EE 314 Term Project

Spring 2024

FPGA Implementation of Isometric Shooter Game

1 Introduction

In this Digital Electronics Laboratory course Term Project, your primary task is to use Verilog HDL on an FPGA platform to create an isometric shooter game. This interesting project helps you develop your critical thinking and problem-solving abilities while also fusing theoretical knowledge with real-world application. Through exploring digital circuit design, logic synthesis, and FPGA programming within the framework of a game development environment, you will acquire vital practical knowledge that can be applied to embedded systems and digital electronics in the real world. This hands-on training fosters creativity and gives you the technical skills necessary to take on challenging problems in the field.

2 Project Definition

You will implement your idea with the help of the FPGA boards that the course provides. Taking inspiration from the iconic video game Space Invaders, your goal is to create an engaging gameplay environment. You will use the VGA interface to construct the primary screen, with a resolution of 640 x 480. In this instance, players will control a central spaceship on a game field that resembles Figure 1.

This spaceship can rotate and fire projectiles to defend against incoming enemies that appear from the boundaries of the field. These enemies approach your spaceship deliberately, testing your ability to maneuver strategically. It is imperative that you destroy these enemies before they manage to get past your fortifications because failing to do so will mean game is over.

Rest assured, detailed specifications and design considerations will be provided to steer you through the implementation process seamlessly.

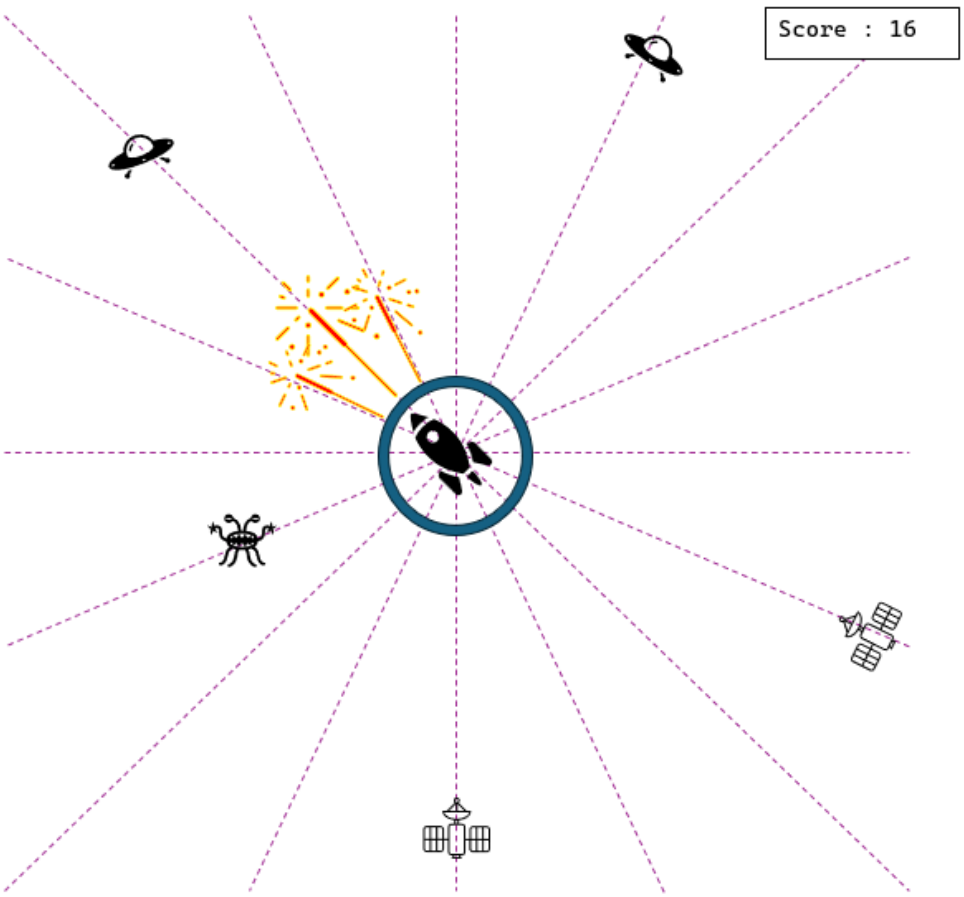

Figure 1: Example game field representation.

Specifications and Design Considerations:

1. **Spaceship Location & Movement:**
    - Spaceship is located in the middle of the game field.
    - The spaceship can rotate within its axis but cannot move laterally across the field.
    - Allow player-controlled rotation of the spaceship using buttons on the FPGA board. For example, you can assign Button 1 for clockwise rotation by 22.5 degrees, Button 2 for counterclockwise rotation by 22.5 degrees. Also, button 3 can be used for firing projectiles.

2. **Enemy Dynamics:**
    - Enemies **randomly** spawn at predefined angles from the center of the field, such as 0, 22.5, 45, 67.5 degrees, and so on (22.5 degrees apart). So that, there are total of 16 trajectories that enemies can spawn and move.
    - There must be a minimum of 2 enemies present on the field at all times.

- The maximum number of enemies on the field simultaneously should not exceed 8.
- Implement at least three distinct enemy types, each with unique health levels and shapes.
- Ensure enemies move gradually and radially towards the center where the player's spaceship is located.

3. **Shooting Modes & Dynamics:**
    - Include two shooting modes:
        - > Mode 1: The gun fires projectiles in a wider spray of 90 degrees, dealing lower damage to enemies.
        - > Mode 2: The gun fires projectiles in a narrower spray of 45 degrees, inflicting higher damage on enemies.
    - The player can switch between these modes using designated switches on the FPGA board.
    - Consider implementing visual indicators or feedback to show the damage dealt to enemies and/or remaining enemy health levels.
    - Design indicators for selected shooting mode.

Shooting mode 1

Consider a case where spaceship shoots in **mode 1** aiming at 22.5-degree trajectory as shown in the Figure 2. This implies that **all the enemies** on trajectories 67.5°, 45°, 22.5°, 0°, 337.5° gets hit by the projectiles.

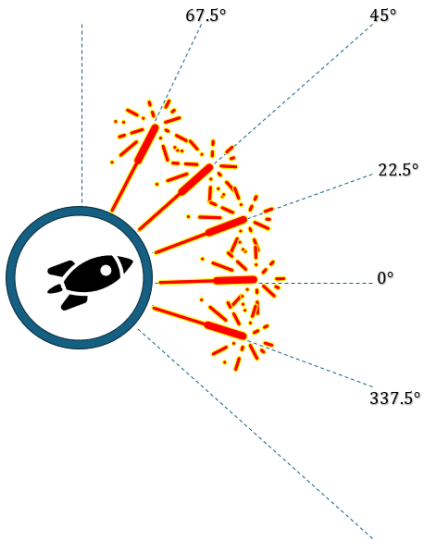

Figure 2: Example shooting for mode 1.

Shooting mode 2

Now, consider a case where spaceship shoots in **mode 2** aiming at 22.5-degree trajectory as shown in the Figure 3. This implies that **all the enemies** on trajectories 45°, 22.5° and 0° gets hit by the projectiles.

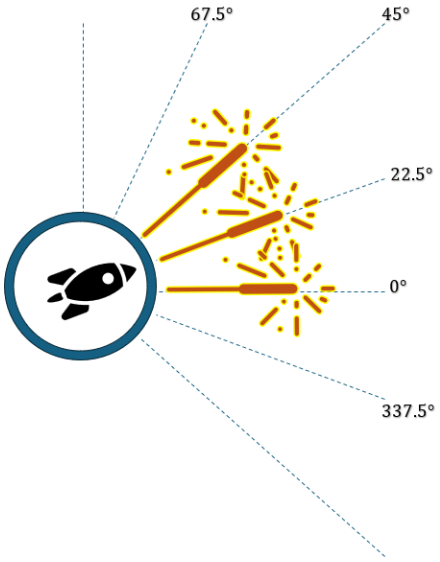

Figure 3: Example shooting for mode 2.

4. **Player Score:**
    - Incorporate a scoring system that tracks and displays the player's score as they defeat enemies.
5. **Game Over Condition:**
    - Define the game over condition when an enemy successfully collides with the player's spaceship.
    - Display a game over message or screen when the game ends, along with the player's final score.
6. **Optional Enhancements (Bonus Credits):**
    - Consider adding animations to projectiles or enemy movements for enhanced visual appeal.
    - Explore adding a background image or graphical elements to the game field for aesthetics.
    - Use LEDs, seven-segment displays on the FPGA board for visual feedback.
    - Consider adding more shooting modes.
    - Consider implementing levelling system that changes enemy speeds and/or health levels etc.

3 Final Words

We hope you will enjoy the project while learning Verilog HDL more thoroughly. We should emphasize that this is not a weekend task. Then, it would be best to allocate enough time for researching and implementing. Please keep in mind that we have limited resources, and we must benefit wisely so that new students can also benefit. FPGA boards are sensitive, fragile, and expensive boards. For the duration you borrowed them from us, you will be responsible for taking care of them.

Ensuring the game is responsive adds to the player experience, with prompt and accurate rotations and projectile firing based on FPGA board inputs. Additionally, the project allows for creative freedom in certain aspects, offering numerous implementation options. We encourage all participants to explore innovative approaches for successful project demonstrations.

Enjoy.