

# Plane Shooter Game - Project Documentation

---

Inspired by Chicken Invaders

## Table of Contents

1. [Project Overview](#)
  2. [Project Structure](#)
  3. [Installation & Setup](#)
  4. [Architecture](#)
  5. [Development Guide](#)
  6. [Docker Deployment](#)
- 

## 1. Project Overview

### Description

Plane Shooter is a web-based arcade-style shooting game built with React and modern web technologies. Players control a plane that must dodge and destroy enemies while managing limited ammunition and health. The game features dynamic scrolling backgrounds, enemy spawning systems, collision detection, and a comprehensive HUD for tracking game stats.

### Game Host

<https://candid-frangipane-82f9e1.netlify.app/>

### Key Features

- **Interactive Gameplay:** Control a player plane using keyboard inputs (arrow keys and spacebar)
- **Enemy System:** Multiple enemy types with varied behaviors and spawn patterns
- **Combat Mechanics:** Shooting system with bullet management and collision detection
- **Health & Ammo Management:** Limited resources that affect gameplay strategy
- **Progressive Difficulty:** Enemy spawn rates increase as the game progresses
- **Visual Effects:** Smooth animations, parallax scrolling backgrounds, and dynamic rendering
- **Game States:** Menu, playing, paused, and game over states with appropriate UI
- **Responsive Design:** Adapts to different screen sizes while maintaining gameplay integrity

### Technology Stack

- **Frontend Framework:** React 18.x with Hooks (useState, useEffect, useCallback, useRef)
  - **Styling:** CSS3 with custom animations and responsive design
  - **Build Tool:** Vite for fast development and optimized production builds
  - **Containerization:** Docker for consistent deployment across environments
  - **Code Quality:** ESLint for code linting and quality assurance
- 

## 2. Project Structure

## Root Directory Layout

```

plane-shooter-app/
├── public/                # Static assets served directly
│   ├── assets/           # Game assets (images, sprites)
│   ├── index.html        # Main HTML entry point
│   └── vite.svg           # Vite logo
├── src/                  # Source code directory
│   ├── components/       # React components
│   │   ├── Bullet.jsx    # Bullet rendering component
│   │   ├── Enemy.jsx     # Enemy rendering component
│   │   ├── GameCanvas.jsx # Main game canvas and logic
│   │   ├── HUD.jsx       # Heads-up display component
│   │   └── Player.jsx     # Player plane component
│   ├── constants/        # Configuration and constants
│   │   └── gameConfig.js # Game parameters and settings
│   ├── hooks/            # Custom React hooks
│   │   ├── audio.js      # Audio management hook
│   │   ├── enemySpawner.js # Enemy spawning logic
│   │   ├── gameState.js  # Game state management
│   │   ├── imageLoader.js # Asset preloading
│   │   ├── mainGameLoopHook.js # Main game loop coordination
│   │   ├── scrollingBackground.js # Background animation
│   │   └── useGameLoop.js # Core game loop hook
│   ├── utils/            # Utility functions
│   │   └── collision.js   # Collision detection algorithms
│   ├── App.jsx           # Root React component
│   ├── main.jsx          # Application entry point
│   └── index.css          # Global styles
├── .dockerignore         # Docker ignore patterns
├── .gitignore            # Git ignore patterns
├── Dockerfile            # Docker configuration
├── eslint.config.js      # ESLint configuration
├── index.html            # HTML template
├── package-lock.json     # Locked dependency versions
├── package.json          # Project metadata and dependencies
├── README.md             # Project readme
└── vite.config.js        # Vite build configuration

```

## Component Hierarchy

```

App.jsx (Root)
├── GameCanvas.jsx (Main Game Container)
│   ├── Player.jsx (Player Plane)
│   ├── Enemy.jsx × N (Multiple Enemies)
│   ├── Bullet.jsx × N (Multiple Bullets)
│   └── HUD.jsx (Game Stats Display)

```

## File Descriptions

## Components

- **Bullet.jsx**: Renders individual bullets with position and appearance based on game state
- **Enemy.jsx**: Handles enemy rendering, positioning, and visual representation
- **GameCanvas.jsx**: Core game container managing all game logic, state, and rendering coordination
- **HUD.jsx**: Displays game statistics including score, health, ammo, and game state messages
- **Player.jsx**: Renders the player plane with current position and visual state

## Constants

- **gameConfig.js**: Centralized configuration containing game dimensions, player stats, enemy parameters, bullet properties, and timing constants

## Hooks

- **audio.js**: Manages game audio including background music and sound effects
- **enemySpawner.js**: Controls enemy generation, spawn timing, and difficulty progression
- **gameState.js**: Manages overall game state including health, score, ammo, and game phase
- **imageLoader.js**: Preloads game assets to ensure smooth gameplay without loading delays
- **mainGameLoopHook.js**: Orchestrates all game systems and coordinates the main game loop
- **scrollingBackground.js**: Creates parallax scrolling effect for game background
- **useGameLoop.js**: Provides the foundational game loop using requestAnimationFrame

## Utils

- **collision.js**: Implements bounding box collision detection for game entities

---

## 3. Installation & Setup

### Prerequisites

Before starting, ensure you have the following installed on your system:

- **Node.js**: Version 16.x or higher (includes npm)
- **npm**: Version 8.x or higher (comes with Node.js)
- **Git**: For cloning the repository
- **Docker** (Optional): For containerized deployment

### Local Development Setup

#### Step 1: Clone the Repository

```
git clone https://github.com/HadeerFawzyy44/plane-shooter-app.git
cd plane-shooter-app
```

#### Step 2: Install Dependencies

```
npm install
```

This command installs all required packages including:

- React and React-DOM for UI rendering
- Vite for development server and building
- ESLint for code quality
- All development dependencies

### Step 3: Start Development Server

```
npm run dev
```

The development server will start on <http://localhost:5173> by default. The application features hot module replacement, so changes to your code will be reflected immediately in the browser without a full page reload.

### Step 4: Access the Game

Open your web browser and navigate to the URL shown in the terminal (typically <http://localhost:5173>). The game should load automatically.

## Environment Configuration

The game uses default configurations defined in [src/constants/gameConfig.js](#). To modify game parameters:

1. Open [src/constants/gameConfig.js](#)
2. Adjust values such as player speed, enemy spawn rates, bullet damage, etc.
3. Save the file and restart the development server

Example configuration options:

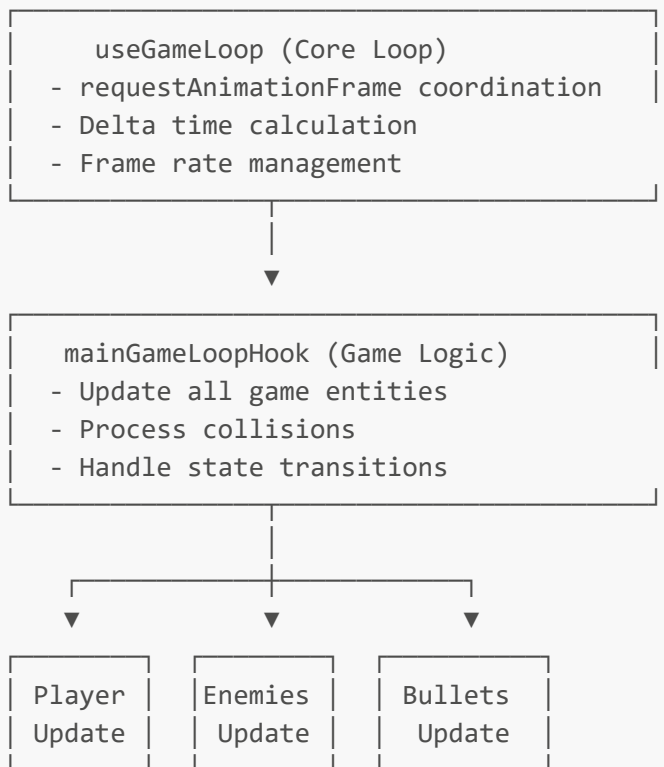
- [GAME\\_WIDTH](#) and [GAME\\_HEIGHT](#): Canvas dimensions
- [PLAYER\\_SPEED](#): Movement speed of player plane
- [ENEMY\\_SPAWN\\_INTERVAL](#): Time between enemy spawns
- [BULLET\\_SPEED](#): Projectile velocity
- [MAX\\_AMMO](#): Starting ammunition count

---

## 4. Architecture

### Game Loop Architecture

The game follows a classic game loop pattern coordinated through custom React hooks:



---

## 5. Docker Deployment

### Docker Hub Image

The application is available as a pre-built Docker image on Docker Hub for easy deployment.

#### Image Information

- **Image Name:** `hadeerfawzyy/plane-shooter-app`
- **Tags:** `latest` (most recent build)
- **Base Image:** Node.js with Nginx for serving static files
- **Size:** Optimized for production deployment

### Pulling and Running from Docker Hub

#### Step 1: Install Docker

**Windows/Mac:** Download and install Docker Desktop from [docker.com](https://docker.com)

**Linux (Ubuntu/Debian):**

```
sudo apt-get update
sudo apt-get install docker.io
sudo systemctl start docker
sudo systemctl enable docker
```

**Linux (CentOS/RHEL):**

```
sudo yum install docker
sudo systemctl start docker
sudo systemctl enable docker
```

**Step 2: Pull the Image from Docker Hub**

```
docker pull hadeerfawzyy/plane-shooter-app:latest
```

This command downloads the pre-built image from Docker Hub. The download time depends on your internet connection speed (image is typically 50-150MB).

**Step 3: Run the Container****Basic Run Command:**

```
docker run -d -p 8080:80 --name plane-shooter hadeerfawzyy/plane-shooter-app:latest
```

**Command Explanation:**

- `-d`: Run container in detached mode (background)
- `-p 8080:80`: Map port 8080 on host to port 80 in container
- `--name plane-shooter`: Assign a friendly name to the container
- `hadeerfawzyy/plane-shooter-app:latest`: The image to run

**Alternative Port Mapping:**

```
# Use port 3000 instead
docker run -d -p 3000:80 --name plane-shooter hadeerfawzyy/plane-shooter-app:latest

# Use port 80 (requires sudo on Linux)
sudo docker run -d -p 80:80 --name plane-shooter hadeerfawzyy/plane-shooter-app:latest
```

**Step 4: Access the Application**

Open your web browser and navigate to:

- `http://localhost:8080` (or whatever port you specified)
- `http://YOUR_SERVER_IP:8080` (for remote servers)

The game should load and be fully playable.

## Docker Container Management

### View Running Containers

```
docker ps
```

### View All Containers (including stopped)

```
docker ps -a
```

### Stop the Container

```
docker stop plane-shooter
```

### Start a Stopped Container

```
docker start plane-shooter
```

### Restart the Container

```
docker restart plane-shooter
```

### Remove the Container

```
# Stop first if running
docker stop plane-shooter

# Remove container
docker rm plane-shooter
```

### View Container Logs

```
docker logs plane-shooter
```

```
# Follow logs in real-time
docker logs -f plane-shooter
```

## Execute Commands in Running Container

```
docker exec -it plane-shooter /bin/bash
```

## Building Your Own Docker Image (Optional)

If you want to build the image locally from source:

### Step 1: Ensure Dockerfile Exists

The repository includes a Dockerfile with the following structure:

```
# Build stage
FROM node:18-alpine AS build
WORKDIR /app
COPY package*.json ./
RUN npm ci
COPY . .
RUN npm run build

# Production stage
FROM nginx:alpine
COPY --from=build /app/dist /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

### Step 2: Build the Image

```
docker build -t plane-shooter-app:local .
```

### Step 3: Run Your Local Build

```
docker run -d -p 8080:80 --name plane-shooter-local plane-shooter-app:local
```

---

## Additional Resources

### Useful Commands Summary



```
# Development
npm install          # Install dependencies
npm run dev          # Start development server
npm run build        # Build for production
npm run preview      # Preview production build

# Docker
docker pull hadeerfawzyy/plane-shooter-app:latest # Pull image
docker run -d -p 8080:80 --name plane-shooter \
  hadeerfawzyy/plane-shooter-app:latest          # Run container
docker ps                                         # List containers
docker logs plane-shooter                        # View logs
docker stop plane-shooter                        # Stop container
docker start plane-shooter                       # Start container
docker rm plane-shooter                          # Remove container
```

## Project Links

- **Docker Hub:** <https://hub.docker.com/r/hadeerfawzyy/plane-shooter-app>
- **Repository:** <https://github.com/HadeerFawzyy44/plane-shooter-app>

---

**Last Updated:** 2025 **Version:** 1.0.0 **Maintainer:** Hadeer Fawzy