# HOPE FOUNDATION – FINAL PROJECT


## 🎮 PyGalaxian —

### 🧩 Overview

**PyGalaxian** is a **2D vertical scrolling space shooter** game built using **Pygame**. It involves controlling a spaceship to defeat enemy ships, saucers, drones, space stations, and eventually a boss, all while avoiding enemy bullets and managing your health.

GUIDE – NEHA MAM

# Game Features

✅ **Main Features:**

•Smooth player movement and shooting

•Different enemy types:

- Basic enemies
- Drones
- Saucers
- Space stations
- Final boss

•Dynamic wave system with increasing difficulty

•Health packs for recovery

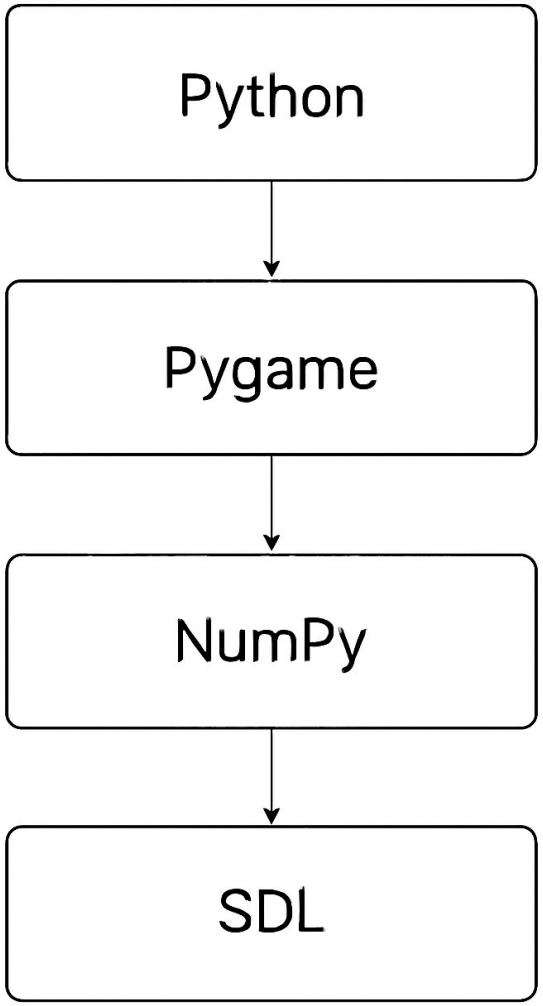
•Score system and health bar

•Cool visual effects like explosions and starfield background

•Background music and sound effects

•Game menu and game over screen

# Technology Stack




| Component | - | Description |
|---|---|---|
| **Language** | - | Python |
| **Library** | - | Pygame |
| **Graphics** | - | Custom sprites and animations |
| **Audio** | - | Sound effects using .ogg and .wav files |
| **Platform** | - | Windows/Linux (Planned: Android) |

# PyGalaxian

## Game Flow

### 1.Start Screen

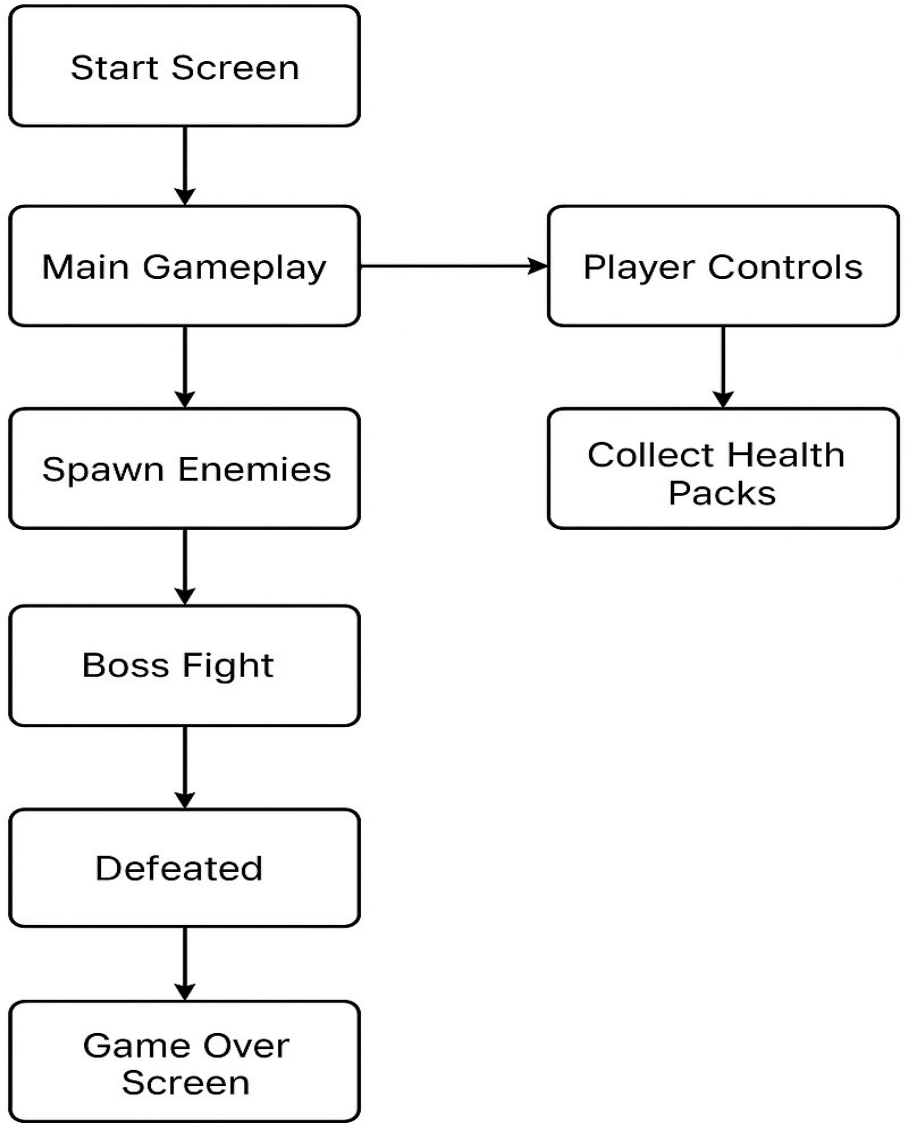
Shows title and Play/Exit options.

### 2.Main Gameplay

1. Spawn enemies according to wave progression.
2. Avoid bullets and collisions.
3. Collect health packs.
4. Defeat the **Boss** in the final stage.

### 3.Game Over Screen

1. Display final score.
2. Option to exit.



# Game Objects

### 👩‍🚀 Player

- Can move left/right and shoot.
- Has health and score.
- Autopilot mode before boss appears.

### 👾 Enemies:

- **Basic Enemy:** Comes in multiple designs, shoots bullets.
- **Drone:** Moves vertically, stops and shoots powerful patterns.
- **Saucer:** Animated circular ship with strong lasers.
- **Station:** Rotating shooter that releases enemies.
- **Boss:** Huge spaceship with complex attacks and multi-directional bullets.

### 💊 Health Packs:

- Restores health.
- Moves in zig-zag patterns.

### 🔫 Bullets:

- Player’s laser (green)
- Enemy bullets (varied colors and shapes)

### 💥 Explosions:

- Animated sprite-based explosion upon enemy death.

# Wave System (Enemy Spawn Logic)

- Uses a wavecounter that increments over time.
- Based on its value, different enemies spawn:

Code:

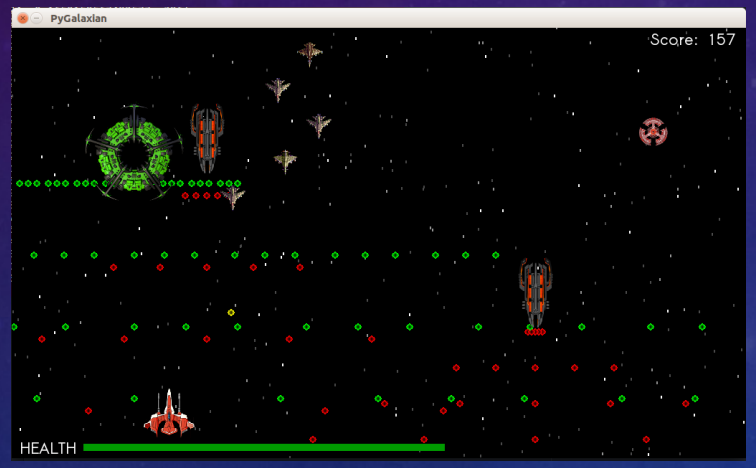
```
def storyboard(wavecounter):
    if 0 <= wavecounter <= 700:
        return 0  # basic enemies
    elif 700 < wavecounter <= 1100:
        return 1  # saucers
    ...
    elif wavecounter > 4400:
        return 11  # boss
```

# Visuals

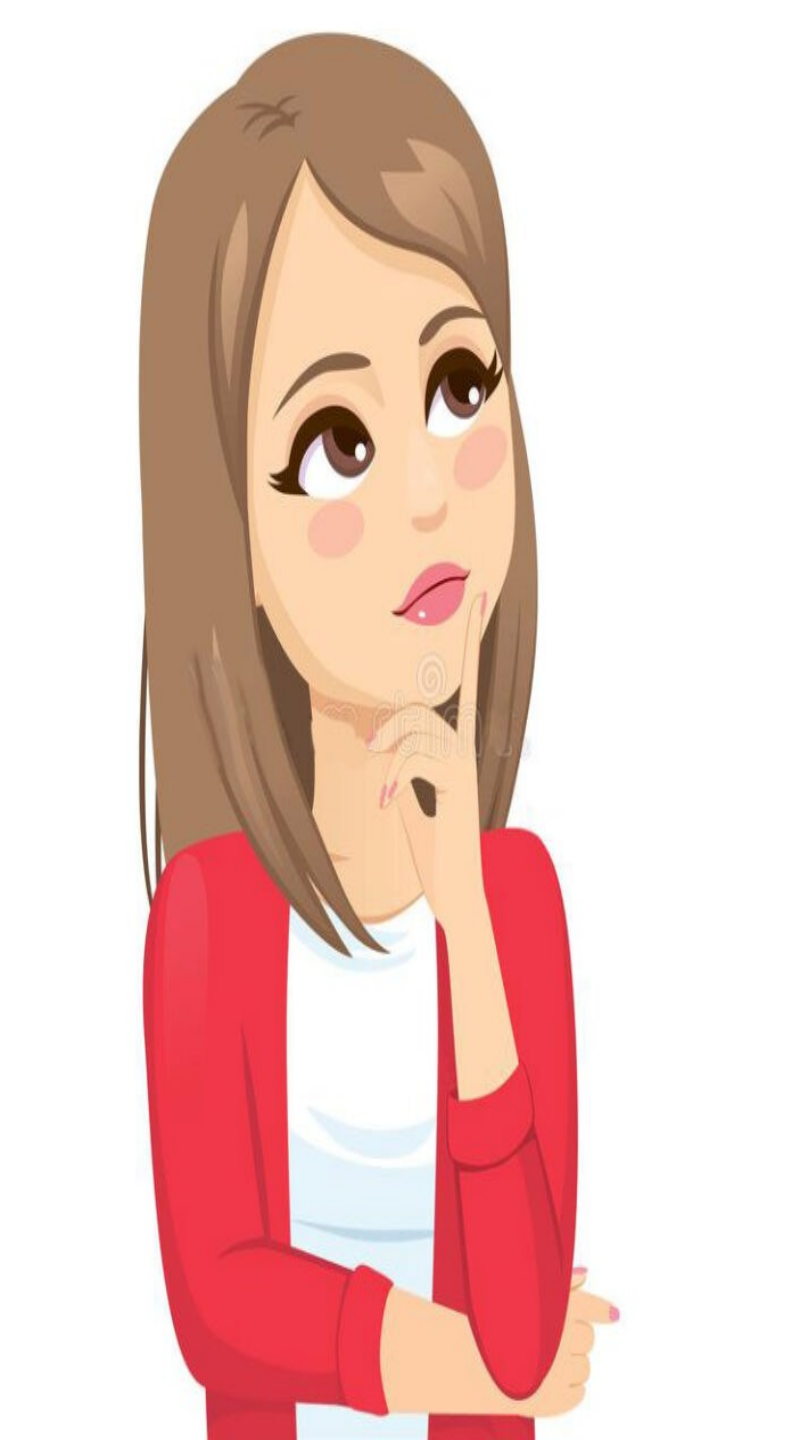
•Main menu-


•Mid-battle scenes with different enemies-


•Boss fight-




•Health and score bars on screen-



## Possible Improvements

- Convert to Python 3 for future-proofing
- Add difficulty levels
- Save high scores
- Add mobile support using Kivy or Pygame CE