

game technology documentation

By Rahul Venkateswaran



April 27, 2025

Games technology module

Documentation Document

**Introductory Overview**

**GitHub Repository Link:** [**https://github.com/Rahul-Venkateswaran12/Asteroids.git**](https://github.com/Rahul-Venkateswaran12/Asteroids.git)

**This document outlines the development of my Asteroids game for the IN2026 ReSIT Coursework, a modern take on the classic arcade game built using C++ and OpenGL with GLUT. I extended the provided codebase to implement a start screen with a navigable menu, gameplay instructions, a high-score system with name input, and an extra lives power-up, fulfilling Part I (option a) and partially fulfilling Part II (one of three required power-ups). The document details all implemented features, codebase changes, implementation explanations with code snippets, errors encountered, and resolutions. My goal was to deliver a functional game with a polished UI, robust gameplay, and a seamless power-up system, while addressing technical challenges like incorrect spaceship respawning.**

**The project was developed in Visual Studio 2022 on Windows, using sprites (spaceship\_fs.png, asteroid1\_fs.png, explosion\_fs.png, extralife\_fs.png) from the Moodle Filmstrips archive. An external sprite was used for the extra lives powerup: extralife\_fs.png. The codebase was managed with Git, with commits tracking feature development and bug fixes, meeting the version control requirement.**

**Part I: Start Screen, Menu, Instructions, and High-Score System**

**1. Changes to Methods, New Methods, New Classes, and Source Files**

**Source Files Modified:**

* **Asteroids.cpp: Main game logic, UI creation, menu navigation, high-score handling, gameplay, and power-up spawning.**
* **Asteroids.h: Class declaration, added OnLifeGained for power-up support.**
* **GUILabel.cpp: Font rendering and alignment, added FONT\_HELVETICA\_18 support.**
* **GUILabel.h: Added FONT\_HELVETICA\_18 enum value.**
* **Player.h: Modified to support extra lives with separate life increment logic.**
* **IPlayerListener.h: Added OnLifeGained for power-up listener interface.**
* **ExtraLife.cpp: Implemented extra lives power-up logic.**
* **ExtraLife.h: Declared ExtraLife class.**

**New Methods:**

* **Player::IncrementLife (Player.h): Increments lives and fires FireLifeGained.**
* **Player::FireLifeGained (Player.h): Notifies listeners of life gain.**
* **Player::ResetLives(Player.h): Resets life count**
* **Asteroids::OnLifeGained (Asteroids.h, Asteroids.cpp): Updates lives display for extra lives.**
* **Asteroids::CreateExtraLife (Asteroids.cpp): Spawns an ExtraLife object.**
* **Asteroids::HasExtraLife (Asteroids.cpp): Checks if an ExtraLife exists.**

**Modified Methods:**

* **Asteroids::CreateGUI (Asteroids.cpp): Added labels for title, menu, instructions, high scores, and name input; adjusted alignments.**
* **Asteroids::Start (Asteroids.cpp): Toggled UI visibility; added power-up spawn timer.**
* **Asteroids::OnKeyPressed (Asteroids.cpp): Implemented menu selection, instructions/high-score exit, and name input.**
* **Asteroids::OnSpecialKeyPressed (Asteroids.cpp): Added up/down arrow menu navigation.**
* **Asteroids::StartGame (Asteroids.cpp): Hid menu UI, showed score/lives, spawned spaceship, started power-up timer and reset lives.**
* **Asteroids::OnTimer (Asteroids.cpp): Handled game over, level progression, and power-up spawning.**
* **Asteroids::OnPlayerKilled (Asteroids.cpp): Updated for life decrements only.**
* **GUILabel::GUILabel (GUILabel.cpp): Added FONT\_HELVETICA\_18 support with tuned mFontWidth.**

**2. Feature Implementation and Code Snippets**

**Start Screen with Menu:**

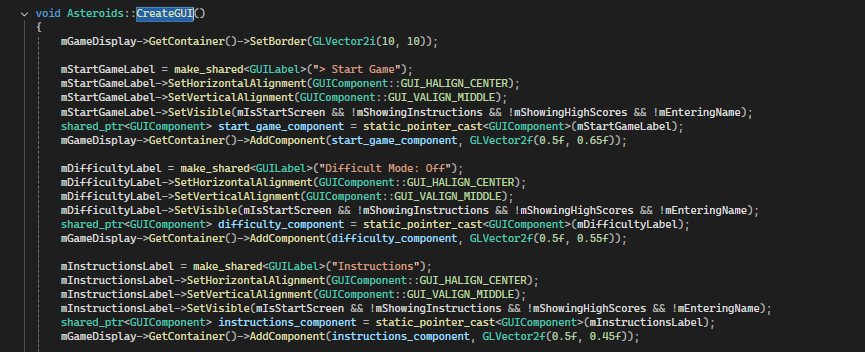
* **Description:**

**A start screen displays a centered “ASTEROIDS” title (FONT\_HELVETICA\_18) and four menu options: “Start Game”, “Difficult Mode” (toggles power-ups), “Instructions”, and “High Scores”. Players navigate using up/down arrows to highlight options and press spacebar to select. “Difficult Mode” toggles mEnablePowerups, disabling extra lives when enabled. Asteroids float in the background (via CreateAsteroids) without a spaceship or score/lives until StartGame. On game start, score/lives labels appear, and the spaceship spawns.**

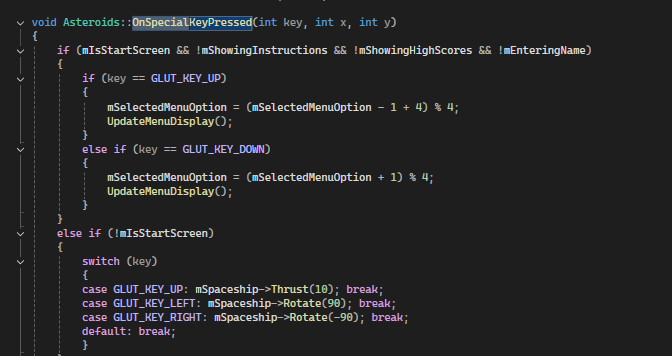
* **Implementation:**

**Created labels in CreateGUI with GUI\_HALIGN\_CENTER for horizontal centering and y-positions (title at 0.85, menu at 0.65 to 0.35) for hierarchy. Handled navigation in OnSpecialKeyPressed and selection in OnKeyPressed. Used UpdateMenuDisplay to add “>” prefix to the selected option. Ensured StartGame hides menu UI and spawns the spaceship.**

**Code Snippet (Asteroids.cpp, CreateGUI):**

****

**Code Snippet (Asteroids.cpp, OnSpecialKeyPressed):**

****

**Instructions Screen:**

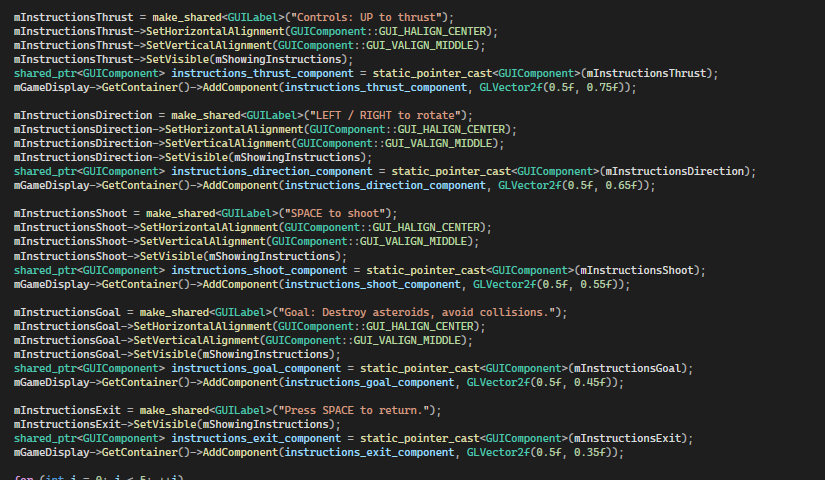
* **Description:**

**Accessible via the menu, displays controls (“UP to thrust”, “LEFT/RIGHT to rotate”, “SPACE to shoot”), objective (“Destroy asteroids, avoid collisions”), and a “Press SPACE to return” prompt. Labels are centered horizontally and aligned vertically (y = 0.75 to 0.35).**

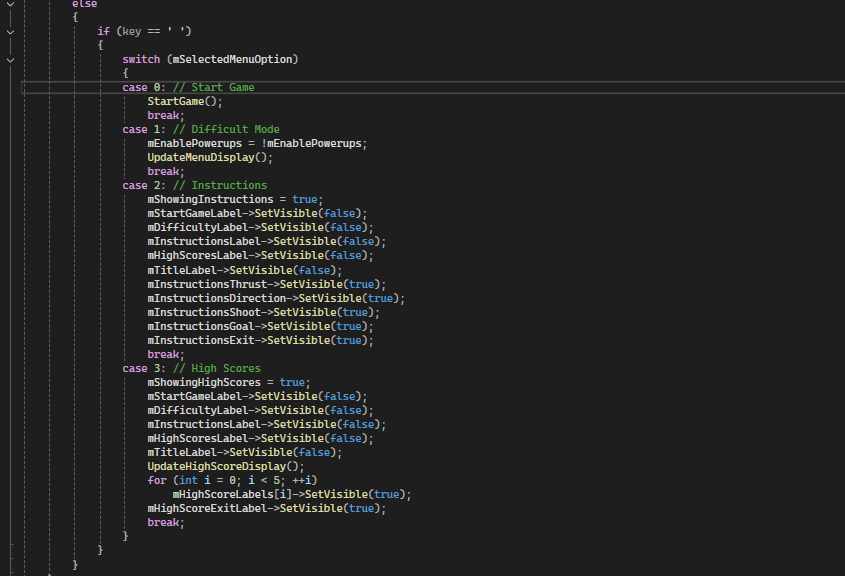
* **Implementation:**

**Added five GUILabel objects in CreateGUI with y-positions spaced at 0.1 intervals. Toggled visibility in Start and OnKeyPressed to show/hide based on menu selection. Used GUI\_HALIGN\_CENTER for alignment.**

**Code Snippet (Asteroids.cpp, CreateGUI):**

****

**Code Snippet (Asteroids.cpp, OnKeyPressed):**

****

**High-Score System:**

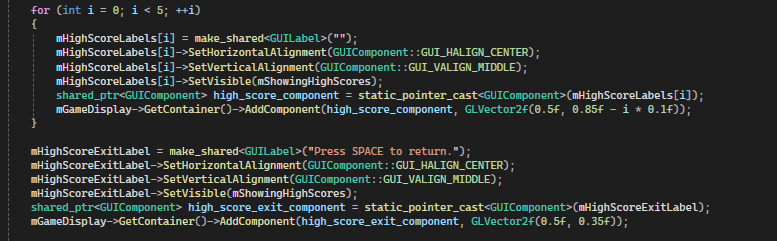
* **Description:**

**Displays top 5 scores with player names from highscores.txt. After game over, players enter a name (up to 8 characters, uppercase/numbers) to save their score. The table aligns vertically with instructions (y = 0.85 to 0.45, exit at y = 0.35).**

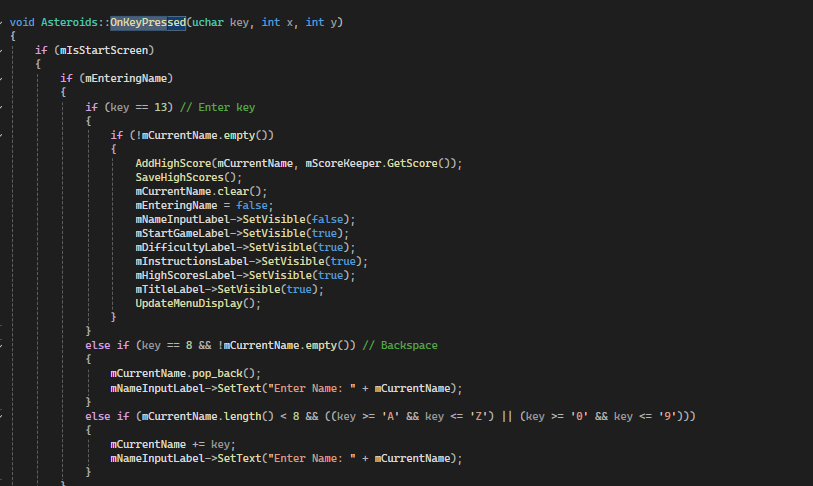
* **Implementation:**

**Created five labels in CreateGUI for scores, used LoadHighScores and SaveHighScores for file I/O, and handled name input in OnKeyPressed. Sorted scores in AddHighScore and updated display in UpdateHighScoreDisplay.**

**Code Snippet (Asteroids.cpp, CreateGUI):**

****

**Code Snippet (Asteroids.cpp, OnKeyPressed):**

****

**UI Enhancements:**

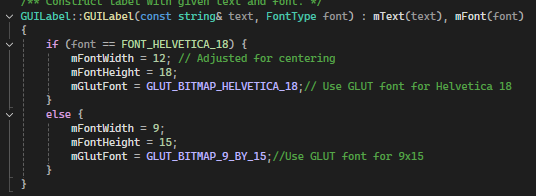
* **Description:**

**Features a larger “ASTEROIDS” title (FONT\_HELVETICA\_18), high-score table aligned with instructions (y = 0.85 to 0.45), and centered UI elements. Score/lives labels are visible only during gameplay.**

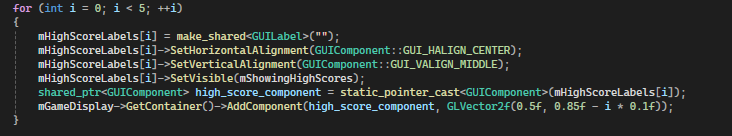
* **Implementation:**

**Modified GUILabel.cpp to support FONT\_HELVETICA\_18 with tuned mFontWidth. Adjusted y-positions in CreateGUI for alignment. Used GUI\_HALIGN\_CENTER for all labels.**

**Code Snippet (GUILabel.cpp, GUILabel constructor):**

****

**Code Snippet (Asteroids.cpp, CreateGUI, high-score exit):**

****

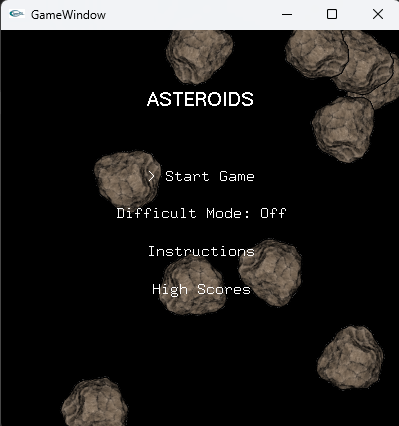
**3. Errors Faced and Resolutions**

* **Error: Menu options didn’t highlight correctly.**
  + **Cause: UpdateMenuDisplay didn’t update labels with “>” prefix.**
  + **Resolution: Modified UpdateMenuDisplay to set text dynamically, e.g., mStartGameLabel->SetText(mSelectedMenuOption == 0 ? "> Start Game" : "Start Game");.**
* **Error: Title was off-center horizontally.**
  + **Cause: mFontWidth = 18 for FONT\_HELVETICA\_18 caused misalignment.**
  + **Resolution: Adjusted mFontWidth to 12 in GUILabel.cpp.**
* **Error: Instructions were not displayed due to width/height limits**
  + **Cause: Initial y-positions were too close.**
  + **Resolution: Set y = 0.75 to 0.35 with 0.1 spacing in CreateGUI.**
* **Error: High-score table was misaligned vertically.**
  + **Cause: Initial y = 0.65 to 0.15 didn’t match instructions’ y = 0.75 to 0.35.**
  + **Resolution: Shifted to y = 0.85 to 0.45, exit to y = 0.35 in CreateGUI.**
* **Error: Scores didn’t save correctly.**
  + **Cause: File I/O formatting issues in SaveHighScores.**
  + **Resolution: Used file << entry.first << " " << entry.second << "\n";.**
* **Error: Spaceship didn’t appear after game start.**
  + **Cause: CreateSpaceship wasn’t called in StartGame.**
  + **Resolution: Added mGameWorld->AddObject(CreateSpaceship()); in StartGame.**
* **Error: Lives not resetting to 3 on game start**
  + **Cause: If player plays again after dying 3 times, lives never is reset to 3 and stays on 0**
  + **Resolution: create a reset lives function in Player.h which is called when the game is started.**

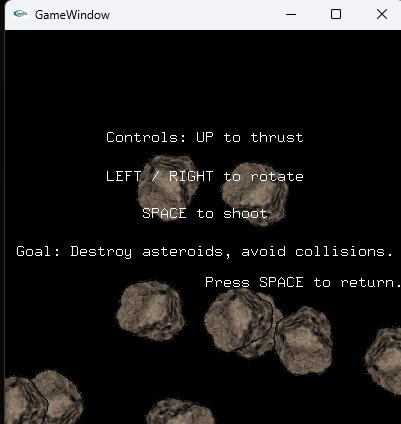
**4. Part 1 Conclusion**

**Part I was fully implemented with a start screen, navigable menu, instructions screen, high-score system, and UI enhancements, meeting all requirements (option a). The UI is polished with centred elements and aligned layouts, and gameplay integrates seamlessly. Errors like UI misalignment, input issues, and file I/O were resolved through iterative testing. The features are robust and visually appealing.**

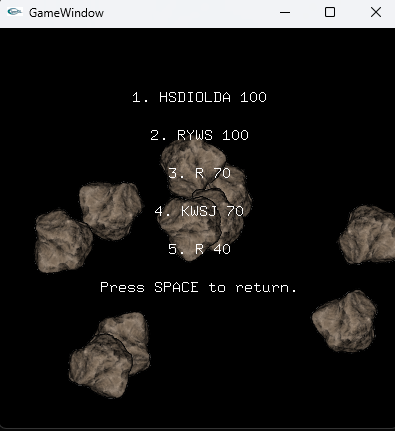
**5. Screenshots of Gameplay  
Main Menu:**

****

**Instructions:**

****

**Highscore Table:**

****

**HighScore Input:  
**

**Part II: Extra Lives Power-Up**

**1. Changes to Methods, New Methods, New Classes, and Source Files**

***(See Part I for shared changes; below are Part II-specific changes)***

**Source Files Modified:**

* **Asteroids.cpp: Added power-up spawning and OnLifeGained.**
* **Asteroids.h: Declared OnLifeGained.**
* **Player.h: Added IncrementLife and FireLifeGained.**
* **IPlayerListener.h: Added OnLifeGained.**
* **ExtraLife.cpp: Implemented power-up collision.**
* **ExtraLife.h: Declared ExtraLife.**

**New Classes:**

* **ExtraLife (ExtraLife.h, ExtraLife.cpp): Manages heart power-up.**

**New Methods**

* **Player::IncrementLife (Player.h): Increments mLives.**
* **Player::FireLifeGained (Player.h): Notifies listeners.**
* **Asteroids::OnLifeGained (Asteroids.h, Asteroids.cpp): Updates lives display.**
* **Asteroids::CreateExtraLife (Asteroids.cpp): Spawns ExtraLife.**
* **Asteroids::HasExtraLife (Asteroids.cpp): Checks for ExtraLife.**

**Modified Methods:**

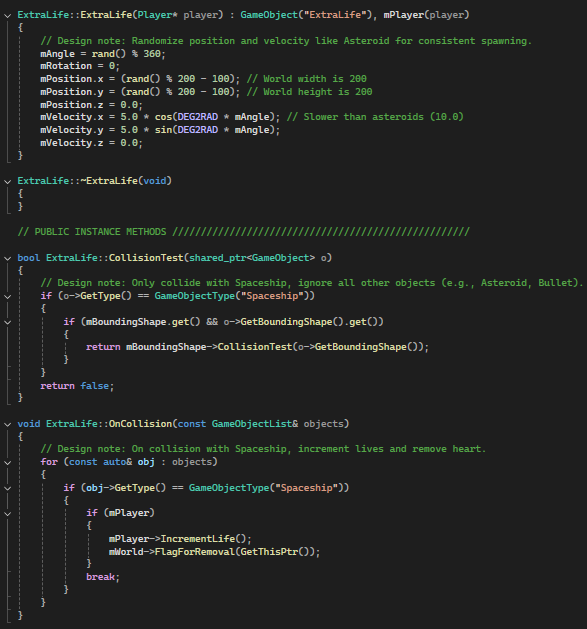
* **Asteroids::Start (Asteroids.cpp): Added power-up timer.**
* **Asteroids::StartGame (Asteroids.cpp): Initialized power-up timer.**
* **Asteroids::OnTimer (Asteroids.cpp): Added SPAWN\_EXTRA\_LIFE.**
* **Asteroids::OnPlayerKilled (Asteroids.cpp): Handled life decrements.**
* **ExtraLife::OnCollision (ExtraLife.cpp): Calls IncrementLife.**

**2. Feature Implementation and Code Snippets**

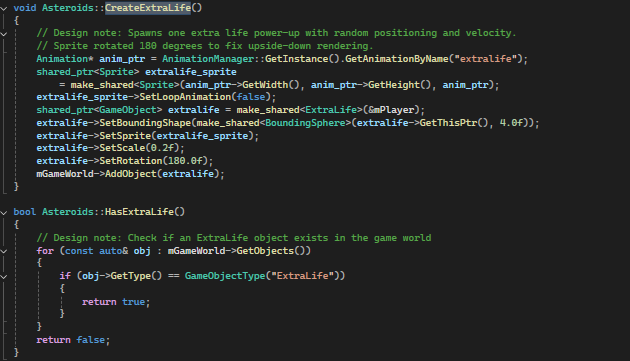
**Extra Lives Power-Up:**

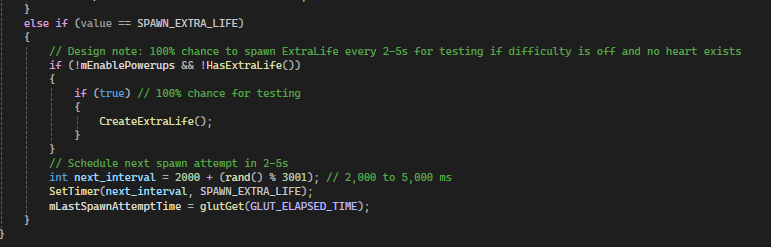
* **Description:** 
  + **A heart power-up (extralife\_fs.png, 32x32) spawns randomly when “Difficult Mode” is off, every 2–5s with 100% probability (for testing; intended 10% chance, 10–20s). On spaceship collision, it increments lives, removes itself, and allows the spaceship to continue moving without respawning. No explosion occurs. Only one heart exists (checked via HasExtraLife). It moves with random velocity (5.0 units/s) and position (±100 units). Disabled when mEnablePowerups = true.**
* **Implementation:** 
  + **Created ExtraLife class with random position/velocity in ExtraLife.cpp. In Asteroids::CreateExtraLife, spawned hearts with a scaled sprite (0.1f), rotated 180° to fix orientation. Handled collisions in ExtraLife::CollisionTest (spaceship only) and OnCollision (calls IncrementLife). Used Player::IncrementLife and FireLifeGained to update lives via Asteroids::OnLifeGained. Added spawn logic in OnTimer and HasExtraLife to ensure single hearts.**

**Code Snippet (ExtraLife.cpp, constructor and OnCollision):**

****

**Code Snippet (Asteroids.cpp, CreateExtraLife and OnTimer):**

****

****

**3. Errors Faced and Resolutions**

* **Error: Spaceship respawned at center after collecting a heart.**
  + **Cause: Player::AddLife called FirePlayerKilled, triggering Asteroids::OnPlayerKilled to schedule a respawn.**
  + **Resolution: Replaced AddLife with IncrementLife and FireLifeGained in Player.h; added OnLifeGained in Asteroids.cpp to update lives without respawning.**
* **Error: Build error (C2509) for undeclared OnLifeGained.**
  + **Cause: Asteroids.h lacked OnLifeGained declaration.**
  + **Resolution: Added void OnLifeGained(int lives\_left) override; to Asteroids.h.**
* **Error: Heart sprite was upside-down.**
  + **Cause: Sprite orientation in extralife\_fs.png.**
  + **Resolution: Set extralife->SetRotation(180.0f) in CreateExtraLife.**

**4. Features Not Implemented**

* **Two additional power-ups (e.g., spiral bullets, invulnerability) due to time constraints and errors late into development.**

**5. Conclusion**

**Part II was partially completed with an extra life power-up which spawns randomly, increments lives, and integrates with “Difficult Mode”. A critical respawn bug and build error were fixed, ensuring functionality. Time constraints prevented two additional power-ups or visual feedback.**

**6. Screenshots of Gameplay:  
Bonus Life Powerup:  
**

**Overall Conclusion**

**The Asteroids game fulfills Part I (start screen, menu, instructions, high scores) and partially fulfills Part II (extra lives power-up). It features a polished UI, robust gameplay, and effective bug fixes. Challenges like UI alignment, input handling, and power-up bugs were resolved through testing. Time constraints limited additional power-ups, but the game is functional and appealing. The Git repository tracks all changes, and the codebase was cleaned (.vs folder removed, project cleaned) for submission.**