Project Vision

# Title: Network Gameplay Tester

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Degree | Primary Role | Secondary Role |
| Matthew Picioccio | BSCS | Mobile Programmer | Server Programmer |
| Andrew Kaplan | BSCSGD | Client Programmer | UX Designer |

# Project Duration

This project is planned to be completed within one semester. The team may continue it and add features in a second semester.

# Overview

This tool is a mobile tool, written in Unity, allowing the user to visualize a peer-to-peer networked gameplay simulation from their network to their friend’s.

Users will launch the tool and use a platform-specific method to connect to their friend, running the app on another device. The game will then run through a variety of real-time gameplay scenarios, with UI that displays the impact of network misprediction and other errors.

The user will also be able to change the parameters of the simulation, and enable and disable different UI elements for clarity.

Possible extensions include server-hosted sessions, additional network gameplay implementations, additional UI, and additional network simulation settings.

# Genre / Subject

Networked multiplayer gameplay

# Target Users

The target user is a video game developer, or a technically-minded video game player, who would like to use this app to visualize the quality of networked play on arbitrary networks, particularly cellular and Wi-Fi networks. These users would like to see the impact of different choices in network gameplay replication on the user experience, in real-world high-latency scenarios.

# Platform

iOS, and Windows during development.

Users

Using a peer-to-peer model, 2 users at minimum, using network communication. Platform-specific peer-to-peer connection mechanisms will be investigated, but a matchmaking server may be required.

# Inspiration

This tool was inspired by our experiences with odd networked gameplay behavior in various mobile games and in high-latency network environments. Typically, high-latency and low-quality network conditions are tested late in development (or after release), when it is difficult to understand the exact source of user issues. This tool provides a transparent implementation to help developers visualize the variable impact of network conditions on gameplay.

# Core Loop

The user will launch the app and be presented with options to host or connect to a Network Gameplay Tester session. Through this UI, they will connect with another user of the service.

Once a session has begun, the app will automatically start running through gameplay scenarios, such as watching a player firing at another player who is circle-strafing around them. The UI will show what the client believes is happening, what the peer believes is happening, and how the two have been reconciled, using various colors and ghost images. The HUD will also show the current network conditions, including latency (“ping”) and packet loss statistics.

The user will be able to shift to different scenarios and enable different UI options to clarify the UI experience.

If available, the user will also be able to introduce simulated network issues, like packet loss, through an in-session menu.

When the player ends the session, they will see a summary of the issues that were encountered (the number of mispredictions, latency statistics, etc.). The user is then able to start a new session.

# Mockups

# 