Testing Manual

Matthew Durcan, Jaden Tian

2025-12-06

Contents

Introduction

The primary components that may require testing:

- **Game Engine** (`game.cpp`) - Physics, collision detection, and game state management
- **Network Layer** (`server.cpp`, `client.cpp`, `protocol.h`) - Multiplayer synchronization
- **GUI Components** (`game_screen.cpp`, `start_screen.cpp`) - User interface and rendering

It's difficult to test this type of application due to the multiplayer aspect, without multiple computers and people, though we can still do some tests despite this. Though, most of it is done through manual testing.

Test Environment

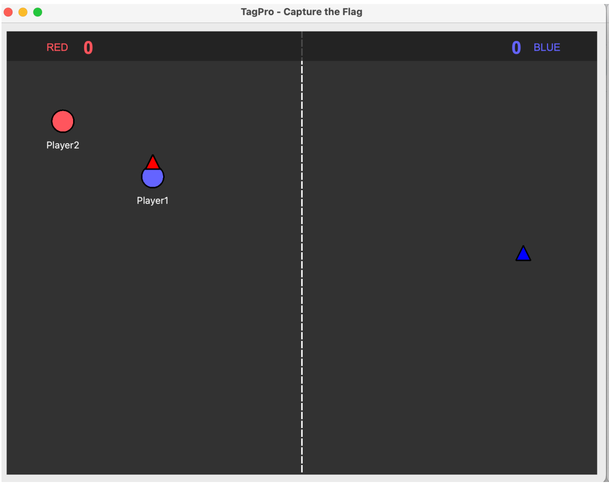
Operating System	Windows 11 / Ubuntu 22.04
Compiler	MinGW-GCC 12.2.0 / GCC 11.3.0
Qt Version	Qt 6.5.0
Build Tools	CMake 3.25, Make/Ninja
Network	Localhost (127.0.0.1)

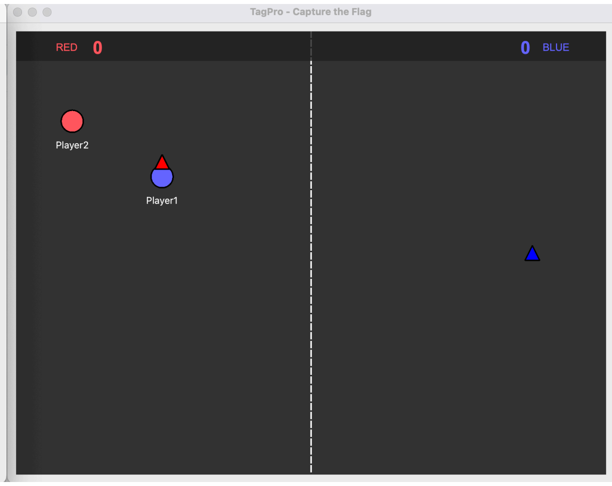
Unit Tests

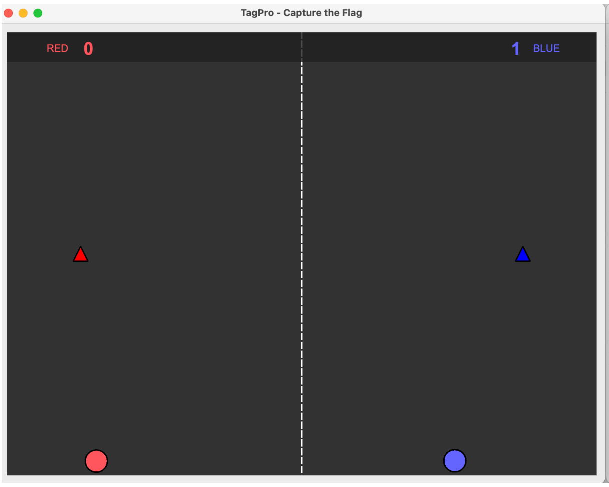
Test Case 1: Flag Capture Mechanics

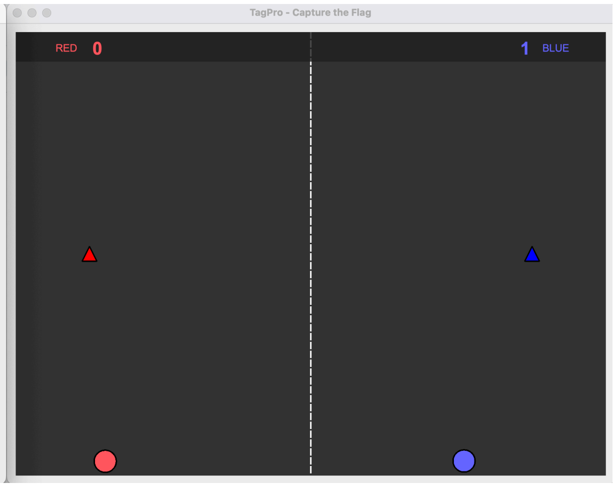
Expected Results:

- Flag disappears from base when picked up
- Flag graphic follows capturing player
- Score increments when flag returned
- Flag resets to base after score or player death









Network Protocol Tests

Test Case 2: Message Framing

Monitor receive buffer parsing

Expected Results:

- All messages received intact
- No buffer overflow or message corruption
- Protocol framing handles partial messages
- The receive buffer is empty, meaning we are receiving one message at a time

Output:

```
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,300,0,0,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,300,0,0,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.985,0,-0.95969,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.954,0,-1.91907,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.908,0,-2.87814,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.846,0,-3.8369,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.754,0,-5.75504,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.647,0,-6.71287,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.524,0,-7.67039,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.386,0,-8.6276,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,700,299.233,0,-9.5845,1,1;2,Player2,100,300,0,0,0,1;
[Client] Receive buffer: , Processing message: 1|0|0|0|0|0|
1,Player1,699.989,299.069,-0.678603,-10.26,1,1;2,Player2,100,300,0,0,0,1;
[Client] Disconnecting from server.
[Client] Receive loop finished.
[Client] Waiting for receivingThread...
[Client] receivingThread has joined.
[Client] Disconnected cleanly.
```

GUI Integration

Test Case 3: Screen Transitions

Tests we considered:

- Returning all clients to home screen if the hosts leaves/closes the lobby
- Returning all clients to home screen after game ends