

ECE 385 Final Project

Proposal: UIUC Campus Treasure Hunt

Haoru Li, Ziheng Qi

1 Idea and Overview

We propose a top-down 2D exploration game where players navigate a UIUC campus map to find hidden treasures. The game features tile-based graphics with iconic campus landmarks and provides proximity hints to guide treasure hunting.

Hardware (SystemVerilog): Tile/sprite renderer at 640×480 resolution, BRAM for map/sprite storage, LFSR random number generator, and PWM audio module. We will reuse the HDMI text controller framework from Lab 7.

Software (C on MicroBlaze): Game loop, collision detection, proximity-based hint system, score/timer display, and keyboard input handling.

Demo: Player uses keyboard to explore the campus map and collect 8-10 randomly-placed treasures with distance-based hints.

2 Block Diagram

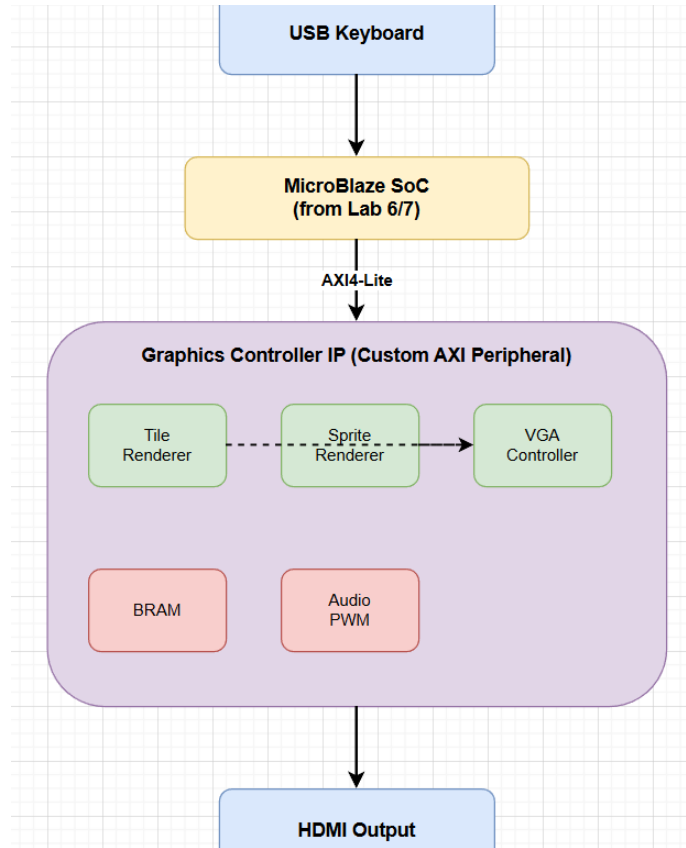


Figure 1: System Block Diagram

Note: MicroBlaze SoC includes the USB HID interface (SPI, GPIOs) from Lab 6 and baseline AXI infrastructure from Lab 7. The Graphics Controller IP extends Lab 7’s text controller with tile/sprite rendering and audio capabilities.

3 List of Features

3.1 Baseline Features

1. **Tile-based Graphics System:** 640×480 resolution display with 16×16 pixel tiles, rendering static UIUC campus map with landmarks
2. **Player Character:** Animated 16×16 sprite with 4-directional movement (WASDkeys)
3. **Collision Detection:** Tile-based obstacle detection preventing from walking through walls
4. **Treasure System:** 8 randomly placed hidden treasures using LFSR seed generation
5. **Proximity Hints:** Distance-based feedback system with three levels of hints based on Manhattan distance
6. **Score Display:** Real-time treasure counter shown on screen (e.g., “Treasures: 5/8”)
7. **USB Keyboard Input:** Full keyboard integration for movement controls

3.2 Additional Features (For Extra Difficulty)

1. **Audio System:** Background music loop and sound effects (treasure found, movement) using PWM synthesis
2. **Timer System:** Countdown or elapsed time display with high score table
3. **Animation System:** 4-frame walking animation for player character in all directions
4. **Multiple Map Regions:** 2 different campus areas with screen transitions between regions
5. **Collectible Types:** Different treasure varieties with varying point values
6. **Particle Effects:** Visual feedback when collecting treasures (sparkle animation)

4 Expected Difficulty

Baseline difficulty: 10-11 points. With additional features like additional sounds(if we successfully implement): **12-13 points.**

Difficulty breakdown:

- **Custom tile/sprite graphics system:** Multi-layer rendering with transparency is more complex than Lab 7’s character-only display
- **Audio synthesis:** PWM-based sound generation adds demonstrable complexity
- **USB keyboard controls:** Responsive game input building on Lab 6 framework
- **Complete game mechanics:** Collision detection, proximity system, score tracking
- **Hardware-software co-design:** Real-time AXI communication for game state updates

5 Proposed Timeline

5.1 Week 1: Foundation and Graphics

- Set up MicroBlaze SoC with USB keyboard (reuse Lab 6/7 components)
- Implement tile-based graphics renderer in SystemVerilog
- Create BRAM storage for tilemap data
- Design sprite rendering module with AXI interface

5.2 Week 2: Core Game Mechanics (Mid-Checkpoint Demo)

- **Checkpoint Goal:** Display static map with keyboard-controlled player sprite
- Implement collision detection in C
- Add LFSR for random treasure generation
- Develop proximity detection and hint system

5.3 Week 3: Polish and Features

- Finalize UIUC campus tilemap graphics (convert assets to appropriate format)
- Implement PWM audio module with background music and sound effects
- Add player walking animation (4-frame cycle)
- Implement UI elements (score display, timer, win screen)

5.4 Week 4: Testing and Final Demo

- **Final Demo:** Complete playable game with all features
- Comprehensive testing and bug fixes
- Performance optimization (60 fps verification, timing closure)

6 Game Graphics Preview

