**Game Programming Document: Project Fantasy**

**Title:**

**Version: 1.0**

**Date: [Date]**

[1. Overview 3](#_Toc153494727)

[1.1 Concept 3](#_Toc153494728)

[1.2 Target Platforms 3](#_Toc153494729)

[1.3 Development Team 3](#_Toc153494730)

[3](#_Toc153494731)

[2. Game Mechanics 3](#_Toc153494732)

[2.1 Core Gameplay 3](#_Toc153494733)

[2.2 Scoring System 3](#_Toc153494734)

[3. Programming Language 3](#_Toc153494735)

[3.1 Language 3](#_Toc153494736)

[3.2 Reasoning 3](#_Toc153494737)

[4. Development Environment 3](#_Toc153494738)

[4.1 IDE 3](#_Toc153494739)

[4.2 Unity Version 4](#_Toc153494740)

[5. Core Systems 4](#_Toc153494741)

[5.1 Player Controller 4](#_Toc153494742)

[5.2 Game Manager 4](#_Toc153494743)

[6. Graphics Engine 4](#_Toc153494744)

[6.1 Rendering 4](#_Toc153494745)

[6.2 Animation 4](#_Toc153494746)

[7. Audio System 4](#_Toc153494747)

[7.1 Sound Effects 4](#_Toc153494748)

[7.2 Music 4](#_Toc153494749)

[8. Input Handling 4](#_Toc153494750)

[8.1 Keyboard Controls 4](#_Toc153494751)

[8.2 Gamepad Support 4](#_Toc153494752)

[9. Networking 5](#_Toc153494753)

[9.1 Single-Player Focus 5](#_Toc153494754)

[9.2 Potential Future Features 5](#_Toc153494755)

[10. Artificial Intelligence 5](#_Toc153494756)

[10.1 Enemy AI 5](#_Toc153494757)

[11. Optimization 5](#_Toc153494758)

[11.1 Performance 5](#_Toc153494759)

[12. Debugging Tools 5](#_Toc153494760)

[12.1 Unity Profiler 5](#_Toc153494761)

[12.2 Debug Logs 5](#_Toc153494762)

[13. Coding Conventions 5](#_Toc153494763)

[13.1 Naming Conventions 5](#_Toc153494764)

[13.2 Code Organization 6](#_Toc153494765)

[14. Technical Challenges 6](#_Toc153494766)

[14.1 Procedural Level Generation 6](#_Toc153494767)

[14.2 Dynamic Audio Integration 6](#_Toc153494768)

[15. Testing Plan 6](#_Toc153494769)

[15.1 Playtesting 6](#_Toc153494770)

[15.2 Quality Assurance 6](#_Toc153494771)

[16. Version Control 6](#_Toc153494772)

[16.1 Repository 6](#_Toc153494773)

[16.2 Branching Strategy 6](#_Toc153494774)

[17. Monetization Strategy 6](#_Toc153494775)

[17.1 Free-to-Play Model 6](#_Toc153494776)

[17.2 Ads Placement 6](#_Toc153494777)

[18. Marketing Integration 7](#_Toc153494778)

[18.1 social media 7](#_Toc153494779)

[18.2 Collaboration 7](#_Toc153494780)

[19. Credits 7](#_Toc153494781)

[19.1 Development Team 7](#_Toc153494782)

[19.2 Special Thanks 7](#_Toc153494783)

# 1. Overview

### 1.1 Concept

CyberRunner is a futuristic endless runner game set in a cyberpunk world. Players control a character running through a neon-lit cityscape, dodging obstacles, and collecting power-ups.

### 1.2 Target Platforms

* Initially: Windows, macOS, Linux
* Potential for expansion to mobile platforms

### 1.3 Development Team

* Lead Programmer: [Your Name]
* Additional programmers, artists, and designers

# 

# **2. Game Mechanics**

### 2.1 Core Gameplay

* Endless runner with procedural level generation
* Player-controlled character with basic actions (jump, slide, dash)
* Obstacle avoidance and power-up collection

### 2.2 Scoring System

* Distance traveled and power-ups collected contribute to the score
* Leaderboards for friendly competition

# 3. Programming Language

### 3.1 Language

* C# with Unity framework

### 3.2 Reasoning

* Unity's robust features and ease of use
* C# for its readability and performance

# 4. Development Environment

### 4.1 IDE

* Visual Studio

### 4.2 Unity Version

* Unity 202X.X

# 5. Core Systems

### 5.1 Player Controller

* Character movement and actions
* Input handling for keyboard and potential gamepad support

### 5.2 Game Manager

* Controls game flow, level generation, and scoring

# 6. Graphics Engine

### 6.1 Rendering

* Unity's built-in rendering pipeline
* Utilize shaders for cyberpunk aesthetics

### 6.2 Animation

* Character animations and dynamic environment elements

# 7. Audio System

### 7.1 Sound Effects

* Footsteps, power-up pickups, obstacle collisions
* Unity's Audio Source for playback

### 7.2 Music

* Dynamic music to match the pace of the game
* Layered tracks for variation

# 8. Input Handling

### 8.1 Keyboard Controls

* Arrow keys for movement
* Spacebar for jump, Ctrl for slide, Shift for dash

### 8.2 Gamepad Support

* Unity Input System for gamepad compatibility

# 9. Networking

### 9.1 Single-Player Focus

* No multiplayer features planned for initial release

### 9.2 Potential Future Features

* Global leaderboards and achievements

# 10. Artificial Intelligence

### 10.1 Enemy AI

* Basic obstacle movement patterns
* Adaptive difficulty based on player performance

# 11. Optimization

### 11.1 Performance

* Efficient memory management and asset optimization
* Implementing level-of-detail (LOD) for distant objects

# 12. Debugging Tools

### 12.1 Unity Profiler

* Identify performance bottlenecks
* Monitor memory usage and frame rates

### 12.2 Debug Logs

* Comprehensive logging for runtime information
* Debugging using Visual Studio

# 13. Coding Conventions

### 13.1 Naming Conventions

* CamelCase for variables, PascalCase for functions and classes
* Descriptive names for clarity

### 13.2 Code Organization

* Modular design with proper use of classes and functions

# 14. Technical Challenges

### 14.1 Procedural Level Generation

* Creating varied and challenging levels dynamically

### 14.2 Dynamic Audio Integration

* Synchronizing music with gameplay events

# 15. Testing Plan

### 15.1 Playtesting

* Regular internal playtests for feedback
* External beta testing for a wider audience

### 15.2 Quality Assurance

* Identify and address bugs through rigorous testing

# 16. Version Control

### 16.1 Repository

* Git for version control
* GitHub for collaboration and backup

### 16.2 Branching Strategy

* Feature branches for development, main branch for stable releases

# 17. Monetization Strategy

### 17.1 Free-to-Play Model

* Ad-supported with optional in-app purchases for cosmetic items

### 17.2 Ads Placement

* Non-intrusive banner ads and rewarded video ads

# 18. Marketing Integration

### 18.1 social media

* Teasers, gameplay snippets, and updates on platforms like Twitter and Instagram

### 18.2 Collaboration

* Partnering with gaming influencers for coverage

# 19. Credits

### 19.1 Development Team

* Lead Programmer: [Your Name]
* Programmers, Artists, Designers

### 19.2 Special Thanks

* Acknowledgments to external contributors and testers