

SAADIA ASGHAR

FATIMA ZAHRA

FATIMA TABASSUM

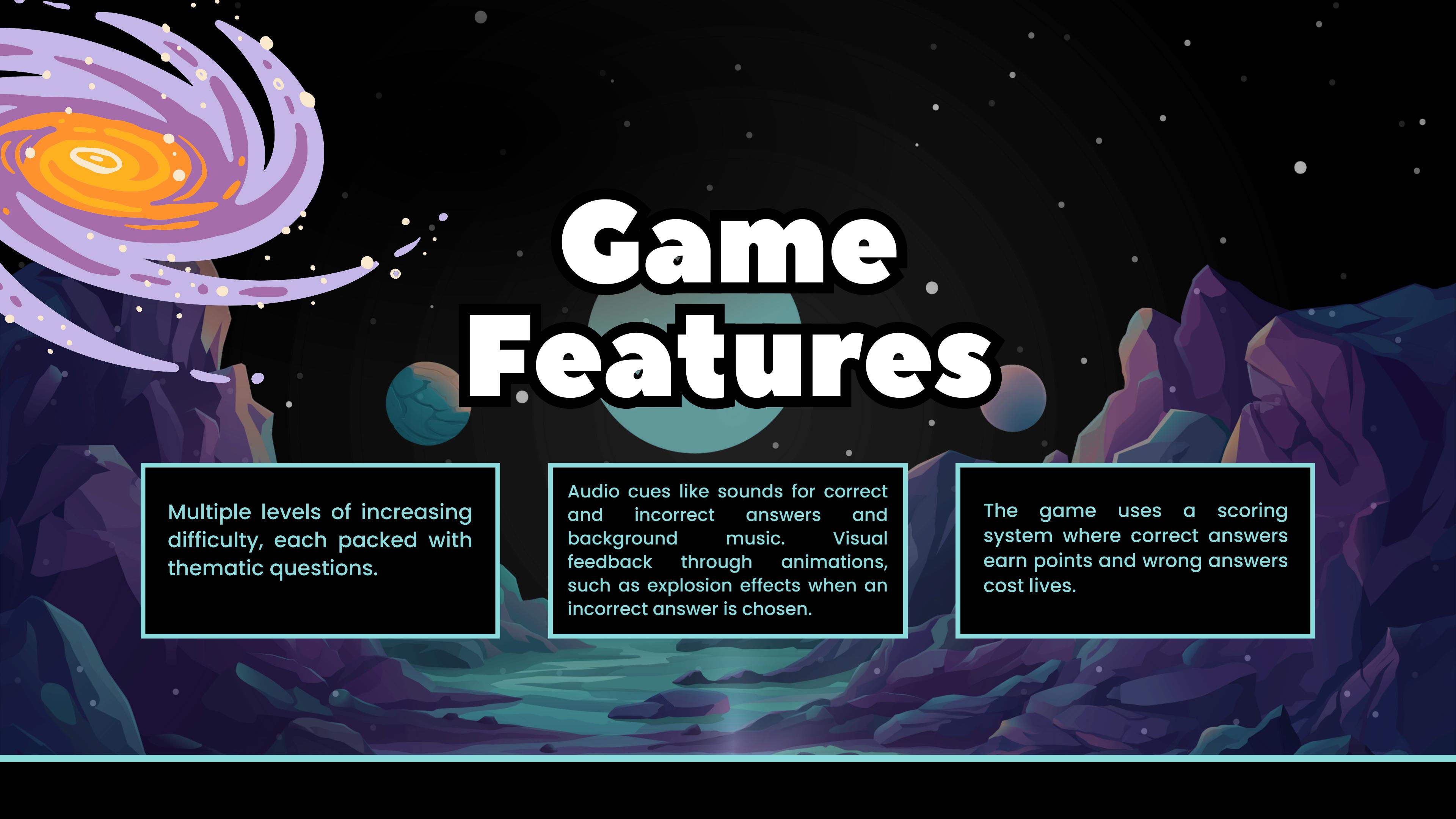
Master Code. Conquer Levels.
Be the Blaster !

CODE BLASTER

Code Blaster: A PROGAMMING Quiz Game

Code Blaster creatively combines the interactive gameplay mechanics of a space shooter with an educational C++ quiz system. This unique blend aims to attract users by making programming education both fun and engaging. Players not only answer questions but also enjoy dynamic game elements such as shooting effects and animations, providing a captivating way to learn C++ concepts through gameplay.





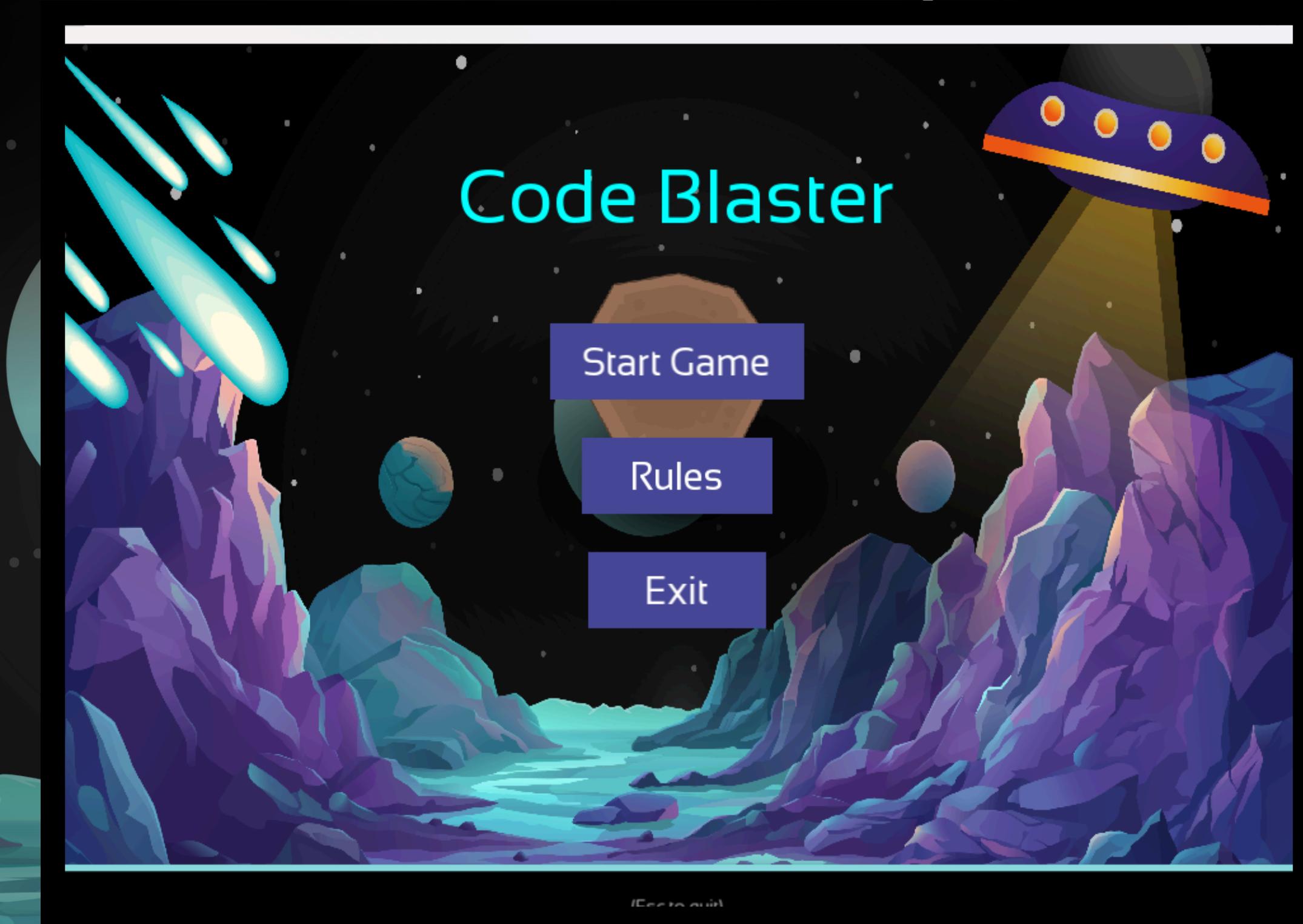
Game Features

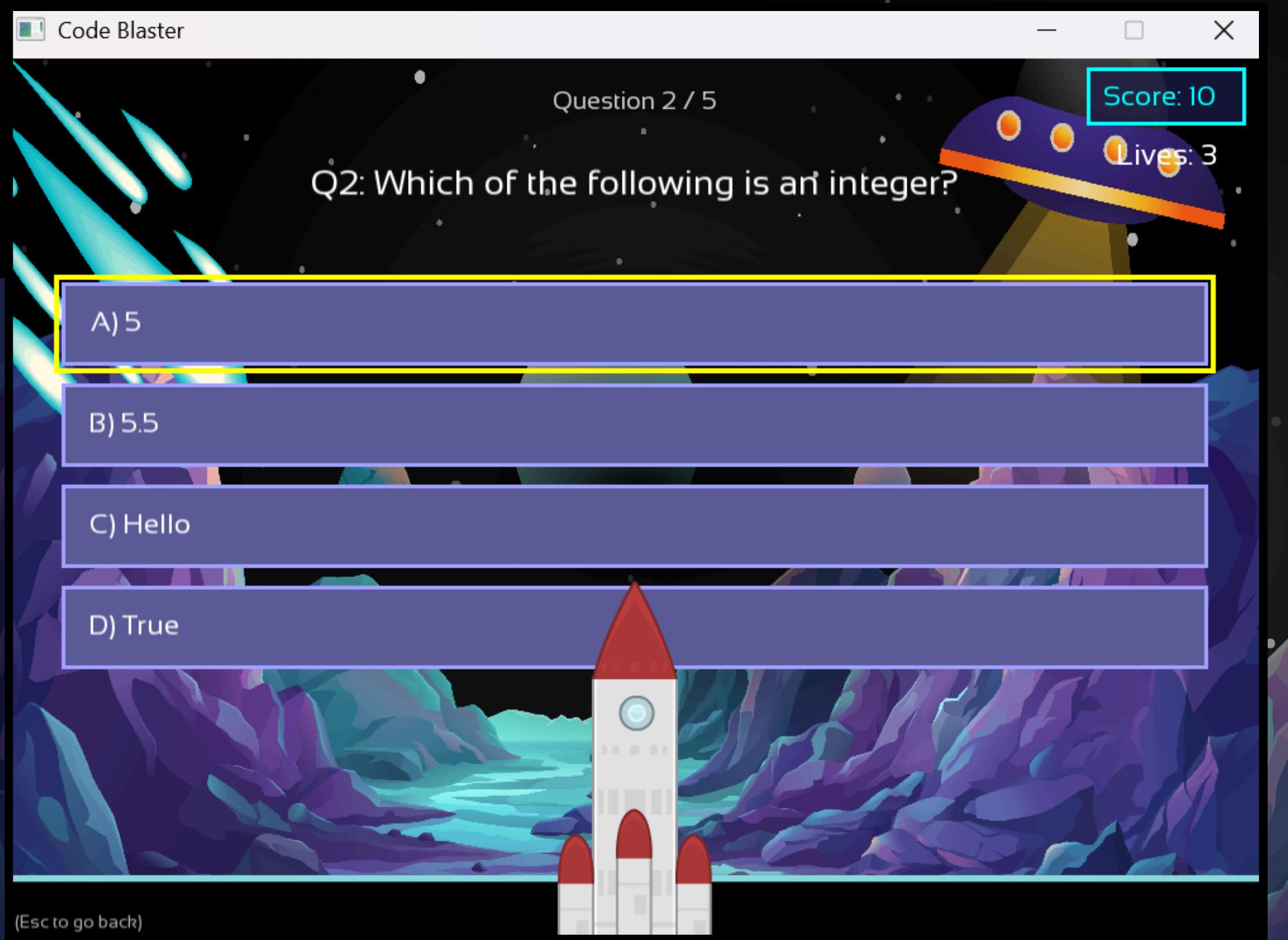
Multiple levels of increasing difficulty, each packed with thematic questions.

Audio cues like sounds for correct and incorrect answers and background music. Visual feedback through animations, such as explosion effects when an incorrect answer is chosen.

The game uses a scoring system where correct answers earn points and wrong answers cost lives.

GUI



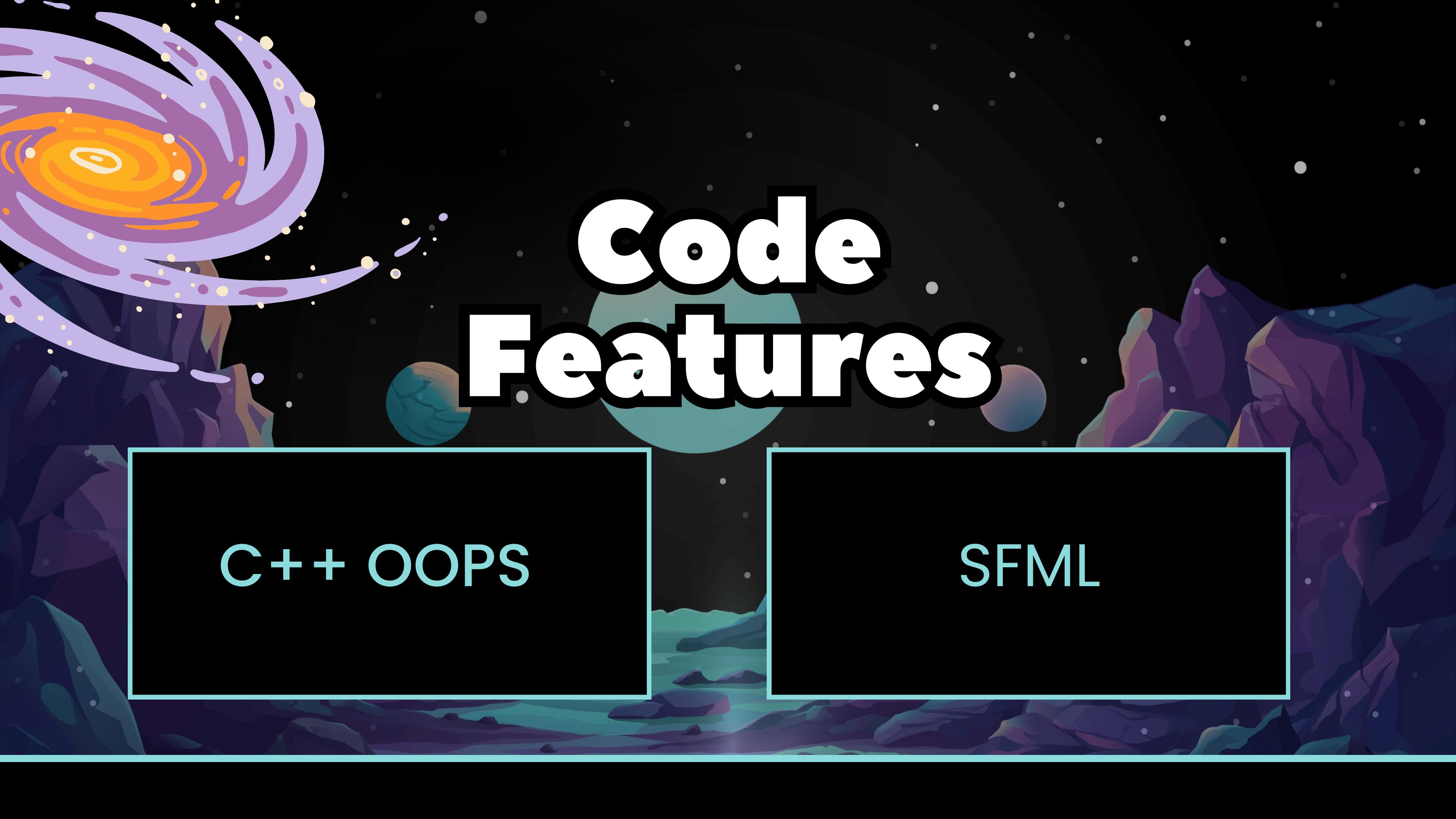


GUI



GAME PLAY

- The gameplay starts at a menu where the player selects a level.
- Within each level, players answer questions by clicking on the options.
- Correct answers reward points and allow the player to continue, while incorrect answers deduct lives.
- Losing all lives results in a game over screen.
- Upon completing all questions in a level, the next level unlocks if the player still has lives remaining.
- There are visual highlights for selected answers and feedback messages to guide the player.



Code Features

C++ OOPS

SFML

OOPS

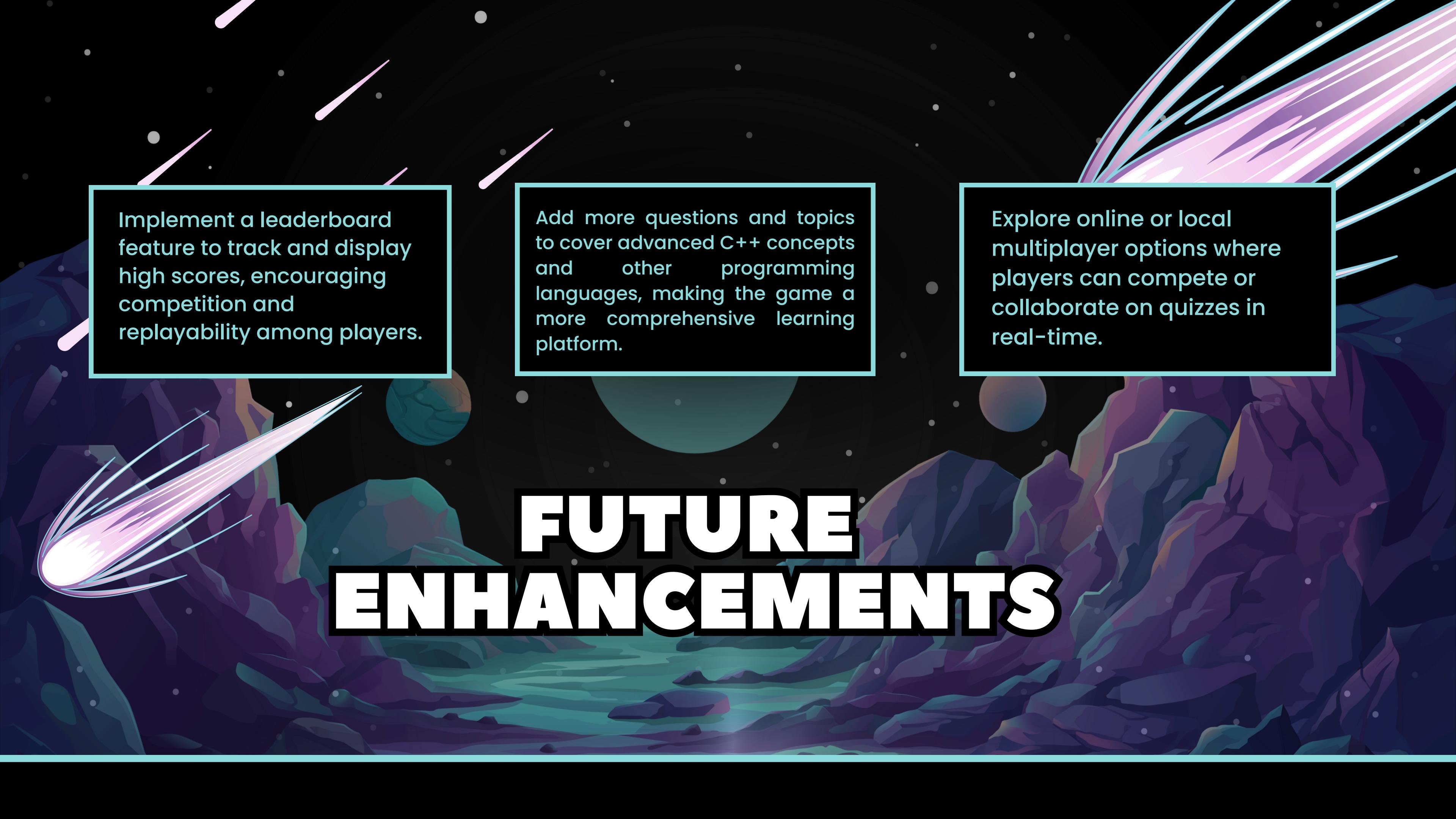
- The game's different components such as Shooter, Bullet, Button, Level, Question, and Explosion are encapsulated as classes to promote modularity and maintainability.
- The main Game class acts as the controller, managing the overall game flow, loading assets, updating states, and rendering.

- An enumeration tracks distinct game states like Menu, Level Selection, In-Game, Rules Screen, and Game Over.
- This allows precise control over event handling and rendering logic, minimizing complexity and bugs.

SFML

- The game is built using C++ and the SFML library, which provides tools for handling graphics, audio, and input events.
- Game assets (fonts, sounds, textures) are loaded from external files, allowing easy future updates or expansions.

- Custom buttons support multiline text and different visual states (hover, disabled), improving accessibility and design flexibility.
- Text elements dynamically update to show score, lives, level progress, and game feedback
- Level unlocking based on player performance encourages continued engagement.



Implement a leaderboard feature to track and display high scores, encouraging competition and replayability among players.

Add more questions and topics to cover advanced C++ concepts and other programming languages, making the game a more comprehensive learning platform.

Explore online or local multiplayer options where players can compete or collaborate on quizzes in real-time.

FUTURE ENHANCEMENTS



LETS
PLAY