Team PUNCHLINE
Test Plan and Result

Overall Test Plan

Our overall plan is to utilize various tests that primarily ensure functionality, consistency, reliability, connection, and safety. For the main tests, we plan to use the integrated *Automated Testing Framework* already built into Unreal Engine 5. For each permutation of gameplay, we will implement the framework while playing that set of rounds. Consistency tests will be based on the overall performance of the game with combinations of characters, stages, and wildcards that may be more prone to stressing a player's systems. To measure reliability, we will be playing the game ourselves throughout a long-term testing session to test for crashing and game uptime. This test will be conducted after the reliability tests to ensure that devs can play their desired character without issue. The safety testing will be conducted via weak attacks on our server. We will be trying to penetrate the instance and manipulate both game and player data to test for network and game vulnerabilities.

Test Case Description

SE.1 **Server Establishment Test**

SE.2 This test ensures that the server is accessible

SE.3 Each of the two separate PCs will connect to the server and then to a lobby through external means with the authorization

SE.4 Inputs: Server ID, Lobby ID

SE.5 Outputs: A successful server ping

SE.6 Normal

SE.7 Blackbox

SE.8 Functional

SE.9 Unit

SD.1 **Server Duel Test**

SD.2 This test ensures that the players can play over the network via the game

SD.3 Each of the two separate PCs will connect to the server through the game application

SD.4 Inputs: Lobby ID

SD.5 Outputs: A successful match connection

SD.6 Normal

SD.7 Blackbox

SD.8 Functional

SD.9 Integration

SS.1 **Server Safety TesS**

SS.2 This test ensures that the server is safe from non-authorized connection

SS.3 A connection will attempt to be established by a non-game owner PC without authorization

SS.4 Inputs: Server ID

SS.5 Outputs: A failed match connection

SS.6 Abnormal

SS.7 Whitebox

SS.8 Functional

SS.9 Unit

CF.1 **Character Functionality Testing**

CF.2 This test will ensure that the button binds trigger the correct commands quickly and ensure that the character's attacks look good

CF.3 Each character will use each move in combat to test if the moves are executed

CF.4 Inputs: Controller inputs, Character Data

CF.5 Outputs: In-game move

CF.6 Normal

CF.7 Blackbox

CF.8 Functional

CF.9 Integration

CT.1 **Character Time Testing**

CT.2 This test will ensure that each attack's frame data is consistent and reliable

CT.3 Each character will use each move in combat to test the frame data

CT.4 Inputs: Controller inputs, Character Data

CT.5 Outputs: Frame Data

CT.6 Normal

CT.7 Whitebox

CT.8 Performance

CT.9 Integration

CI.1 **Character Impact Testing**

CI.2 This test will ensure that the actual in-game hit and hurt boxes interact accurately

CI.3 Each character will be used against each other character with hitbox visualization enabled to ensure successful hit-hurt interaction for each move to each character

CI.4 Inputs: Controller inputs, Character Data

CI.5 Outputs: Damage // hit-hurt connection data

CI.6 Normal

CI.7 Blackbox

CI.8 Functional

CI.9 Integration

SV.1 **Stage Visual Testing**

SV.2 This test will ensure that stages are visually correct in orientation

SV.3 Each character will use each move on each combat stage to prevent clipping and And enhance visual clarity

SV.4 Inputs: Controller inputs, Character Data, Stage Data

SV.5 Outputs: In-game Visual Data

SV.6 Normal

SV.7 Whitebox

SV.8 Performance

SV.9 Integration

WS.1 **Wildcard Success Testing**

WS.2 This test will ensure that the Wildcards do not cause any crashes and work as intended in the game

WS.3 Each character will be tested with the different wild cards on each stage

WS.4 Inputs: Controller inputs, Character Data, Wildcard Setup, Stage Data

WS.5 Outputs: Frame Data, On-screen visuals, Machine performance, Server ping

WS.6 Normal

WS.7 Whitebox

WS.8 Functional

WS.9 Integration

WP.1 **Wildcard Performance Testing**

WP.2 This test will ensure that the Wildcards create a negligible effect on the game performance

WP.3 Each character will be tested with the different wild cards on each stage

WP.4 Inputs: Controller inputs, Character Data, Wildcard Setup, Stage Data

WP.5 Outputs: Frame Data, On-screen visuals, Machine performance, server ping

WP.6 Normal

WP.7 Whitebox

WP.8 Performance

WP.9 Integration

FC1.1 **Full Combat Testing 1**

FC1.2 This test will ensure that two users can connect to the server and play a game

FC1.3 Two players will setup and play a game as normal on the virtual instance specifically testing for crashes and hardware failures

FC1.4 Inputs: Controller inputs, Character Data, Wildcard Setup, Stage Data, Server Data

FC1.5 Outputs: Frame Data, On-screen visuals, Machine performance, Server Ping

FC1.6 Normal

FC1.7 Blackbox

FC1.8 Functional

FC1.9 Integration

FC2.1 **Full Combat Testing 2**

FC2.2 This test will ensure that two users can connect to the server and play a game with comfortable PC performance

FC2.3 Two players will setup and play a game as normal on the virtual instance specifically testing for QoL and fun factor

FC2.4 Inputs: Controller inputs, Character Data, Wildcard Setup, Stage Data

FC2.5 Outputs: Frame Data, On-screen visuals, Machine performance, Server Ping

FC2.6 Normal

FC2.7 Whitebox

FC2.8 Performance

FC2.9 Integration

Overall Test Case Matrix

Test Case ID	**Normal/Ab.**	**Black/Whitebox**	**Funct/Perform**	**Unit/Integr.**
SE	Normal	Black	Funct	Unit
SD	Normal	Black	Funct	Integr.
SS	Normal	White	Funct	Unit
CF	Normal	Black	Funct	Integr.
CT	Normal	White	Perform	Integr.
CI	Normal	Black	Funct	Integr.
SV	Normal	White	Perform	Integr.
WS	Normal	White	Funct	Integr.
WP	Normal	White	Perform	Integr.
FC1	Normal	Black	Funct	Integr.
FC2	Normal	White	Perform	Integr.