# Bad Things in Outer Space : The Game

https://github.com/VoidFyre/BadSpaceGame

Created By : Damyon Olson, Noah Doering, Mohit Singh

Revision: 2.0.0

GDD Template Written by: Benjamin “HeadClot” Stanley

Reformatted by: Brandon Fedie

Written By : Damyon Olson
Edited By : Noah Doering and Mohit Singh

## License

Overview
- ➢ Theme / Setting / Genre
- ➢ Core Gameplay Mechanics Brief
- ➢ Targeted platforms
- ➢ Project Scope
- ➢ Influences
- ➢ The Elevator Pitch
- ➢ Project Description

What sets this project apart?
- ➢ Core Gameplay Mechanics

Story and Gameplay
- ➢ Story
- ➢ Gameplay

Assets Needed
- ➢ 2D
- ➢ Sound
- ➢ Code

Schedule
- ➢ Milestone 1 (March 30th)
- ➢ Milestone 2 (April 12)
- ➢ Final Game Submission

Updates to the Game Design Document
- ➢ Milestone 1 (Update 2.0.0)

# Overview

## ➢ Theme / Setting / Genre

- Bad Things in Outer Space is a Sci-Fi 2D Shooter game that takes place in our vast and ever expanding universe.

## ➢ Core Gameplay Mechanics Brief

- A fully functional and customizable spaceship, including modifications that can be made to alter the stats of the ship.
- Enemies for the player to fight.
- A health bar that will deplete whenever the player takes damage
- Boss battles that will be significantly harder that fighting other enemies in the game.

## ➢ Targeted platforms

- PC

## ➢ Project Scope

- Game Development Timeline
  - Workload and Timeline :
    - Game Design : 1 Week
    - Development Design : 4 Weeks
    - Testing and Bug Fixes : 1 Week
    - Polishing and Finalizing : 1 Week
- Our Team :
  - Mohit Singh
    - Lead Developer
  - Damyon Olson
    - Coordinator, Asset Designer, Developer
  - Noah Doering
    - Lead Asset Designer

## ➢ Influences

- Killer Klowns From Outer Space
  - We took inspiration mainly from the title. Using the absurd wording to give a good idea that our game is meant to be light hearted and fun.
- Sci-Fi Media

        - A space theme means using inspiration from other sci-fi media. We decided to look into other sci-fi type games to help with deciding on how we wanted to implement certain mechanics into our game.
    - Galaga / Asteroids
        - These two games held a lot of influence for our creation of our own game.

## ➢ The Elevator Pitch

An asteroids inspired space-themed shooter with a modular ship that can be upgraded as you progress through waves of enemies, and challenge unique bosses.

## ➢ Project Description

The scope of our project began as rather ambitious, though throughout the development process, it has shown signs of becoming a great and entertaining game that can be enjoyed by anyone. Bad Things in Outer Space : The Game is a Sci-Fi space based shooter game where you control a spaceship with modular configuration capabilities. Soar through the universe and fight waves of enemies that progressively get harder, all while collecting new parts to upgrade your defensive and offensive capabilities. After a certain determined number of waves, the player will encounter bosses who will be significantly harder than the normal enemies that the player encounters on their extra-terrestrial travels.

## What sets this project apart?

- A unique idea merging mechanics seen in popular games, built entirely by hand using Python's Pygame library
- A fun and replayable gameplay loop that progressively gets both easier and harder depending on how upgraded your ship is.

### ➢ Core Gameplay Mechanics

- Player Health
  - The player character is given health, meaning there is an active consequence to putting your character into a dire situation.
  - Every time the player character either runs into an enemy spaceship or gets hit by an enemy projectile, the player character's health and respective health bar will deplete by a certain amount, depending on how much damage is taken. Running into an enemy spaceship is both rewarding and dire, as it will destroy the enemy, but also deal damage to the player character.
- Item System
  - The player character is able to collect items during the game to upgrade their ship as the levels progress.
  - There are 5 rarity levels planned for the items that can be collected by the character. Common, Uncommon, Rare, Epic and Legendary. Each item will fit into a slot within the players inventory and will boost the players stats depending on what the item is. These items can affect stats such as : health, damage, shield, speed, etc.

# Story and Gameplay

### ➢ Story

- You're an explorer traveling the universe in order to map out the universe. Along your journey you encounter waves of alien life forms whose sole purpose is to destroy you. As you delve deeper into the unknown universe, the waves of these attackers continue to grow stronger and more difficult to overcome. Your new goal is to uncover what these attackers may be hiding.

### ➢ Gameplay

- Maneuver the player character through the vastness of outer space. Fight off hordes of alien attackers. Upgrade your ship to take on harder enemies, and find out what the attackers were hiding from you this entire time.

# Assets Needed

- **2D**
  - Textures
    - Character's
      - Player Character
      - Various Different Enemies
      - A handful of bosses
    - Projectiles
      - Differ depending on the enemy, and the upgrades that have been put onto the players ship.
    - UI
      - Title Screen, Pause Screen, Player Inventory

- **Sound**
  - Ambient Sounds
    - Title Theme
    - Game Theme
    - Boss Theme
  - Sound List (Player)
    - Fire Projectile
      - Main Projectile
      - Special Projectile
    - Take Damage
    - Death Sounds
  - Sound List (Enemies)
    - Normal Enemies
      - Fire Projectile
      - Take Damage
      - Death Sound
    - Boss
      - Fire Projectiles
        - Main Projectile
        - Special Projectile
      - Take Damage
      - Death Sound

## ➢ Code

- main.py
  - The code used to execute the game.
- View
  - GameView.py
    - Functions relating to creating a view of the game for the player. This includes menus, the main game, and sound effects
  - MenuView.py
    - Functions relating to UI that are present within the game. This includes the Main Menu, Start Menu, and Credits Menu
  - PowerUpView.py
    - Functions relating to power ups that can be collected by the player throughout the game
  - SpaceshipView.py
    - Functions relating to the spaceship, such as the inventory and drawing the spaceship into the game.
- Model
  - Enemy.py
    - Functions related to the enemies within the game. This includes movement and shooting
  - GameModel.py
    - Functions related to the game as a whole. This includes the update function, which is run within the game loop to constantly update player data and enemy data.
  - GameState.py
    - Functions that load in visuals and updates visuals within the game loop. Some functions include drawing text, and event checking.
  - Laser.py
    - Functions that relate to the main projectiles within the game. Helps with detecting collision between the lasers fired and if they hit enemies or the player.
  - Player.py
    - Functions that help display when a player shoots and drawing the health bar for the player.
  - PowerUp.py
    - Will hold the code for power ups in the game.
  - Projectile.py
    - Functions for the projectiles fired during the game
  - Spaceship.py
    - Functions for drawing the different ships. Includes cooldowns, shooting and laser movement.

- Controller
  - GameController.py
    - Holds functions for controlling the game as a whole. Functions include, collision checking, checking for key presses, running the update script, running the draw scripts, generating random enemies, updating the dashboard, redrawing the window, and the game loop.

# Schedule

- **Milestone 1 (March 30th)**
  - Tasks :
    - Finalize game concepts and mechanics
    - Create level designs for the first three stages
    - Develop player movement and shooting mechanics
    - Create basic enemy AI, eg. movement and shooting
    - Develop a UI
  - Updated Game Design Document :
    - Include finalized game concept and mechanics
    - Include level designs for the first three levels
    - Describe player movement and shooting mechanics
    - Describe basic enemy AI movement and shooting
- **Milestone 2 (April 12)**
  - Tasks:
    - Finish level designs
    - Implement level power ups and different ranked ship upgrades (Common, Uncommon, Rare, Epic, Legendary)
    - Design and implement sound effects and music
    - Create main menu and pause menu screens
    - Conduct playtesting and make necessary adjustments
    - Begin large scale bug testing
    - Begin finalization
  - Updated Game Document :
    - Include all completed level designs
    - Describe power-ups and upgrades
    - Include sound effects and music implementation details
    - Include main menu and pause menu screen design
    - Describe playtesting results and adjustments made based on feedback
    - Begin finalizing game document
- **Final Game Submission**
  - Tasks :
    - Finalize the game
    - Submit the final game and game design document
  - Updated Game Document :
    - Include all updated and finalized game details
    - Describe final game development and submission.

# Updates to the Game Design Document

## ➢ Milestone 1 (Update 2.0.0)

- The Game Design Document was completely revamped using a template found online. This was used to better outline the process we've been going through to ensure everything is organized well.
- The following highlights what was accomplished and what was changed during milestone 1 :
  - **Changes to your game and development design :**
    - Due to overall timeline changes, we have compressed a few of our original ideas to better fit within the timeline. The highlight of our new milestone checks is shown in the Schedule
    - We have begun work on the Ship mechanics as far as basic movement and navigation through the map.
    - Enemies who are able to spawn into the map and shoot at the character to deal damage.
    - A working health bar that depletes every time a projectile hits the player., or when the player collides with an enemy ship.
  - **How has your game evolved since you started working on it?**
    - Our original concept involved the ship moving up and down on a map, shooting horizontally at enemies moving towards the ship. It evolved from that into an omnidirectional shooter, where enemies can attack from all four sides of the map in waves.
  - **An updated version of your project timeline**
    - **Updated Milestones can be seen under the Schedule section of our Game Design Document.**
    - Task 1 : Player Movement and Control
      - Mohit : 80%
        - - Created Controller Input
        - Worked on altering speed of player
        - Projectiles
        - Health Bar
      - Damyon : 20%
        - Slightly altered player movement
        - Worked with Noah on assets for the ship and projectiles.
    - Task 2 : Assets
      - Noah : 80%
        - Created inventory, spaceship, and power up assets to be used in game
      - Damyon : 20%

            - Worked with Noah on the assets used
    - Task 3 : Enemies
        - Mohit : 100%
            - Created enemy movement and shooting mechanics
- **What tasks have been postponed or moved up?**
    - All tasks from the original milestone three have been moved up into milestone 2, since most tasks in milestone three have dealt with the beginning stages of finalization and bug fixing, we felt that these would be safe to move.
- **Any technical challenges you and your team have encountered**
    - Some technical challenges mainly come from the use of GitHub. With three of us working on this project, there have been some occasional misshapes when dealing with pushing and pulling. The best solution for this would be for us to create branches.
    - Other technical challenges include ensuring that we understand the syntax for Pygame. Sometimes working between Recs and Sprites can get confusing/complicated.
- **How will these challenges impact your development timeline?**
    - The timeline of our development hasn't really been impacted due to these challenges. Thankfully, they've been rather easy to solve and work around.
- **Will your final game design need to change?**
    - Our final game design probably won’t need to be changed or altered. The state of the game has remained relatively the same. The only change that I can see that needs to be made is that we've opted to go for a 'free roam' type strategy as opposed to a stationary one. Although this decision is subject to change.