**# Game Design Document: Space Explorer**

# ## Technical Requirements

- Unity Version: Unity 6000.0.37f1

- Target Platform: Windows

# # 1. Overview

Space Explorer is a 2D arcade-style space shooter where players pilot a spaceship through dangerous space environments, battling enemies, avoiding asteroids, and collecting stars. The game features multiple types of enemies, boss battles, and a scoring system to encourage replayability.



# # 2. Game Elements

## ## 2.1. Spaceship (PlayerController)

## 

- Description: A 2D spaceship that the player controls with smooth movement and tilt animations.

- Functionality:

- Movement:

- Full directional control using keyboard inputs

- Arrow keys or WASD for movement

- Smooth tilt animation when moving horizontally (20-degree maximum tilt)

- Combat:

- Shoots bullets using spacebar

- 0.2-second shooting cooldown

- Health System:

- 3 hearts maximum health

- Visual heart display in UI

- Damage from various enemy types

- Visual Effects:

- Entrance animation from bottom of screen

- Explosion effect on taking damage

- Bullet firing effects

- Scoring:

- Score display in UI

- Score changes visible to player

## ## 2.2. Asteroids (AsteroidController)

- Description: Hazardous space rocks that fall from the top of the screen.



- Functionality:

- Movement:

- Constant downward speed of 2 units per second

- Automatic destruction when leaving screen bounds

- Combat:

- 3 health points

- Takes damage from player bullets

- Damages player on collision

- Effects:

- Explosion animation on destruction

- Audio feedback on destruction

- Spawn System (AsteroidSpawner):

- Initial maximum spawn rate of 5 seconds

- Spawn rate increases over time

- Minimum spawn interval of 1 second

- Random horizontal position spawning

## ## 2.3. Enemy Ships (EnemyController)

- Description: Hostile spaceships with offensive capabilities.



- Functionality:

- Movement:

- Downward movement at 2 units per second

- Screen boundary checking

- Combat:

- Customizable health system

- Player-tracking bullet system

- Collision damage to player

- Effects:

- Engine visual effects

- Destruction animation

- Audio feedback

- Scoring:

- Point rewards on destruction

## ## 2.4. Boss Ships (BossController)

- Description: Powerful enemy ships with complex attack patterns.



- Functionality:

- Movement:

- Complex movement patterns

- Screen position constraints

- Combat Phases:

- Initial attack phase (3 separate attacks)

- Movement phase with rapid-fire attacks

- Custom bullet patterns

- Visual Features:

- Entrance and exit animations

- Shooting animations

- Health system

- Sound Effects:

- Attack sound effects

- Movement sound effects

## ## 2.5. Stars (StarController)

- Description: Collectible items that provide score bonuses.



- Functionality:

- Movement:

- 2 units per second downward speed

- Automatic cleanup when off-screen

- Rewards:

- 10 points per collection

- Audio feedback on collection

- Spawn System (StarSpawner):

- Initial 5-second maximum spawn rate

- Progressive spawn rate increase

- Minimum 1-second spawn interval

## ## 2.6. Background Elements

- Scrolling Background (Scroller):

- Continuous vertical scrolling effect

- Material-based UV offset

- Planet System (PlanetController):

- 5 unique planet types

- Sequential appearance every 20 seconds

- Automatic position reset

- Queue-based management system



# # 3. Game Systems

## ## 3.1. Score System (ScoreManager)

- Features:

- Real-time score tracking

- Sprite-based number display

- High score persistence using PlayerPrefs

- Score change feedback

- Scoring Rules:

- Star Collection: +10 points

- Collision Damage: -10 points

- Enemy Destruction: Variable points



## ## 3.2. Game State Management (GameManager)

- Game States:

- Gameplay

- Game Over

- Pause

- State Features:

- Gameplay:

- Normal time scale

- Active spawners

- Player control enabled

- Game Over:

- Spawners disabled

- High score check

- UI updates

- Pause:

- Zero time scale

- Pause menu active

- Input handling



# # 4. Technical Implementation

## ## 4.1. Scene Management

- Managed Scenes:

- Main Menu

- Game Levels

- Features:

- Asynchronous scene loading

- State persistence between scenes

- Memory management

## ## 4.2. Audio System

- Sound Effects:

- Weapon firing

- Explosions

- Item collection

- Enemy attacks

## ## 4.3. Save System

- Persistent Data:

- High scores

- PlayerPrefs utilization

- Data management functions

## ## 4.4. Input System

- Keyboard Controls:

- Movement: Arrow keys/WASD

- Shooting: Spacebar

- Pause: ESC key

## ## 4.5. Visual Effects

- Particle Systems:

- Explosions

- Engine effects

- Animation Systems:

- Character animations

- UI transitions

- Environment effects