**Systems Design Document for Flappy Shark**

This SDD serves as a blueprint for the game's systems. It’s vital to ensure each system is modular, scalable, and maintainable for smooth development and potential future updates. If using Unity, many of these systems can be implemented using its built-in features in conjunction with C# scripting.

**1. Core Gameplay Mechanics**

* **One-Tap Control System**
  + *Description*: Players tap the screen to make the shark swim upwards. When released, the shark starts to sink.
  + *Interactions*: With obstacles and terrain.
  + *Variables*: Swimming speed, acceleration.

**2. Obstacle System**

* **Adaptive Spawner System**
  + *Description*: Dynamically spawns mines, missiles, and other potential obstacles.
  + *Interactions*: With player character (shark).
  + *Variables*: Spawn rate, obstacle speed, obstacle types.
* **Obstacle Behavior**
  + *Description*: Defines how each obstacle behaves.
  + *Interactions*: With the player character, terrain, and other obstacles.
  + *Variables*: Movement patterns, activation triggers.

**3. Scoring System**

* **Point Accumulation**
  + *Description*: Players accumulate points based on distance traveled and treasures collected.
  + *Interactions*: With treasures and the distance the player travels without hitting an obstacle.
  + *Variables*: Point values for distance, treasures.

**4. Music Playlist System**

* **Dynamic Track Controller**
  + *Description*: Shuffles and plays tracks from a specified playlist.
  + *Interactions*: With game events (like game start, game over).
  + *Variables*: Playlist array, current track, shuffle logic.

**5. Background Parallax System**

* **Layered Motion Controller**
  + *Description*: Multiple background layers move at different speeds to create a parallax effect.
  + *Interactions*: With player movement and game speed.
  + *Variables*: Speed for each layer, layer ordering.

**6. User Interface System**

* **Menu Controller**
  + *Description*: Manages the main menu, settings, and other UI elements.
  + *Interactions*: With player inputs (tap/click).
  + *Variables*: Current screen, button states.
* **Game Over Logic**
  + *Description*: Triggers when the player hits an obstacle and manages related UI.
  + *Interactions*: With obstacle system, scoring system.
  + *Variables*: Final score, high score.

**7. Game Progression System**

* **Level Difficulty Controller**
  + *Description*: Adjusts the difficulty of the game as the player progresses.
  + *Interactions*: With obstacle system, scoring system.
  + *Variables*: Increase in obstacle spawn rate, change in obstacle types.

**8. Player Feedback System**

* **Sound and Visual Feedback Controller**
  + *Description*: Provides immediate feedback to player actions, such as successful treasure collection or hitting an obstacle.
  + *Interactions*: With game events like collecting treasures, hitting obstacles, etc.
  + *Variables*: Sound effects, particle effects, screen shake.

**9. Save and Load System**

* **Game State Controller**
  + *Description*: Manages the saving and loading of player progress and high scores.
  + *Interactions*: With the scoring system and game over logic.
  + *Variables*: Current score, high score, game settings.