Part 1 - Noun Analysis

(Worked in tandem with Claude LLM)

Game Elements and Objects:

- Ship (the central player-controlled object)
- Asteroids (primary obstacles)
- Saucers (enemy objects)
- Bullets (projectiles)
- Engines (propulsion system)
- Particles (for visual effects)
- Screen/Canvas (game display area)

Gameplay Mechanics:

- Lives (player resource)
- Score (performance metric)
- Points (scoring system)
- Collisions (interaction between objects)
- Force (physics element)
- Velocity (movement characteristic)
- Momentum (physics property)
- Rotation (movement type)
- Teleportation/Hyperspace (movement mechanic)
- Levels (game progression)
- Knockback (physics effect)
- Aim (targeting system)

Technical Components:

- Classes (code organization)
- Properties (object attributes)
- Variables (data storage)
- Functions (code procedures)
- Hierarchy (class organization)
- State (game condition tracking)
- IDE (development environment)
- Commits (version control)
- Repository (code storage)

Project Structure:

- Features (implementation requirements)
- Tasks (additional requirements)

- Marks (grading units)
- Grade (final evaluation)
- Opportunities (marking sessions)
- Reflection (documentation requirement)

Pruned list:

Core Game Objects:

- Ship (player's triangle-shaped vessel)
- Asteroids (large, medium, small sizes)
- Saucers (large and small enemy ships)
- Bullets (from both ship and saucers)
- Particles (for explosions)

Essential Properties:

- Position (P5.Vector for all game objects)
- Velocity (P5.Vector for movement)
- Direction (for ship rotation and bullet firing)
- Force (for engine thrust)

Game State Tracking:

- Score (running total)
- Lives (player's remaining attempts)
- Level (current game stage)
- Invincibility (temporary player state)
- Game state (global state)

Mermaid Diagram

```
classDiagram
    GameObject <|-- Ship
    GameObject <|-- Asteroid
    GameObject <|-- Bullet
    GameObject <|-- Saucer
    GameObject <|-- Particle
    GameManager "1" --> "*" GameObject
    GameManager "1" --> "1" SoundManager
    GameManager "1" --> "1" UIManager
    GameManager "1" --> "1" LeaderboardManager
    ParticleSystem "1" --> "*" Particle

    class GameObject {
        +Vector position
        +Vector velocity
        +float rotation
        +boolean isActive
        +float radius
        +update()
        +draw()
        +checkEdges()
        +checkCollision(GameObject other)
        +destroy()
    }

    class Ship {
        -int lives
        -boolean isInvincible
        -boolean engineOn
        +rotate(float angle)
        +thrust()
        +shoot()
        +teleport()
        +takeDamage()
        +resetPosition()
    }

    class Asteroid {
        -string size
        -float rotationSpeed
```

```
    +split()
    +getPoints()
}

class Bullet {
    -float lifespan
    -GameObject source
    +checkLifespan()
}

class Saucer {
    -string size
    -float accuracy
    -float shootingInterval
    +updateAI()
    +shoot()
    +calculateAim()
}

class GameManager {
    -int score
    -int level
    -boolean isGameOver
    +startGame()
    +updateGame()
    +spawnAsteroids()
    +spawnSaucer()
    +checkLevelComplete()
    +updateScore()
}

class SoundManager {
    -Map sounds
    +playSound(string name)
    +playMusic()
    +stopMusic()
}

class UIManager {
    +drawScore()
    +drawLives()
    +showTitleScreen()
    +showGameOver()
}
```

```
class LeaderboardManager {
    -Array scores
    +addScore(string name, int score)
    +getTopScores()
    +saveScores()
    +loadScores()
}

class ParticleSystem {
    -Array particles
    +emit(Vector position, int count)
    +update()
    +draw()
}

class Particle {
    -float lifespan
    -color color
    +update()
    +isDead()
}
```