color changes, bomb placement, or explosion effects.

- **VGA Output**

The FPGA continuously reads pixel data from the framebuffer and generates the VGA timing and video output signals. This allows the display to update in real time while keeping the game logic on the processor side.

- **Graphics Objects**

The display will include both static and dynamic objects. Static objects include the map tiles and walls, while dynamic objects include the two players, bombs, and explosion effects.

V. Major Tasks

- Develop the VGA graphics system on the FPGA to generate VGA timing signals and display the grid-based game map and objects using a framebuffer.
- Design the input system using external game controllers connected through the GPIO interface, and create a hardware module that reads button states and sends input data to the processor.
- Build the core game logic in software, including player movement, tile coloring rules, bomb placement, explosion effects, and win condition checking.
- Handle collision detection between players, walls, and bombs to ensure valid movement and correct interaction with map elements such as soft walls and hard walls.
- Support audio playback using the onboard audio CODEC for background music and sound effects such as bomb placement, explosions, and tile capturing.
- Combine all system components and conduct testing to ensure stable gameplay, real-time graphics updates, and correct interaction between hardware and software modules.