The Game

Overview

Cosmic Escape is a 2D top-down arcade survival game set in a science-fiction universe. The player is an astronaut who has been abducted and imprisoned by aliens. After managing to break out of a holding cell, the player must navigate through a series of increasingly hostile environments in order to escape. The game combines fast-paced movement with strategic pathfinding, requiring players to utilize reflexes and environmental tools to survive.

Core Gameplay & Objectives

The primary objective in each level is to locate and collect Energy Crystals. Collecting these crystals powers the Teleporter, which serves as the level's exit. Once the teleporter is activated, the player must reach it to progress to the next level.

The game operates under high-pressure constraints:

- Time Limit: Each level has a strict 60-second timer. If the timer reaches zero before the player escapes, the level restarts.
- Controls: The player controls the astronaut using standard WASD controls for multidirectional movement.
- System: The game features a pause menu, accessible via the ESC key, allowing players to halt gameplay tactically.

Enemies & Vision Mechanics

The alien ship is patrolled by enemies that utilize a "Vision Cone" mechanic. If the player enters an enemy's vision range or collides with them, the player is captured, and the level restarts immediately.

- Aliens: Mobile units that patrol specific routes. They require the player to time their movements carefully to slip past their vision.
- Turrets: Stationary enemies that rotate their field of view. Players must wait for the turret to look away before crossing its vision range.

Environmental Hazards & Puzzle Elements

The map is filled with environmental dangers and obstacles that the player must overcome:

- **Alien Slime:** Sticky patches on the floor that significantly reduce the player's movement speed, making them vulnerable to chasing enemies or running out of time.
- **Radiation Zones:** Hazardous areas that fill a hidden toxicity meter. If the player stays in a radiation zone for too long, they die due to radiation poisoning, causing a level restart.
- **Laser Gates:** Lethal energy barriers that block paths. Touching a laser results in an instant loss.
- **Interactive Buttons:** Players must locate buttons scattered across the map to open locked mechanical doors or deactivate laser gates, often requiring backtracking or risky detours.

Power-Ups

To aid in the escape, four distinct types of power-ups can be found within the levels. Each offers a temporary advantage:

1. Alien Charm (Invulnerability): Acts as a "second life." If the player is caught by an enemy while holding the charm, the charm is consumed, and the player is granted a brief period of invulnerability instead of restarting the level. This allows the player to retreat to a safe position.
2. Time Stop: Temporarily freezes all enemy movement and halts turret rotation. However, enemies remain lethal; the player will still be caught if they collide with a frozen enemy or walk into their static vision cone.
3. Invisibility: Renders the player completely undetectable by enemies for a short duration, allowing them to sprint through vision cones safely.
4. Jetpack: Provides a 50% increase to movement speed, essential for outrunning patrols or traversing large rooms quickly to beat the 60-second timer.

Changes From Our Initial Plan

Our initial plan included adding more levels and increasing the difficulty. We originally intended to introduce more features for the higher levels, such as flying enemies and a special key mechanic that players would need to collect in order to activate the teleporter. These features were meant to enhance difficulty and create a more

complete, “game-like” experience. We also hoped to include an intro sequence before gameplay, such as a short story or video, along with more varied map designs.

In terms of class design, we stayed fairly consistent with our original UML diagram. While we made some adjustments along the way, such as initially removing the parent enemy class and keeping enemies as separate classes, we later reversed that decision in Phase 3. We introduced additional classes specifically to support vision features for enemies and animation functionality, like DoorAnimationController and ButtonAnimationController. These helped us implement animations in a cleaner and more modular way. Ultimately, our final structure aligns closely with our initial UML.

Regarding the maps and visuals, we followed our initial ideas and layout plans, making them more visually appealing while removing some features, like flying enemies, due to time constraints.

Important Lessons Learned

Developing this game taught us several important lessons about game development, time management, and team coordination. One of the biggest takeaways was the importance of separating game logic from the game engine and maintaining clean, high-quality code. Because we weren’t fully consistent with this early on, it made testing significantly more challenging. This experience showed us why good design practices and clear structure are essential.

We also learned the value of starting with a strong, detailed plan. Our Phase 1 deliverables were well discussed and clearly defined, which helped us divide responsibilities and communicate effectively as a team. Having a shared understanding of the project from the start made later development smoother.

Another major lesson was using version control effectively. Throughout the project, we all improved at using GitHub, writing clear commit messages, and organizing our work so team members could easily understand changes. This strengthened our collaboration and prevented confusion when integrating different parts of the project.

Lastly, we understood the importance of going through documentation and also gained valuable experience in reading and understanding documentation. This skill became quite essential as we worked with Litiengine, which had outdated information and very few tutorials or samples to refer to.

Tutorial

We chose to make a video, showcasing our main features, scenarios and gameplay.

Link to the video
Link to jar