**User Story 1: Player Movement**

As a player, I want to control my character's movement.

**INVEST:**

Independent: This story can be developed and tested without reliance on other major features.

Negotiable: Specifics of movement controls can be adjusted based on player feedback.

Valuable: Movement is fundamental for gameplay and user interaction.

Estimable: Implementing basic movement controls is straightforward.

Small: Focuses on movement without additional complexities.

Testable: Movement controls are observable and measurable.

**3Ws:**

Who: Player

What: Control character movement

Why: To navigate the game world

**acceptance criteria**

-player sprite appears on screen

-moves left/right/up/down while A/D/S/W keys are down

- not moving if ADSW keys not down

- stops moving at screen boundary

- sprite visually corresponds to direction of movement

**User Story 2: Enemy Encounter**

As a player, I want to encounter enemies that pose a challenge and must be avoided or defeated.

**INVEST:**

Independent: Enemy encounters can be implemented without other complex systems.

Negotiable: Enemy behaviour and difficulty can be adjusted based on playtesting.

Valuable: Adds excitement and challenge to the gameplay experience.

Estimable: Implementing basic enemy encounters is achievable within the MVP scope.

Small: Focuses on introducing basic enemies

Testable: Enemy encounters can be verified for correct behaviour and how challenging the level is.

**3Ws:**

Who: Player

What: Encounter enemies

Why: To introduce challenge and excitement

**Acceptance Criteria:**

-Enemies spawn at predefined locations in the game world.

-Enemies move towards the player.

-Player can defeat enemies by shooting at them.

**User Story 3: Scoring and Game Over**

As a player, I want to earn points and eventually reach a game-over state to track my progress.

**INVEST:**

Independent: Scoring and game-over conditions can be implemented regardless of other features.

Negotiable: Scoring rules and game-over conditions can be adjusted based on gameplay balance.

Valuable: Provides a clear objective on what the player has to do.

Estimable: Implementing scoring and game-over logic is straightforward.

Small: Focuses on basic scoring and game-over conditions without advanced features.

Testable: Scoring and game-over conditions can be verified through gameplay testing.

**3Ws:**

Who: Player

What: kill all enemies or get killed by enemies

Why: To track progress and provide challenge

**Acceptance Criteria:**

-Player score increases the longer they stay alive or defeating enemies.

-game ends when player kills all the enemies or when player dies.

-Game-over screen displays with final score and option to restart.

**User Story 4: Player Shooting**

As a player, I want to shoot projectiles at enemies to defeat.

**INVEST:**

Independent: Shooting mechanics can be implemented without major dependencies.

Negotiable: Specifics of shooting behaviour such as projectile type and firing rate can be adjusted based on gameplay needs.

Valuable: Adds strategic depth and combat options for engaging with enemies.

Estimable: Implementing basic shooting mechanics should be simple.

Small: Initial implementation of shooting without complex features like weapon upgrades or choosing weapons.

Testable: Shooting mechanics can be verified through interaction with enemies.

**3Ws:**

Who: Player

What: Shoot projectiles at enemies

Why: To defeat enemies and win the game.

**Acceptance Criteria:**

-Player can press space to shoot projectiles.

-Projectiles are instantiated from the player's position and move towards the direction they are fired.

-Projectiles collide with enemies, reducing enemy health or defeating them.

-Shooting has a cooldown period to prevent rapid firing.