PROJECT FUNCTIONAL SPECIFICATION DOCUMENT

TEAM COMPOSITION

- **SALVAN Achilles** - Technical Director
- **BENNJAKHOUKH Rayan** - Group Leader
- **HADJAB Mehdi** - Game Manager
- **GREGORY-LUCAS Clement** - Team Coordinator
- **PUIIJALON Milan** - Creative Director

PROJECT NAME: Rooftop Rivals
DATE: January 2026
VERSION: 1.0

---

1. CONTEXT AND OBJECTIVES

1.1 Project Genesis

When we initially brainstormed ideas for this video game, we immediately gravitated toward the concept of iconic games, titles that transcend age barriers and skill levels. We envisioned a game that could be enjoyed casually with family and friends, yet simultaneously offer depth for competitive players seeking mastery. The legendary Mario Kart franchise served as our primary inspiration, as it exemplifies this dual-nature design philosophy perfectly. These are the games that create lasting memories through shared laughter, friendly rivalry, and moments of excitement.

1.2 Market Analysis and Gaming Landscape

The contemporary gaming market is saturated with exceptional titles. We have adventure games with breathtaking visuals, competitive multiplayer experiences with addictive gameplay loops, and technically sophisticated products that push hardware limitations. Many of these games are enjoyed with friends through voice chat and online connectivity. However, we've observed a concerning trend: the gradual disappearance of local multiplayer experiences, those precious moments when players gather physically in the same room, sharing a screen and creating memories together.

The gaming industry's evolution toward complexity and online-centric experiences has inadvertently diminished the social, in-person gaming culture that defined earlier generations. Many of us fondly remember evenings spent playing Mario Kart with family members or friends during our childhood—moments that forged genuine connections and created memories that persist decades later. We believe this type of shared gaming experience is irreplaceable and must be preserved in modern game design.

1.3 Vision and Core Philosophy

Our primary objective is to develop a game that rekindles this communal gaming atmosphere while adapting it to contemporary expectations. We aim to create an experience that prioritizes human interaction and brings people together physically, even while maintaining the option for competitive play at higher skill levels.

The Mario Kart series perfectly exemplifies our target versatility: some players enjoy it casually with their children, parents, or friends, while others compete in organized tournaments, pursuing mastery and championship titles. We aspire to replicate this spectrum of engagement, ensuring our game appeals to both casual social players and dedicated competitors.

---

2. PROJECT SCOPE

2.1 Stakeholders and Target Audience

Target Demographics

Our game is designed to appeal to an exceptionally broad demographic range:

Age Groups:

- **Children (6-12 years):** Simple controls and colorful visual design make the game accessible to younger players
- **Teenagers (13-19 years):** Fast-paced action and competitive elements appeal to this demographic's preferences
- **Adults (20-50 years):** Nostalgic elements and social aspects cater to this group's gaming habits
- **Seniors (50+ years):** Intuitive mechanics ensure accessibility for less experienced gamers

Player Profiles:

- **Casual/Social Gamers:** Players seeking entertainment for social gatherings, family events, or relaxation

- **Competitive Players:** Individuals interested in mastering game mechanics, optimizing strategies, and competing at high levels
- **Adventure Seekers:** Players who enjoy exploration, discovering map secrets, and experiencing the chase dynamic from both hunter and hunted perspectives

Stakeholder Identification

Primary Stakeholders:

- Development team members (ourselves) who will conduct continuous internal testing throughout development
- Selected friends and acquaintances who will participate in playtesting sessions
- End users who will provide valuable feedback through early access programs

Secondary Stakeholders:

- Technical advisors and mentors providing guidance on game development best practices
- Potential future publishers or distribution platforms

2.2 General Product Description

Visual Theme and Aesthetic Direction

We have committed to a futuristic aesthetic inspired by cyberpunk and neo-noir visual styles. Our game world features high-speed pursuit gameplay set in futuristic metropolitan environments during nighttime hours. The setting emphasizes verticality, with gameplay occurring one top of towers and buildings and across interconnected rooftops.

The visual palette is deliberately contrasted: predominantly dark urban environments illuminated by vibrant neon lighting, holographic advertisements, and luminescent architectural elements. This aesthetic choice draws heavy inspiration from titles like Cyberpunk 2077, which we admire for both artistic merit and thematic coherence. The cyberpunk aesthetic aligns perfectly with our game's narrative context and provides opportunities for visually striking environmental design.

Narrative Context and Lore

Our game is set in a near-future society where traditional sports have evolved into extreme urban spectacles. In this world, "Chase Tag" has become a mainstream competitive sport, broadcast globally and attracting massive audiences. Professional runners compete in sanctioned events across megacity rooftops, turning parkour and pursuit into entertainment phenomena.

Players assume the roles of competitors in this high-stakes sport, navigating dangerous urban terrain while hunting or evading their opponent. The sport's popularity has led to the

development of specialized equipment, power-enhancing technologies, and purpose-built urban courses designed for maximum spectacle and challenge.

Core Gameplay Mechanics

Primary Gameplay Loop:

The fundamental gameplay revolves around a dynamic hunter-hunted relationship between two players. One player assumes the "Hunter" role, attempting to tag their opponent, while the other becomes the "Hunted," attempting to evade capture for as long as possible.

When the Hunter successfully tags the Hunted, roles immediately reverse, the former Hunter becomes the new Hunted, and vice versa. This role-swapping mechanic ensures continuous engagement for both players and prevents either role from becoming stale or frustrating.

Power-Up System:

Strategic collectible power-ups are distributed throughout the map, providing temporary advantages to players who successfully acquire them. These power-ups serve multiple gameplay functions:

- Rewarding map knowledge and exploration
- Creating dynamic risk-reward scenarios (pursuing a power-up may expose you to capture)
- Introducing tactical variety and preventing matches from becoming predictable
- Balancing skill disparities between players of different experience levels

Importantly, power-ups provide advantages but do not guarantee victory, skilled players can still succeed through superior movement, strategy, and game sense even without power-up assistance.

Scoring and Victory Conditions:

Each player has an individual timer that accumulates time spent in the "Hunted" role. The player with the highest accumulated "Hunted time" at the end of a round wins that round. This scoring system incentivizes consistent evasion rather than a single lucky escape.

Match Structure:

- Each complete game consists of 3 rounds
- Each round lasts 5 minutes
- The player who wins the most rounds (2 out of 3) wins the overall match
- Tiebreaker rules apply if necessary

Technical Implementation

Development Platform:

- Programming Language: Python
- Game Engine: Panda3D
- Target Operating Systems: Windows 10 and Windows 11

Optimization Philosophy:

Given our commitment to accessibility, we are prioritizing code optimization to ensure smooth performance across a wide range of hardware configurations. Our goal is to make the game playable on modest laptop specifications without requiring high-end gaming hardware. This optimization work includes:

- Efficient rendering techniques to maintain stable frame rates
- Memory management optimization to reduce resource consumption
- Level-of-detail (LOD) systems for 3D assets
- Streamlined physics calculations

---

3. FUNCTIONAL REQUIREMENTS

3.1 Core Gameplay Controls

Movement Controls

- **W Key:** Forward movement/running
- **A Key:** Turn left/strafe left
- **D Key:** Turn right/strafe right
- **Space Bar:** Jump (vertical movement for traversing obstacles and gaps)
- **Shift Key:** Crouch/slide (for navigating low spaces or evasive maneuvers)

Interaction Controls

- **E Key:** Collect/pick up power-up items
- **R Key:** Activate/use collected power-up
- **Q Key:** Tag opponent (Hunter only)

System Controls

- **Escape Key:** Open in-game menu (pause game, access settings)

3.2 Power-Up System (Examples)

The following power-ups represent our initial design concepts, subject to balance testing and iteration:

1. **Double Jump:** Grants one additional mid-air jump, enabling access to otherwise unreachable areas
2. **Freeze Ray:** Temporarily immobilizes opponent for 3 seconds
3. **Speed Dash:** Brief burst of increased movement speed
4. **Invisibility Cloak:** 5 seconds of visual invisibility
5. **Shield:** Provides one-time protection against being tagged
6. **Smoke Bomb:** Creates vision-obscuring cloud to facilitate escapes
7. **Grappling Hook:** Enables rapid vertical or horizontal movement to specific anchor points
8. **Radar Pulse:** Briefly reveals opponent's location on the map

3.3 Scoring and Timer System

Timer Mechanics:

- Each player has an individual accumulation timer
- Timer activates immediately when player assumes "Hunted" role
- Timer pauses when player becomes "Hunter"
- At round end, player with highest accumulated time wins that round

Victory Conditions:

- Best of 3 rounds determines match winner
- In case of 1-1 tie, sudden death third round occurs
- Additional tiebreaker mechanics may be implemented if needed

3.4 User Interface Elements

Heads-Up Display (HUD)

- **Score Table:** Positioned at top-center of screen, displays current round wins for each player
- **Personal Timer:** Shows player's current accumulated "Hunted" time in current round
- **Role Indicator:** Clear visual indication of whether player is currently Hunter or Hunted
- **Power-Up Inventory:** Icon showing currently held power-up (if any)
- **Minimap :** A little map on the bottom right of the screen to have a global vision of the map

In-Game Menu (Accessed via Escape Key)

- **Audio Settings:**
    - Sound effects volume slider
    - Music track selection
    - Control the volume of the music

- **Game Controls:**
    - Resume game button
    - Restart match button
    - Return to main menu button
    - Exit game button
- **Settings:**
    - Controls remapping options
    - Display resolution options

Main Menu System

Initial Launch Screen:

- Game title and logo
- "PLAY" button
- "SETTINGS" button
- "CREDITS" button
- "EXIT" button

Play Submenu:

- **Host Game:** Generate unique lobby code for friends to join
- **Join Game:** Enter lobby code to join friend's hosted game

3.5 Multiplayer Architecture

Lobby System:

- Host player generates a unique alphanumeric room code
- Room code is shareable via external communication (text, voice chat, etc.)
- Joining player enters code to connect to hosted game

Network Requirements:

- Stable internet connection required for both players
- Client-server architecture with one player acting as host
- Latency compensation mechanisms to ensure fair gameplay
- Reconnection protocols for handling disconnections

3.6 Stretch Goals and Optional Features

If development time and resources permit, we may implement:

- **Additional Power-Ups:** Expanded variety for greater strategic depth
- **Customizable Match Settings:** Player-adjustable round duration, power-up spawn rates

- **Multiple Maps:** Different urban environments with unique layouts and aesthetics
- **Character Customization:** Cosmetic options for player avatars

---

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Assumptions and Dependencies

System Requirements and User Assumptions

Player Competencies:

- Players possess basic English language comprehension (all UI and instructions are in English)
- Players have fundamental computer literacy (file navigation, game installation)
- Players understand basic gaming conventions (menus, controls, objective-based gameplay)

Technical Requirements:

- Minimum Operating System: Windows 10 or Windows 11
- Active internet connection required for multiplayer functionality
- Minimum hardware specifications:
    - Processor: Dual-core CPU (2.0 GHz or higher)
    - RAM: 4GB minimum
    - Graphics: Integrated graphics capable of basic 3D rendering
    - Storage: 500MB available space
    - Network: Broadband internet connection

External Dependencies

Development Tools:

- **GitHub:** Version control and collaborative code management
- **Panda3D Game Engine:** Core rendering and game logic framework
- **Blender:** 3D asset creation and modeling
- **Python Libraries:** Networking, audio, and utility functions

Feedback Channels:

- Playtester availability for user feedback sessions
- Regular team meetings for internal evaluation
- Potential community forums or Discord server for broader feedback

4.2 Performance Requirements

- **Frame Rate Target:** Minimum 30 FPS on minimum specification hardware; 60 FPS on recommended specifications
- **Network Latency:** Playable with up to 150ms ping; optimal at sub-50ms
- **Load Times:** Map loading should complete within 15 seconds maximum
- **Memory Usage:** Maximum 2GB RAM consumption during gameplay

4.3 Reliability and Stability

- **Crash Rate Target:** Less than 1 crash per 10 hours of gameplay
- **Bug Severity Classification:** Critical bugs must be resolved before release; minor bugs documented for post-launch patches
- **Save State:** Game state preservation if unexpected disconnection occurs

4.4 Usability and Accessibility

- **Control Responsiveness:** Input lag below 100ms
- **UI Clarity:** All interface elements clearly legible at standard resolutions
- **Tutorial Quality:** New players should understand core mechanics within 5 minutes
- **Colorblind Considerations:** UI elements distinguishable for common color vision deficiencies

---

5. ACCEPTANCE AND VALIDATION CRITERIA

5.1 Testing Methodology

Internal Testing Phases:

Our quality assurance process involves multiple testing stages throughout development:

Alpha Testing (Development Team):

- Continuous testing during feature implementation
- Immediate bug identification and documentation
- Mechanics validation against design specifications
- Performance profiling on various hardware configurations

Beta Testing (External Playtesters):

- Structured playtesting sessions with friends and volunteers
- Controlled environment testing to observe player behavior
- Feedback collection through surveys and interviews

- Balance testing with players of varying skill levels

Stress Testing:

- Network stability under various connection conditions
- Performance testing on minimum specification hardware
- Extended play sessions to identify gameplay fatigue or exploitation
- Edge case scenario testing

5.2 Quality Metrics

Bug Tracking:

- All identified bugs logged in tracking system with severity classification
- Critical bugs (crashes, game-breaking issues) prioritized for immediate resolution
- Non-critical bugs addressed based on impact and feasibility

Gameplay Balance:

- Power-ups should provide advantage without guaranteeing victory
- No single strategy should dominate all others
- Both Hunter and Hunted roles should feel engaging and fair
- Match outcomes should reflect player skill more than random chance

User Experience Metrics:

- Player retention: Do playtesters want to play multiple matches?
- Learning curve: How quickly do new players grasp core mechanics?
- Frustration points: What elements cause player dissatisfaction?
- Enjoyment rating: Subjective satisfaction scores from playtesters

5.3 Iterative Improvement Process

Version Control and Iteration:

For each development version, we will maintain detailed documentation:

- **Changelog:** All modifications, additions, and removals
- **Pros Analysis:** Features and elements that are working well
- **Cons Analysis:** Issues, problems, and areas needing improvement
- **Feedback Summary:** Compiled playtester comments and observations

This iterative approach ensures continuous improvement while preserving successful elements. Our goal is to systematically refine the game through multiple versions, ultimately delivering the highest quality product achievable within our constraints.

Key Performance Indicators:

- Playtest satisfaction scores
- Bug frequency and severity trends
- Feature completion percentage
- Performance benchmark results
- Balance metrics from competitive play

---

6. SCHEDULE AND ESTIMATION

Phase 1: Learning and Prototyping

Duration: Mid-October to Early January (~11 weeks)

Primary Objectives:

- Master essential development tools and technologies
- Establish foundational game systems
- Create functional vertical slice

Specific Deliverables:

- Panda3D engine proficiency among all team members
- Network programming fundamentals implementation
- Blender 3D modeling skills acquisition
- Functional main menu system
- Basic multiplayer server infrastructure
- Music integration system
- Points and scoring logic
- Preliminary website for project documentation

Challenges Encountered: Blender's learning curve proved steeper than initially anticipated, resulting in delays to 3D asset production. This bottleneck required schedule adjustments and task reprioritization to maintain overall timeline integrity.

Phase 2: Graphics Development and AI Integration

Duration: Early January to End of March (~12 weeks)

Primary Objectives:

- Implement visual assets and environmental design
- Develop artificial intelligence systems
- Integrate all game systems cohesively

Specific Deliverables:

- Complete 3D asset library (characters, environments, props)
- Primary gameplay map fully designed and implemented
- Finalized and polished menu system with all functionality
- AI programming for dynamic behaviors and pathfinding
- System integration ensuring all components communicate properly
- Visual effects and particle systems

Key Milestone: Mid-term presentation scheduled for end of March, showcasing integrated systems and demonstrating core gameplay loop.

Phase 3: Integration, Testing, and Finalization

Duration: End of March to End of May (~9 weeks)

Primary Objectives:

- Comprehensive testing and quality assurance
- Bug resolution and gameplay refinement
- Final polish and presentation preparation

Specific Deliverables:

Testing Phase:

- Internal bug hunting across all systems
- Performance optimization for target hardware specifications
- Multiple playtesting sessions with external users
- Structured feedback collection and analysis
- Balance adjustments based on feedbacks

Finalization Phase:

- Critical bug fixes and stability improvements
- Audio polish (sound effects, music mixing)
- Animation refinement and visual polish
- UI/UX improvements based on feedback
- Final presentation materials preparation

Final Milestone: Comprehensive final presentation scheduled for end of May, demonstrating completed product and documenting development journey.

---

7. APPENDICES

7.1 Glossary of Technical Terms

Power-Up: A collectible item that temporarily enhances a player's abilities (speed, health, special moves, etc.).

Bug: An error in the code causing unintended behavior, glitches, or crashes in the game.

Game Engine: A software framework providing core tools for game development (graphics, physics, sound). Examples: Pygame, Panda3D, Unity.

Panda3D: An open-source game engine for 3D game development, popular in education and research.

Code Optimization: The process of improving code performance to increase speed and reduce memory usage without changing functionality.

Frame Rate (FPS): The number of images displayed per second. Higher FPS means smoother gameplay (30 FPS minimum, 60 FPS optimal).

Latency/Ping: The delay between a player's action and the server's response, measured in milliseconds. Lower is better for multiplayer.

Client-Server Architecture: A network setup where one player hosts the game (server) a

7.2 References and Resources

Official Documentation:

- Panda3D Documentation: https://docs.panda3d.org/1.10/python/index
- Python Official Tutorial: https://www.w3schools.com/python/

Community Resources:

- YouTube tutorials and gamedev channels
- GitHub repositories and example projects
- Stack Overflow Q&A community: https://stackoverflow.com/questions

Development Tools:

- GitHub for version control and collaboration
- Blender for 3D asset creation
- Discord for team communication
