

Hardware Design

Final Project Proposal

FPCAT – Battle Cat on FPGA

Team 01

112062122 謝佳晉 112062144 范升維

Table of Contents:

1. OVERVIEW	1
1.1 MOTIVATION	1
1.2 BRIEF INTRODUCTION.....	2
2. PROJECT INFORMATION.....	2
2.1 HIGH-LEVEL STRUCTURE.....	2
2.2 MODULE DETAILS.....	3
2.3 ESTIMATED COST	3
3. FUTURE PLAN	3
3.1 TODO LIST	3
3.2 SCHEDULE	4

1. Overview

1.1 Motivation

The Battle Cats has been my favorite mobile game since I was in elementary school. Initially, I wasn't very interested in the game when my classmate showed me some gameplay scenes. However, after downloading and playing it for a while, I couldn't help but be captivated by The Battle Cats.



▲ Figure 1.1: one of my Instagram post, kind of old

The stage and character designs are absolutely top-tier, even by today's standards. To share the fun of playing The Battle Cats with more people, we plan to create our own version of the game using FPGA. By showcasing this project on our GitHub account, we hope to demonstrate the game's appeal while also strengthening our applications for schools or jobs in the future. Out of all the stages in the game, we've chosen three of the most classic ones, as we believe they will be especially "interesting" to players.

1.2 Brief Introduction

At the start, the game begins with a **start scene**. After pressing the **start button**, the game transitions to the **stage selection menu**. Players can choose from three stages, each with designated cats assigned to them.

In the **game scene**, players will have access to buttons for **cats**, **tower**, **purse**, and **pulse**. To **summon cats**, players must accumulate enough money, which can be earned over time and by defeating enemies. The **tower** can emit light waves after it finishes charging. Players can also spend money to expand the **purse capacity**.

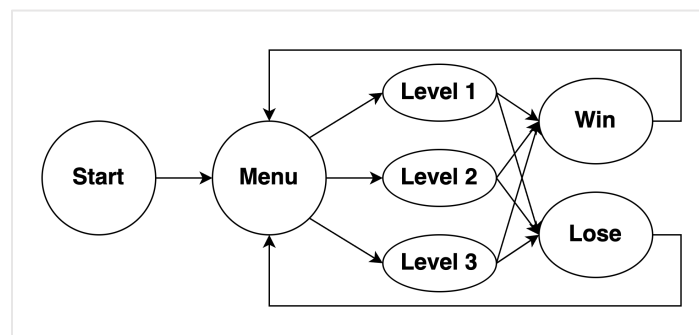
The goal is to destroy the enemy tower using the player's cats. If the player's cats succeed, the player wins. If the enemy destroys the player's tower, the player loses.

After triggering the **win scene** or **lose scene**, players can return to the **stage selection menu** by pressing the **OK button** in the center.

2. Project Information

2.1 High-Level Structure

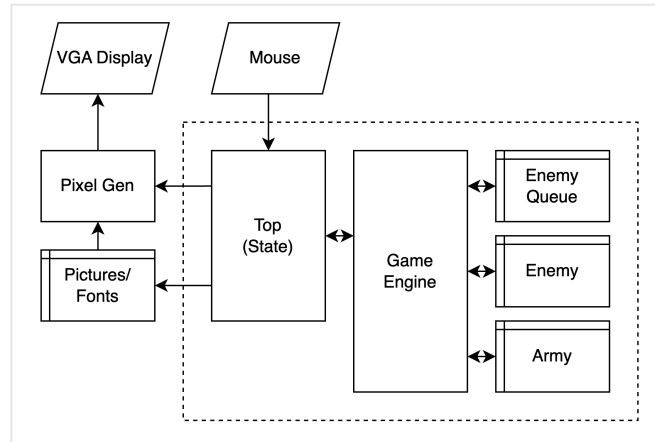
Figure 2.1 below illustrates our planned State Diagram. In homage to classic game opening sequences, we've incorporated a Start page as a nostalgic element before transitioning to the Menu page, where players can select from three progressively challenging levels. For the pause functionality implementation, we opted to directly freeze both the display and game mechanics rather than creating a separate state.



▲ Figure 2.1: State Diagram

2.2 Module Details

Mouse input serves as our primary control interface, facilitating key actions such as character deployment and turret operation. The Pictures/Fonts directory maintains essential resources including sprite images and typography assets. The Enemy Queue manages time-based enemy spawn sequences



across all three levels, while the Enemy and Army components track real-time unit information such as positioning and health points. Each character operates according to its own state diagram, encompassing various states including attack, idle, and knockback. The Game Engine oversees overall game progression, handles font rendering, and manages other auxiliary processing tasks.

2.3 Estimated Cost

Fortunately, our project requires no budget.

3. Future Plan

3.1 Todo List

- I/O Infrastructure
- IP(Storage) Protocol and Allocation
- Pictures and Fonts' Creation
- Configuration of Layer Rendering Hierarchy
- Start Scene and Menu Scene
- Game Scene and Game Engine
- Win Scene and Lose Scene

3.2 Schedule

Week	Time	Plans	Notes
13	Mon - Wed	R: I/O Infrastructure J: Pictures and Fonts' Creation	
	Thu - Sat	R: Start Scene and Menu Scene J: IP(Storage) Protocol and Allocation	Thu: Lab 6 Report
	Sun	R: Start Scene and Menu Scene J: Game Scene and Game Engine	
14	Mon - Fri	R, J: Game Scene, Game Engine and Configuration of Layer Rendering Hierarchy	
	Sat - Sun	R, J: Win Scene and Lose Scene	
15	Mon - Sun	R, J: Debug and Test	Thu: Final Exam
16	Mon - Thu	R, J: Conduct Report	Thu: Demo