Seekers Test Plan for Milestone 3

CPSC 427 – Video Game Programming

# Team: Los Pollos Hermanos Gaming (LPHG)

Samuel Yiu (32494940)

Ahmed Al Busaidy (29200722)

Jakob Khalil (86176674)

Amir Goodarzvard Chegini (56851199)

## 1. Save/Load System Testing

### 1.1. Save Game Functionality

* Press F5 to save the game state
* Verify save file creation in saves directory
* Test multiple save states
* Test path: src/systems/SaveLoadSystem.hpp

### 1.2. Load Game Functionality

* Press F6 to load the most recent save
* Verify correct restoration of:
  + Player position
  + Health/Energy status
  + Inventory items
  + World state
* Test loading from different save points

## 2. Open World & Dungeon System

### 2.1. Open World Environment

* Verify proper player spawn location
* Test environment object placement:
  + Trees
  + Rocks
  + Bonfire
  + Portal
* Test path: src/systems/OpenWorldMapCreatorSystem.hpp

### 2.2. Dungeon Generation

* Test dungeon entrance portal interaction
* Verify procedural generation of:
  + Rooms
  + Corridors
  + Enemy placement
  + Exit placement
* Test path: src/systems/ProceduralGenerationSystem.hpp

### 2.3. World Transition

* Test seamless transition between open world and dungeon
* Verify proper state preservation during transitions
* Check lighting system changes between environments
* Test path: src/app/MapManager.hpp

### 2.4. Visual Environment

* Test different skybox themes:
  + Open world skybox
  + Dungeon skybox
* Verify light orb placement and illumination in dungeons
* Check ambient lighting transitions
* Test path: src/renderer/SkyboxRenderer.hpp

## 3. Enhanced Combat System

### 3.1. Lock-On System

* Right-click to toggle enemy lock-on
* Verify camera behavior during lock-on
* Test lock-on range limitations
* Check auto-disable on enemy death
* Test path: src/systems/GameplaySystem.hpp

### 3.2. Health System

* Test health potion consumption (Key: 1)
* Verify maximum potion limit (3)
* Test health restoration amount
* Check potion UI display
* Test path: src/components/GameplayComponents.hpp

## 4. Interaction System

### 4.1. Bonfire Interaction

* Press F near bonfire to interact
* Verify rest animation plays
* Test health/energy restoration
* Check potion refill
* Test path: src/systems/InteractionSystem.hpp

### 4.2. Portal Interaction

* Press F near portal to interact
* Verify transition animation
* Test loading screen
* Check world state preservation

## 5. Memory Management

### 5.1. Resource Management

* Test extended gameplay (>5 minutes)
* Monitor memory usage during:
  + World transitions
  + Combat sequences
  + Asset loading/unloading
* Verify proper cleanup of:
  + Destroyed entities
  + Unused assets
  + Audio resources

### 5.2. Performance Optimization

* Monitor FPS counter during:
  + Heavy combat
  + World transitions
  + Multiple enemies
  + Particle effects
* Test path: src/app/Application.hpp

## 6. Input Handling

### 6.1. Input Robustness

* Test rapid input combinations
* Verify proper handling of:
  + Invalid key combinations
  + Multiple key presses
  + Mouse movement during transitions
* Test path: src/app/InputManager.hpp

## 7. Audio System

### 7.1. Sound Effects

* Verify proper audio scaling with distance
* Test combat sound effects
* Check environmental audio
* Verify proper audio cleanup
* Test path: src/systems/AudioSystem.hpp

## 8. Stability Testing

### 8.1. Extended Gameplay

* Test continuous gameplay for 5+ minutes
* Verify consistent performance
* Monitor resource usage
* Check for memory leaks

### 8.2. Error Handling

* Test recovery from:
  + Invalid save files
  + Corrupted assets
  + Network disconnection
  + Low memory conditions

## 9. Comprehensive Gameplay Testing

### 9.1. Five-Minute Feature Demo

* Verify all features can be demonstrated within 5 minutes:
  1. Open world exploration
  2. Bonfire interaction with sitting animation
  3. Health potion system
  4. Enemy lock-on system
  5. Portal transition to dungeon
  6. Save/Load functionality
  7. Different lighting environments

### 9.2. Progressive Gameplay

* Test non-repetitive gameplay elements:
  + Enemy variety and placement
  + Dungeon layout variation
  + Combat scenarios
  + Environmental interaction opportunities

## Test Execution Notes:

### 1. Environment Setup

* Clean build directory
* Fresh game instance
* Various hardware configurations
* Different display resolutions

### 2. Test Documentation

* Record all test results
* Document any bugs found
* Track performance metrics
* Note system resource usage

### 3. Success Criteria

* Stable 60+ FPS
* No memory leaks
* Smooth world transitions
* Consistent save/load functionality
* Proper error handling
* Clean audio management

### 4. Bug Priority Levels

* P0: Game-breaking bugs
* P1: Severe gameplay impact
* P2: Minor gameplay issues
* P3: Visual/Audio glitches
* P4: Quality-of-life improvements
* P5: Future enhancements