BAD THINGS FROM OUTER SPACE THE GAME

https://github.com/VoidFyre/BadSpaceGame

Demo Video : **https://www.youtube.com/watch?v=nO4ctBXVECw**

Created By : Damyon Olson, Noah Doering, Mohit Singh

Revision: 4.0.0

GDD Template Written by: Benjamin “HeadClot” Stanley

Reformatted by: Brandon Fedie

Written By : Damyon Olson

Edited By : Noah Doering and Mohit Singh

License

Introduction

- **Name of the game**
  - Bad Things in Outer Space : The Game
- **Installation and Playing**
  - Step 1: Proceed to the github link posted at the beginning of this documentation
  - Step 2: Download a copy of the code to your desktop
  - Step 3: Ensure that you have Python 3.11, Pygame 2.3.0, and Pygame-menu 4.4.2 installed
  - Step 4: Navigate to the game folder containing main.exe
  - Step 5: Launch main.exe
- **Controls**
  - Use W A S D to move
  - Left click fires primary weapon
  - Right click fires secondary weapon
  - Escape pauses the game

Overview

- **Theme / Setting / Genre**
  - Bad Things In Outer Space : The Game is an endless shooter game that takes place in outer space.

- **Core Gameplay Mechanics Brief**
  - The player is tasked with destroying many waves of enemies, and dodge their attacks.
  - The enemies have a chance of dropping upgrades for the player to collect to make their ship better.
  - The player and the enemies both have a working health system where when damage is taken the health bar will deplete.

- **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 Galaga inspired space-themed endless shooter arcade game with a modular ship that can be upgraded as you progress through waves of enemies.

Project Description

Bad Things in Outer Space : The Game (BTIOS:TG) is a Galaga inspired endless shooter arcade game where waves of enemies will attempt to destroy the player character. The ship is modular and completely customizable with each upgrade collected by the player. These upgrades will change the ship, weapons, and thruster, giving each a respective boost of health, damage, and speed. The goal of the game is to attempt to beat your high score.

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 the upgrades that are applied to the players ship.

❖ Core Gameplay Mechanics

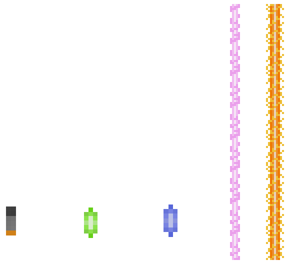
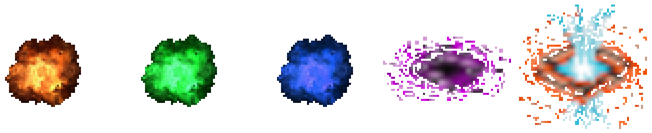
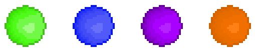
- ➢ Player Health
    - ■ The player character is given health, meaning there is an active consequence to putting your character into a dire situation.
    - ■ Every time the player character either runs into an enemy spaceship or gets hit by an enemy projectile, the player character's health and respective 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 the ship's statistics by bettering the health, speed, and weapon damage.

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
    - Projectiles
      - Differ depending on the enemy, and the upgrades that have been put onto the players ship.
      - Primary (Common, uncommon, rare, epic, legendary):
        - 
      - Secondary (Common, uncommon, rare, epic, legendary):
        -
    - Secondary Projectile Explosions (Common, uncommon, rare, epic, legendary):
      - 
    - UI
      - Title Screen, Pause Screen, Game Over Screen
    - Upgrade Orbs
      - One for each different upgrade level
        - 

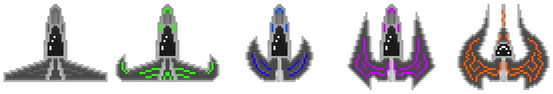
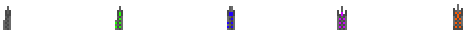
- **Sound**
  - Ambient Sounds
    - Game Theme
  - Sound List
    - Weapon Firing (Primary and Secondary)
    - Death
    - PowerUp Sound
    - Health Sound

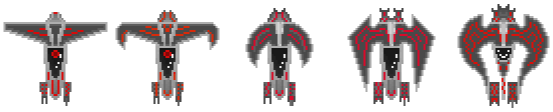
❖ Code

- Note : We are not following the MVC architecture strictly. We tried to follow a different design principle known as Low Coupling High Cohesion to make implementation easier, and allow each of us to be able to work on the project separate from each other and still understand what is happening. LCHC allows for high readability and maintainability.
- main.py
  - The code used to execute the game.
- View
  - Enemy View
    - Code to load in the enemy sprites.
  - Game Over Menu View
    - Shows the game over menu
  - Game View
    - Display setup for the game (window name, size, etc)
  - Main Menu View
    - All of the information regarding the main menu
  - Moving Background View
    - Moves the background during the game
  - Pause Menu View
    - Pause menu for the game
  - Spaceship View
    - Creates the view for the spaceships in the game
- Model
  - Enemy
    - Contains the class and helper functions for enemy mechanics
  - Explosion
    - Contains the class and helper functions for explosions
  - Game Model
    - Contains the update function for everything within the game
  - Game State
    - Contains the class and helper functions for memorizing the game state. When the game is paused or restarted, this class remembers everything it needs to know to make sure it does these things correctly.
  - Health Aid
    - Contains the class and helper functions for the health packs
  - Laser
    - Contains the class and helper functions for the projectiles
  - Player
    - Contains the class and helper functions for the player character
  - Spaceship

        - Contains the class and helper functions for each spaceship in the game
    - Upgrade
        - Contains the class and helper functions for the upgrades in the game
    - Save Stats
        - Saves information regarding the best score player has achieved and the best kill count player has achieved
- Controller
    - Game Controller
        - Contains the class and functions to keep the game running

Characters

- **Player Character**
  - 
    - These ships represent the Starter Ship (Common), as well as Uncommon, Rare, Epic, and Legendary variants of the ships
    - The player can customize these ships using the following:
      - Primary Weaponry:
        
        -
      - Secondary Weaponry:

        -
      - Thrusters:
        -
- **Enemies**
  - 
    - These ships represent the common, uncommon, rare, epic, and legendary enemies a player may encounter during the game.
    - Each enemy varies in strength and power, adding a challenging aspect when a character happens to encounter a more difficult enemy.

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.
        - We’ve had some issues with audio files not working when trying to code them to cue during certain actions.
        - Occasionally the absolute path to find files within the game wouldn’t work, and we’d have to troubleshoot and use different python resources to be able to maneuver through and grab those assets.
    - **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.

- **Milestone 2 (Update 3.0.0)**
    - The game document itself stayed mostly the same, aside from a couple of changes to include pictures of the player characters and some enemies that could be encountered during the run.
    - The following highlights what was accomplished and what was changed during milestone 2 :
        - **Changes to your game and development design :**
            - For this milestone, we have implemented the GUI for the game, as well as some new assets to better fit the aesthetic of the game.

- We’ve also opted to go for an endless shooter as opposed to a level system. The player will now encounter ‘waves’ of enemies rather than pre-specified enemies. These enemies will get progressively harder.

- **How has your game evolved since you started working on it?**
  - Since our original concept, and our current changes, it hasn't evolved too much. Especially since we re-decided to go for an endless shooter. Our hope is that we will still be able to implement some sort of modular system for the ship to allow for an easier time fighting harder enemies.
- **An updated version of your project timeline**
  - **Updated Milestones can be seen under the Schedule section of our Game Design Document.**
  - Task 1 : GUI Implementation
    - Mohit : 80%
      - Created the base code for adding a start menu and pause menu to the game
      - Also implemented a moving background.
    - Damyon : 20%
      - Created a simple design using Figma for an implementation of the different menus.
  - Task 2 : Assets
    - Noah : 80%
      - Finished some assets to be used in the game including the inventory system and a few of the extra elements that will be implemented into the game as far as character design.
        - Multiple Tiers of Ship
        - Multiple Tiers of Bullets
        - Multiple Tiers of Thrusters
        - Multiple Tiers of Weapons
        - Multiple Tiers of Weapons
    - Damyon : 20%
      - Worked closely with Noah to help design and provide creative insight on the design of certain aspects of the assets.
        - Color’s for rarities, small design changes
- **What tasks have been postponed or moved up?**
  - As of now, the remaining tasks that need to be implemented are mainly polishing and implementing the modular system. Both of which shouldn’t be too difficult to put into the game to make it complete. If we have time, we would like to implement some sort of Boss battle system to add an extra layer of challenge to the game. This will likely be a single boss that shows up after the player completes 5/10 waves.

        - We hope to allow the boss to have a higher chance of dropping a more powerful upgrade for the player ship.
      - **Any technical challenges you and your team have encountered**
        - We've had an issue with pausing the game where the player character and enemies are still able to shoot and move after the player dies.
        - This likely also happens when the game is paused.
      - **How will these challenges impact your development timeline?**
        - It shouldn't have an impact on our development timeline, as these challenges should be simple to overcome within the future.
      - **Will your final game design need to change?**
        - The final game design won't need to change.
- **Final Submission**
  - **Game Design**
    - Mechanics/Technology
      - What is the gameplay loop of your game?
        - There is an infinite replayability gameplay loop within our game. Because it is an endless shooter, the idea is that the player will play once, see their score, and play the game over again in an attempt to beat it. It's not storing this score, but it is showing the player as a way to motivate them to continue playing so that they can beat their previous score.
      - What are the core mechanics of your game, and how do they contribute to the gameplay loop?
        - The core mechanics to our gameplay is the ability to fire weapons to destroy, the ability to upgrade your ship, and the ability to regain health while fighting. These mechanics are simple enough to catch a player's attention, and fun enough to keep them playing.
      - What is your game's gimmick and how does it contribute to your game?
        - The main gimmick of our game is the ability to upgrade the player characters ship. After destroying an enemy, there is a chance that the enemy will drop an upgrade orb. The player is able to pick up this orb, and their ship will change. The changes will alter either the ship itself, primary weapon, secondary weapon or the thruster.
        - If the player character already has a high level upgrade, it will not be replaced with a lower one, only a more powerful one.
        - This gets rid of frustration when grabbing items, and allows the ship to become more powerful as waves progress.
      - How does your game differ from other games in the same genre?
        - We give the player a working health bar, as well as the random upgrade system, and a health regen system. These, at least as

far as we are aware, create a unique gameplay experience that differs from other games in our genre.

- Player Experience
    - What emotions do you want the player to experience while playing your game?
        - We want our players to get just the right amount of excitement and frustration out of playing the game. Not so much frustration to where the game becomes unplayable, but enough to push the player to want to try again.
    - What kind of challenges will the player face, and how will they be overcome?
        - The player will face waves of enemies that progressively get more difficult. Once the player kills these enemies, there is a chance for an upgrade to fall, this will allow the player to overcome the challenge by making their ship more powerful.
    - What kind of rewards will the player receive for progressing through the game?
        - Rewards are given in the form of loot drops from enemies. Although there isn't a 100% chance that the player will be rewarded, there is always a chance for one to drop.
    - What kind of feedback will the player receive while playing the game?
        - After reaching the 'Game Over' screen, the player will be shown their kill count and the score they have accumulated during that playthrough.
    - What kind of audio and visual elements will enhance the player's experience?
        - We've added explosion animations for when the enemy ships are destroyed, stylized towards the different tier of weapon used.
        - We've also outsourced some sounds from a free website to bring music, weapon firing, death, and power up sounds.

Game Design Changes

- What was the original design and concept when you proposed the game?
    - The original concept implemented a much more customizable ship system, where the player could equip and unequip the ships items at will. However, due to time constraints we decided to go with a much more straightforward approach, where the player will collect different upgrades from destroyed ships, and these upgrades will automatically attach to the ship.
- How and why did the design change over time?
    - Our design changed over time based mainly on what we felt like we had time to do and implement into the game. The biggest changes that were

made had to do with time constraints, and wanting to keep the game simple yet enjoyable.

- What was your original plan for the game mechanics? How and why have they changed?
    - Our original plan for the game mechanics showed the player ship being able to traverse omni-directionally through space. This mechanic was changed to be a one directional travel, where the player ship is moving up, and the enemy ships are moving down towards the player. The reason we changed this was to keep the game complex enough to still be a challenge, but simple enough so that we wouldn't run out of time while working on the project.
- What was your original plan for the game gimmick? How and why has it changed?
    - Our original plan for the game gimmick had higher customizability for the ship. The player would be able to pick and choose what upgrades they were able to apply. There were also a couple more upgrade options that we opted out of using. We instead went with a system that would automatically equip upgrades to the player when collected, changing the ship visually, but not allowing the player to change it themselves. Similarly to our reasoning before, we chose to make these changes mostly for the sake of simplicity to ensure the game would be turned in on time.

Game Development and Documentation

- Codebase can be found in the documentation here : Code
- Are there any major bugs or flaws in your game we should be aware of? (Undocumented issues/bugs will result in a score deduction)
    - Sometimes the health of the enemy ships will overlap with the ship itself, although we do not consider this a bug.
    - When the player kills an enemy, the laser that was shot by the enemy is also destroyed.
- What tools did you use to facilitate collaboration or code versioning (e.g. git/github, VSCode Liveshare, etc.)
    - For the duration of this collaborative project, we used GitHub to share our code versioning.

Group Member Roles, Tasks, and Performance

- Mohit Singh
    - Programming - 80%
        - Spent majority of his time programming the game
        - Implemented a good bit of functionality regarding player movement, enemy spawn rates, health spawning, and projectile firing for both player and enemies.
    - Assets - 10%

- Spent some of his time finding assets for the game including background and some sounds.
- Game Design Document - 10%
  - Spent some of his time on the Game Design Document, helping edit and make changes prior to the release of the document.

- Noah Doering
  - Programming - 40%
    - Spent some of his time programming endgame functionality into the game, such as explosions doing damage to enemies, and altering enemy spawn rates.
  - Assets - 50%
    - Spent most of his time designing sprites to be used in the game. Created every player, enemy, projectile and explosion.
  - Game Design Document - 10%
    - Spent some of his time helping edit and make changes to the GDD prior to the release of the document.
- Damyon Olson
  - Programming - 40%
    - Spent some of his time programming the power up functionality into the game. Finished work on the power up system, loot box chances, and the collection of the powerups.
  - Assets - 10%
    - Spent some of his time creating sprites and finding sound for the game. Finished the upgrade orb sprites as well as the health pack sprite.
  - Game Design Document - 50%
    - Spent most of his time creating and perfecting the game design document for submission.
- Project Timeline
  - Milestone 1
    - Created basic movement and basic enemy AI so that the player could move and shoot at enemies on screen.
  - Milestone 2
    - Added a menu system to the game, as well as finalizing most assets used for players and enemies
  - Final Submission
    - Added power ups and health, as well as perfecting some of the rougher edges of the program prior to submission.