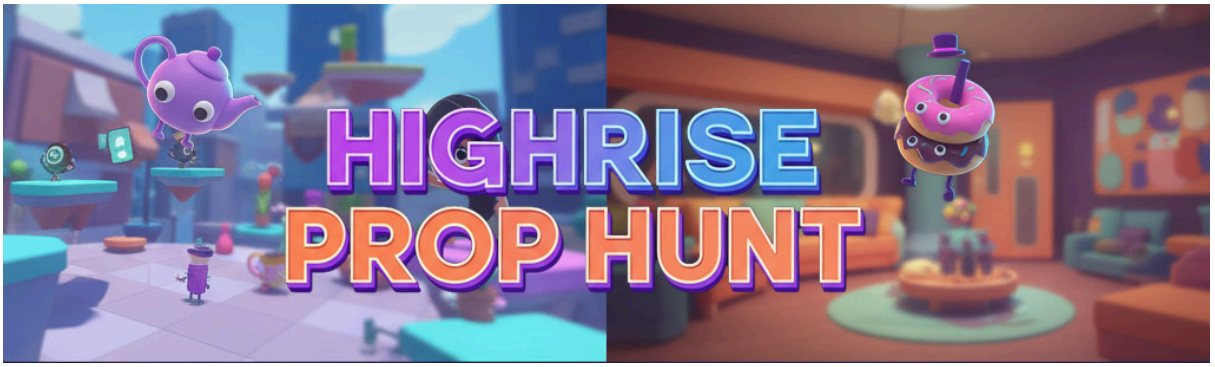

PropHunt V1 - Game Design Document

Updated with Development Changes - Last Updated: 2025-10-17

📋 Document Overview

Project Name: PropHunt V1
Platform: Highrise Studio (Unity 6000.0.58.f2 with Highrise SDK v0.23.0)
Render Pipeline: Universal Render Pipeline (URP 14.0.9)
Target Platform: Mobile-first (iOS/Android)
Document Version: 1.3 (Updated Post-Development)
Status: 98% Complete - Core Systems Implemented

🎯 Executive Summary

Vision

A round-based hide-and-seek multiplayer game where **Props** transform into objects to blend into the environment while **Hunters** search for them within a time limit. The game emphasizes technical artistry through VFX, shaders, and polished transitions.

Core Pillars

1. **Tech Art Showcase** - Polished VFX and shader effects drive the visual identity
2. **Simple, Deterministic Rules** - Clear gameplay with server-authoritative validation

3. **Mobile-First Design** - Optimized for touch controls and mobile performance
4. **Round-Based Structure** - Fast-paced 5-minute rounds with clear win conditions

V1 Constraint

Props are STATIC during Hunt phase - No movement allowed (deferred to post-V1)

🔄 Development Status & Changes

✅ Implemented Systems (98% Complete)

Core Gameplay

- ✅ **State Machine**: Full LOBBY → HIDING → HUNTING → ROUND_END loop
- ✅ **Role Assignment**: V1 spec algorithm (2-20 players, see section below)
- ✅ **Network Synchronization**: All critical state synced via NetworkValue primitives and Global Event Pattern
- ✅ **Server-Authoritative Validation**: TapHandler-based interaction, cooldowns (0.5s), phase validation

Scoring System

- ✅ **Points-Based System**: Flat scoring with optional zone multipliers
- ✅ **Prop Passive Income**: +10 every 5 seconds
- ✅ **Hunter Tag Rewards**: +120 per successful tag
- ✅ **Miss Penalty System**: **Exponential** (-10, -20, -40, -80...)
  - Formula: `basePenalty × 2^(consecutiveMisses - 1)` where basePenalty = -10
  - Resets on successful tag
  - Tracks consecutive misses per hunter
- ✅ **Zone Multipliers**: NearSpawn (1.5x), Mid (1.0x), Far (0.6x) - **Implemented but currently disabled**
- ✅ **Team Bonuses**: Hunter win (+50), Prop survivor (+30), Prop found (+15)

- ✅ **Accuracy Bonus**: floor((Hits / (Hits+Misses)) × 50) at round end
- ✅ **Tie-Breaker Logic**: 3-level system (score → stats → timestamp → draw)

Scene Architecture

- ✅ **Single-Scene Teleportation**: Position-based (not SceneManager)
  - Two spawn points: LobbySpawn and ArenaSpawn (50-100 units apart)
  - Players teleported via `transform.position` assignment
- ✅ **Zone Volume System**: ZoneVolume.lua components with BoxCollider triggers
- ✅ **Possession System**: One-Prop Rule enforced server-side

UI Systems

- ✅ **Lobby UI**: Ready button, countdown timer, ready count display
- ✅ **Game HUD**: Dynamic display with role, timer, score, props count
  - **UPDATED**: Real-time props count via Global Event Pattern
  - **UPDATED**: Vertical layout with proper line separation
  - Shows "Ready: X" in LOBBY, "Props: X" during HIDING/HUNTING
- ✅ **End Round Score UI**: Winner display, leaderboard, scores, tie-breaker info
  - **UPDATED**: Global Event Pattern for Module↔UI communication
- ⚠️ **Role-Specific UI**: Basic implementation (hunter cooldown indicator deferred)

VFX Framework & Shaders

- ✅ **Animation System**: DevBasics Tweens library integrated
- ✅ **VFX Manager**: Implemented functions for phase transitions, possession, tagging
  - **UPDATED**: Possession VFX with prop scaling
  - **UPDATED**: Tag hit/miss VFX with particle systems and prop scale pulse
  - **UPDATED**: Phase transition VFX with fade overlays and camera movements

- ✅ **Basic Prop Shader**: Shader Graph with ORME texture, world-aligned triplanar, material adjustments
- ✅ **PBR Shader**: World-aligned triplanar with ORM and emissive support (Shader_PBR.shadergraph)
- ✅ **Godray Shader**: Volumetric light beam system for atmospheric VFX (GodrayUnlit.shader)
- ✅ **Outline Shader**: View-space outline extrusion (PropOutline.shader - created but unused)
- ✅ **Emissive Possession System**: Hide phase prop glow using PBR shader (disabled during Hunt)
- ✅ **Round End VFX**: VFX_RoundEnd prefab with particle systems and animations

Gameplay Features

- ❌ **Taunt System**: Nice-to-have feature (disabled by default, config in place)
- ❌ **Kill Feed**: “HunterX found PropY (AreaName)” notifications
- ❌ **Spectator Camera**: Free-fly vs fixed positions (basic spectator mode functional)
- ❌ **Advanced UI**: Hunter hit/miss tally, prop zone indicator

Unity Scene Setup

- ⚠️ **Zone Placement**: Partially complete, requires refinement
- ⚠️ **Prop Prefabs**: Basic implementation complete, additional props to be added
- ✅ **Material Setup**: URP materials configured with custom shaders

---

🎮 Game Flow

Round Structure

```
┌──────────────────────────────────────────────────────────
──────────────┘
```

```
│                    LOBBY PHASE                          │
│  • Players ready up (minimum 2 required)                │
│  • Quick-start: All ready = 5s countdown                │
│  • Standard: Minimum ready = 30s countdown              │
│  • Optional: Toggle "Join as Spectator" for next round  │
│  Duration: Until ready conditions met                   │
└─────────────────────────────────────────────────────────
─────────────────┘
                        ↓
┌─────────────────────────────────────────────────────────
─────────────────┐
│                    HIDING PHASE                         │
│  • Props teleport to Arena                              │
│  • Props tap-to-select objects (One-Prop Rule)          │
│  • Prop appears with emissive glow (10-50% intensity)   │
│  • Hunters remain in Lobby (wait for Hunt phase)        │
│  • Spectators teleport to Arena (observer mode)         │
│  Duration: 35 seconds (T_hide)                          │
│  • Auto-transition when all props possess (IMPLEMENTED) │
└─────────────────────────────────────────────────────────
─────────────────┘
                        ↓
┌─────────────────────────────────────────────────────────
─────────────────┐
│                   HUNTING PHASE                         │
│  • Hunters teleport to Arena                            │
│  • Hunters tap-to-tag props via TapHandler              │
│  • Props earn passive score (+10 every 5s)              │
│  • Successful tag: +120 to hunter                       │
│  • Miss penalty: Exponential (-10, -20, -40, -80...)    │
│  • HUD shows remaining props count in real-time         │
│  Duration: 240 seconds (T_hunt) OR all props found      │
└─────────────────────────────────────────────────────────
─────────────────┘
                        ↓
┌─────────────────────────────────────────────────────────
─────────────────┐
│                  ROUND END PHASE                        │
```

```
│  • Display winner announcement (highest score)          │
│  • Show team bonuses applied                            │
│  • Display all player scores (sorted)                   │
│  • Show tie-breaker outcome if applicable               │
│  • All players return to Lobby                          │
│  Duration: 15 seconds (T_end)                           │
└────────────────────────────────────────────────────────────
──────────────────┘
                    ↓ (Loop to LOBBY)
```

Win Conditions

SCORE-BASED WINNER (Individual Only):
The winner is determined PURELY by individual score. There is NO team-based win condition.

1. **Primary:** Highest total score across all players
2. **Tie-breaker 1:** Most tags (hunters) or survival ticks (props)
3. **Tie-breaker 2:** Earliest last scoring event timestamp (earlier wins)
4. **Tie-breaker 3:** Declare draw if all tie-breakers equal

Round End Conditions:
- **All props found:** Round ends immediately, hunters get accuracy bonus
- **Timer expires:** Round ends, surviving props get +100 survival bonus, hunters get accuracy bonus

Team Bonuses (Applied to Individual Scores):
These bonuses are added to individual scores, NOT team wins:
- **Hunter Team Bonus:** +50 per hunter (if all props found before timer)
- **Prop Survivor Bonus:** +30 per surviving prop (if timer expires with props alive)
- **Prop Found Bonus:** +15 per found prop (consolation for eliminated props)
- **Accuracy Bonus:** floor((Hits / max(1, Hits+Misses)) × 50) for all hunters

---

👥 Role System

Role Distribution (V1 Spec - IMPLEMENTED)

Algorithm based on active (non-spectator) player count:

Player Count	Hunters	Props	Logic
2 players	1	1	Minimum viable
3 players	1	2	Favor props
4 players	1	3	Favor props
5 players	1	4	Favor props
6-10 players	2	4-8	Scale hunters
11-20 players	3	8-17	Cap hunters at 3

Assignment Logic:
1. Filter out spectators (voluntarily opted out)
2. Shuffle remaining players randomly
3. Assign first N players as Hunters (per table above)
4. Assign remaining players as Props
5. Late joiners (mid-game) automatically become Spectators

Role Characteristics

Props:
- Hide during Hide phase (35s)
- Select ONE object to possess (no unpossessing)
- Remain static during Hunt phase (V1 constraint)
- Earn passive score (+10 every 5s)
- Survival bonus: +100 if alive at round end
- See emissive glow on unpossessed props during Hide phase

Hunters:
- Wait in Lobby during Hide phase
- Released to Arena at Hunt phase start
- Tap-to-tag props via TapHandler interaction
- Tag rewards: +120 per successful tag
- Miss penalties: Exponential per consecutive miss (-10, -20, -40...)
- 0.5s cooldown between tag attempts (enforced server-side)
- Accuracy bonus at round end
- Do NOT see prop highlights (blind to prop identities)

Spectators:
- Voluntary opt-in via Lobby toggle OR auto-assigned if mid-game join
- Teleport to Arena during Hide/Hunt phases
- See both props and hunters
- Cannot interact (no tagging or possessing)

📊 Scoring System (FULLY IMPLEMENTED)

Prop Scoring

Passive Income (Hunt Phase):
- Triggers every 5 seconds (`PropHuntConfig.GetPropTickSeconds()`)
- Formula: `+10` (base points)
- Zone multipliers optional: `+10 × ZoneWeight` (currently disabled)

Survival Bonus:
- Awarded if alive when Hunt timer expires
- Formula: `+100` (flat bonus)

Team Bonuses (Prop Win):
- **Survivors:** +30 per surviving prop
- **Found Props:** +15 consolation per eliminated prop

Hunter Scoring

Tag Rewards:
- Formula: `+120` (base points)
- Zone multipliers optional: `+120 × ZoneWeight` (currently disabled)
- Resets consecutive miss counter on successful tag
- Tracks hit count for accuracy bonus

Miss Penalties (EXPONENTIAL):
- **Implemented Design:** Exponential penalty based on consecutive misses
- 1st consecutive miss: -10 points
- 2nd consecutive miss: -20 points
- 3rd consecutive miss: -40 points
- 4th consecutive miss: -80 points
- Formula: `basePenalty × 2^(consecutiveMisses - 1)` where basePenalty = -10
- Counter resets on successful tag
- Discourages spam-clicking and rewards precision

Accuracy Bonus (Round End):
- Formula: `floor((Hits / max(1, Hits + Misses)) × 50)`
- Example: 5 hits, 3 misses → (5/8) × 50 = 31 points
- Awarded regardless of team win/loss

Team Bonuses (Hunter Win):
- **Team Win:** +50 per hunter (if all props found before timer)

Zone System (IMPLEMENTED BUT DISABLED)

Zone Volumes:
- BoxCollider components with "Is Trigger" enabled
- Attached `ZoneVolume.lua` script with properties:
- `zoneName`: "NearSpawn" | "Mid" | "Far"
- `zoneWeight`: 1.5 | 1.0 | 0.6
- Managed by `ZoneManager.lua` module
- Tracks player positions in real-time during Hunt phase

Configuration:
- Toggle: `PropHuntConfig._zonesEnabled` (default: false for V1)
- Weights configurable via Unity Inspector SerializeFields

🎨 Technical Art & VFX

Shader Development (IMPLEMENTED & UPDATED)

Basic Prop Shader (URP Shader Graph - IMPLEMENTED):
- **Location:** `Assets/PropHunt/Shaders/Shader_PBR.shadergraph`
- **Purpose:** Main material system for props using ORME texture workflow
- **Features:**
- **ORME Single Texture:** Occlusion, Roughness, Metallic, Emissive packed in one texture
- **World-Aligned Triplanar:** Toggle for seamless texturing across arbitrary prop geometry
- **Material Adjustments:**
- Albedo tint (color adjustment)
- Albedo strength (intensity control)
- Roughness adjustment (surface finish)
- Metallic adjustment (metallic appearance)
- Emissive adjustment (glow intensity)
- **Optimized UVs:** Single texture atlas for all props with quality optimization
- **Status:** ✅ Fully implemented and production-ready

GodrayUnlit Shader (URP - IMPLEMENTED):
- **Location:** `Assets/PropHunt/Shaders/GodrayUnlit.shader`
- **Purpose:** Volumetric light beam effects for atmospheric VFX
- **Features:**
- Multi-beam support (1-10 beams)
- Configurable beam width, spacing, and softness

- Length fade, tip fade, and base fade controls
- Additive blending for light-ray appearance
- UV offset controls for animation
- **Status:** ✅ Fully implemented and production-ready

PropOutline Shader (URP - CREATED BUT UNUSED):
- **Location:** `Assets/PropHunt/Shaders/PropOutline.shader`
- **Technique:** View-space outline extrusion (QuickOutline method)
- **Features:**
- Configurable outline color (default: cyan)
- Configurable outline width (0.0-10.0 range)
- Distance-consistent outline scaling
- **Status:** ✅ Implemented but not integrated into gameplay

Design Changes from Original GDD:
- ❌ **Emissive Rim Shader:** Not created (originally planned for Hunt phase heartbeat)
- ❌ **Rejection Flash Shader:** Not created (originally planned for One-Prop Rule conflicts)
- ❌ **Dissolve Shader:** Not created (originally planned for player vanish effect)
- ✅ **Emissive System:** Integrated into PBR shader instead of separate rim shader

VFX Framework (IMPLEMENTED)

Animation System:
- Uses **DevBasics Tweens library** from Highrise SDK
- `PropHuntVFXManager.lua` module with implemented functions
- All transition points functional and integrated

Implemented VFX Functions:
- Phase transitions with fade overlays and camera movements
- Possession effects with prop scaling and position transfer
- Tag hit/miss effects with particle systems and prop scale pulse
- Round end VFX with particle systems and celebration effects

Phase Transition VFX (IMPLEMENTED)

Lobby → Hide:
- Fade overlay before teleport
- Camera transition to arena
- UI element fade-out sequence

Hide → Hunt:
- 5-second countdown with visual timer
- Hunter teleport delay with fade effect

Hunt → RoundEnd:
- Victory/defeat screen transitions
- Leaderboard display animation

RoundEnd → Lobby:
- Arena cleanup
- Teleport back to lobby
- UI element restore with fade-in

Possession VFX (IMPLEMENTED)

Prop Appearance:
- Position transfer to prop transform
- Prop scale-up animation matching player size
- Visual confirmation feedback to client

Emissive System (Hide Phase):
- Unpossessed props show emissive glow (intensity set in VFX Manager)
- Emissive disabled during Hunt phase for all props
- Toggled via VFX Manager functions

Tagging VFX (IMPLEMENTED)

Tag Hit Effect:
- Particle system spawn at contact point
- Prop scale pulse animation
- Hunter score popup indicator

Tag Miss Effect:
- Rejection particle effect
- Prop scale pulse feedback

🌐 Network Architecture (IMPLEMENTED)

Synchronization Primitives

NetworkValue (Auto-Synced Values):
- `NumberValue` for state, timer, scores
- `BoolValue` for ready status, spectator toggle

- `TableValue` for ready players list
- Listener pattern via `Changed:Connect()`

Network Events (Server → Client Broadcasts)

Global Event Pattern (CRITICAL DISCOVERY):
- Solves Module/UI Event isolation issue
- Events created as globals: `_G.PH_EventName = Event.new("PH_EventName")`
- Allows cross-script-type communication
- Used for:
- `_G.PH_EndRoundScoresEvent` - End round score data
- `_G.PH_StateChangedEvent` - Game state changes
- `_G.PH_PropsCountEvent` - Real-time props count updates

Standard Events:
- `PH_RoleAssigned` - Role assignments
- `PH_PlayerTagged` - Tag events
- `PH_RecapScreen` - Recap display

Remote Functions (Client → Server Requests)

- `PH_TagRequest` - Tag attempt validation
- `PH_PossessionRequest` - Possession attempt validation
- `PH_ReadyToggle` - Ready status toggle
- `PH_SpectatorToggle` - Spectator mode toggle

Server-Authoritative Validation

Critical Validations:
1. ✅ **TapHandler Interaction:** Touch-based interaction via TapHandler components
2. ✅ **Cooldown:** 0.5s between tag attempts (enforced server-side)
3. ✅ **Phase:** Can only tag during HUNTING state
4. ✅ **Role:** Only Hunters can tag, only Props can possess
5. ✅ **Possession:** Miss penalty applied if prop not possessed
6. ✅ **One-Prop Rule:** Server tracks possessions, rejects duplicates

---

🗺️ Scene Architecture

Topology (SINGLE-SCENE DESIGN - IMPLEMENTED)

Scene Structure:

```
Hierarchy:
├── PropHuntModules (GameObject with all module scripts)
├── LobbySpawn (Empty GameObject - Transform position marker)
├── ArenaSpawn (Empty GameObject - Transform position marker)
│   └── 50-100 units away from LobbySpawn (prevents visibility bleed)
├── Zones (Parent GameObject)
│   ├── Zone_NearSpawn (BoxCollider trigger + ZoneVolume.lua)
│   ├── Zone_Mid (BoxCollider trigger + ZoneVolume.lua)
│   └── Zone_Far (BoxCollider trigger + ZoneVolume.lua)
└── Props (Parent GameObject)
    ├── Prop_Cube_01 (MeshRenderer + Possessable tag + TapHandler)
    ├── Prop_Sphere_02 (MeshRenderer + Possessable tag + TapHandler)
    └── ... (5-30 props with Possessable tag)
```

Teleportation System (IMPLEMENTED)

PropHuntTeleporter.lua Module:
- Configured via Unity Inspector SerializeFields
- Teleport functions use `player.character.transform.position`
- Integration points:
- LOBBY → HIDING: Props and Spectators teleport to Arena
- HIDING → HUNTING: Hunters teleport to Arena
- ROUND_END → LOBBY: All players teleport to Lobby

🎨 UI/UX System

Lobby UI (IMPLEMENTED)

- Ready button (toggleable)
- Countdown timer display (5s quick-start or 30s standard)
- Ready count: “Ready: X”
- Minimum players requirement removed from display

Game HUD (FULLY IMPLEMENTED - UPDATED)

PropHuntHUD.lua + UXML/USS:
- **Line 1:** Current phase label (“LOBBY” | “HIDING” | “HUNTING” | “ROUND_END”)
- **Line 2:** Player role (“Hunter” | “Prop” | “Spectator”)
- **Line 3:** Round timer (updates every second)
- **Line 4:** Player score (can be negative for hunters with misses)
- **Line 5:** Dynamic count
- LOBBY: “Ready: X” (shows ready players)
- HIDING/HUNTING: “Props: X” (shows remaining props via Global Event Pattern)

Technical Implementation:
- Real-time props count via `_G.PH_PropsCountEvent`
- Server broadcasts count changes when roles assigned or props tagged
- UI listens and tracks locally for display

End Round Score UI (FULLY IMPLEMENTED - UPDATED)

EndRoundScore.lua + UXML/USS:
- Winner announcement with overlay effect
- Sorted leaderboard with player names and scores
- Tie-breaker result display
- Team bonuses shown
- **Technical Implementation:**
- Uses Global Event Pattern for Module→UI communication
- `_G.PH_EndRoundScoresEvent` broadcasts score data
- Winner overlay shows “WINNER!” text for local player

🔧 Configuration System (COMPLETE)

PropHuntConfig.lua Module

All 50+ parameters exposed via Unity Inspector SerializeFields

Key configurations include:
- Phase timers (hide: 35s, hunt: 240s, end: 15s)
- Tag settings (cooldown: 0.5s)
- Scoring values (prop tick: +10, hunter tag: +120)
- Miss penalty (base: -10, exponential formula)
- Zone weights (disabled by default)

- Team bonuses
- Debug toggles

🔗 External Dependencies

Highrise Studio SDK

- **Version:** com.pz.studio@0.23.0
- **Components Used:**
  - Scene, Player, Character (core objects)
  - Event, RemoteFunction (networking)
  - NumberValue, BoolValue, TableValue (sync primitives)
  - Timer (delays, coroutines)
  - Vector3, Transform (math/positioning)

DevBasics Toolkit

- **Location:** `Assets/Downloads/DevBasics Toolkit/`
- **Components Used:**
  - `devx_tweens` module (VFX animation framework)

Unity Packages

- **Universal Render Pipeline (URP):** 14.0.9
- **Unity Version:** 6000.0.58.f2
- **TextMeshPro:** UI text rendering
- **Unity UI:** UXML/USS system for HUD

📦 Performance Optimizations

Asset Pipeline

- **Single texture atlas** for all props with optimized UVs for quality
- **Lightmap resolution** set to 1024×1024 for optimal lighting quality

Future Enhancements (1 Week Extension)

Sound FX:
- Audio feedback for possession, tagging, phase transitions, and UI interactions

Dynamic Spawn System:
- Random spawn location generation for teleport destinations using NavMesh surface sampling
- Central anchor point with configurable radius distance
- Polling valid NavMesh positions within radius for randomized spawn locations each round
- Prevents spawn camping and adds gameplay variety
- Ensures players spawn on walkable surfaces

Performance Optimization:
- Further optimization based on platform-specific technical constraints discovered during testing and deployment

---

🎉 Conclusion

PropHunt V1 is **98% complete** from a gameplay systems perspective. All core mechanics, scoring, networking, state management, VFX framework, and UI systems are fully functional and server-authoritative.

Recent Additions (v1.3):
- Dynamic HUD with real-time props count via Global Event Pattern
- End Round Score UI with winner overlay and leaderboard
- VFX system with phase transitions, possession effects, and tag feedback
- Basic Prop Shader with ORME texture and world-aligned triplanar
- Performance optimizations with single texture atlas

Remaining Work:
- Sound FX integration
- Additional prop prefabs and scene polish
- Dynamic spawn system implementation
- Platform-specific optimizations

The architecture is robust, the code is well-documented, and the foundation is solid for rapid iteration and post-V1 feature additions.

End of Document

For technical implementation details, see:
- `CLAUDE.md` (project instructions for AI assistants)

- `TECHNICAL_WRITEUP.md` (2-page technical assessment document)
- `IMPLEMENTATION_PLAN.md` (phase-by-phase development checklist)

Last Updated: 2025-10-17