ECE1724 F1 Performant-Software-Systems-with-Rust

BALL BALL U

Edward S. Rogers Sr. Department of Electrical & Computer Engineering

Litao(John) Zhou - 1006013092

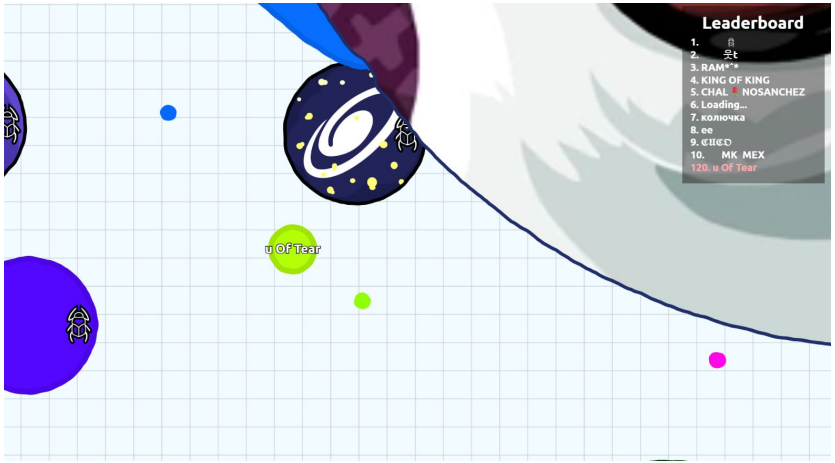
Siyu Shao - 1007147204

Chuyue Zhang - 1005728303

UNIVERSITY OF TORONTO

DEFY GRAVITY

Inspiration (Agar.io and Battle of Balls)



UNIVERSITY OF TORONTO

DEFY GRAVITY

What we want?

- A real-time multiplayer game that share the concept, players control balls that grow by consuming smaller balls of other players and scattered food item
- A software consist of performance, concurrency model, and memory safety

UNIVERSITY OF TORONTO

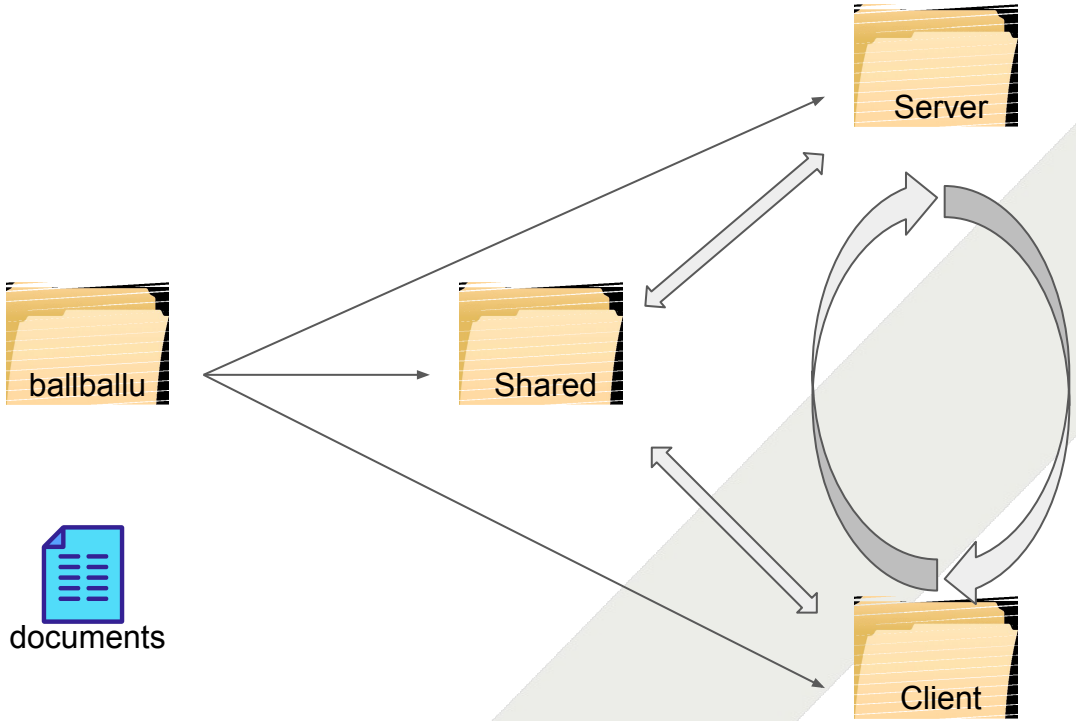
DEFY GRAVITY

RUST

UNIVERSITY OF
TORONTO

DEFY
GRAVITY

Project Structure: Server-Client Structure



Shared - for common shared code

Game Objects

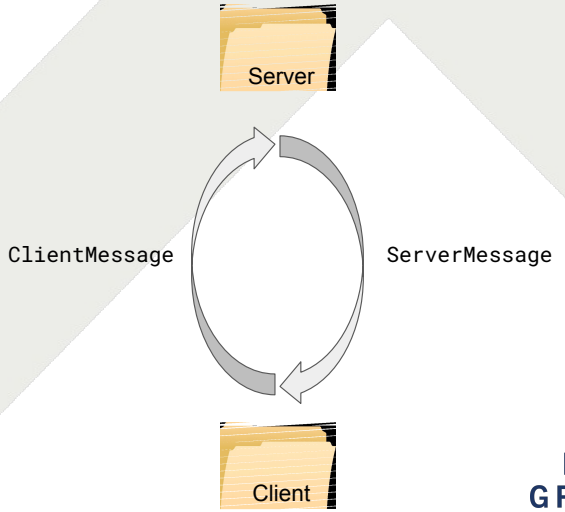
- Player struct: id, name, pos, radius, score, speed, velocity.
- Dot struct: pos, radius, score value (2/5/10), color.
- World boundaries & constants.

Game Mechanics

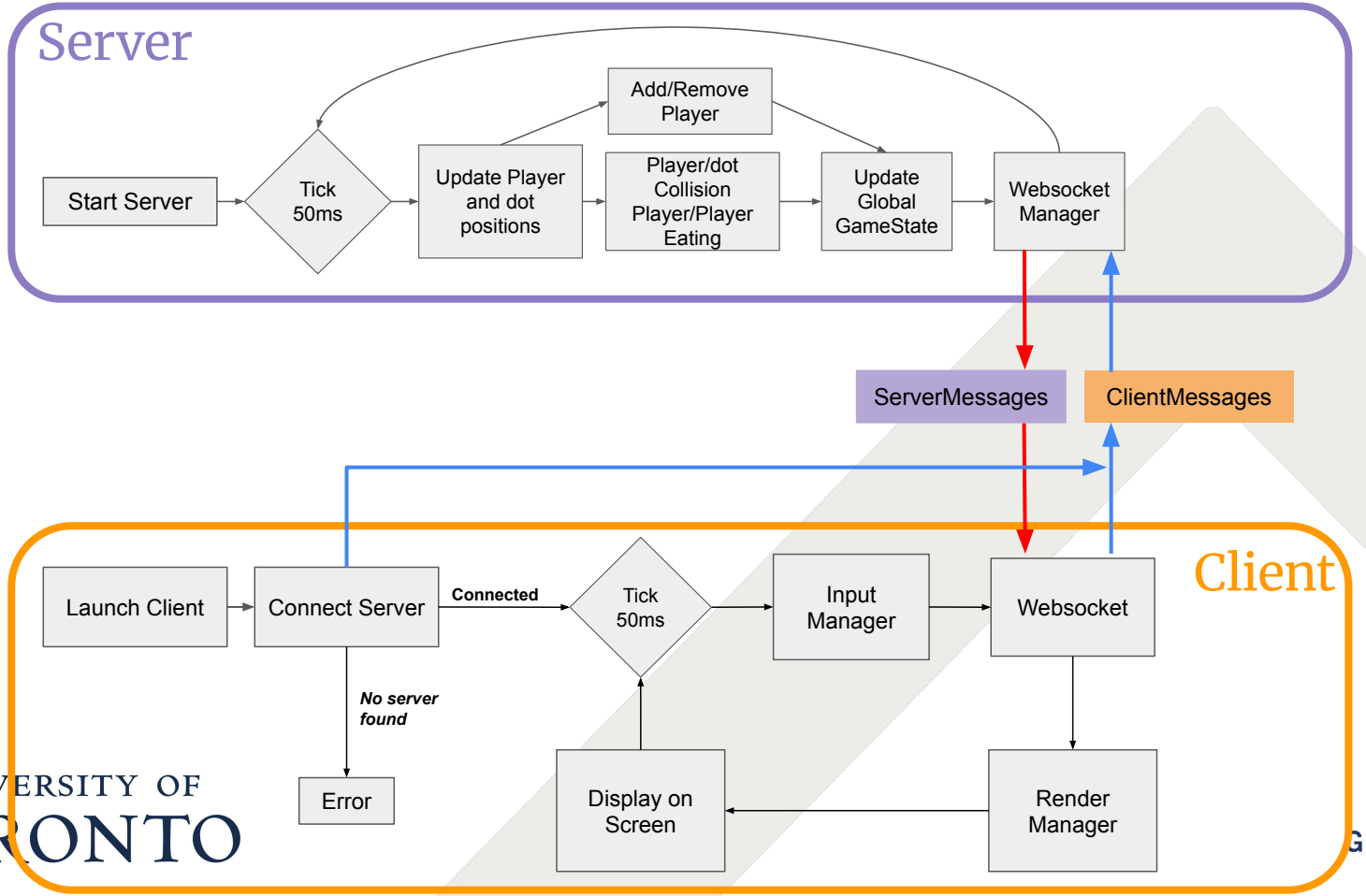
- Size ∝ score
- Speed ∝ 1/score
- Dot eating logic
- Player-vs-player eating logic
- Respawn logic
- Collision detection

Protocol & Serialization

- **`ClientMessage`**: input, join game, etc.
- **`ServerMessage`**: state snapshots, acknowledgments.



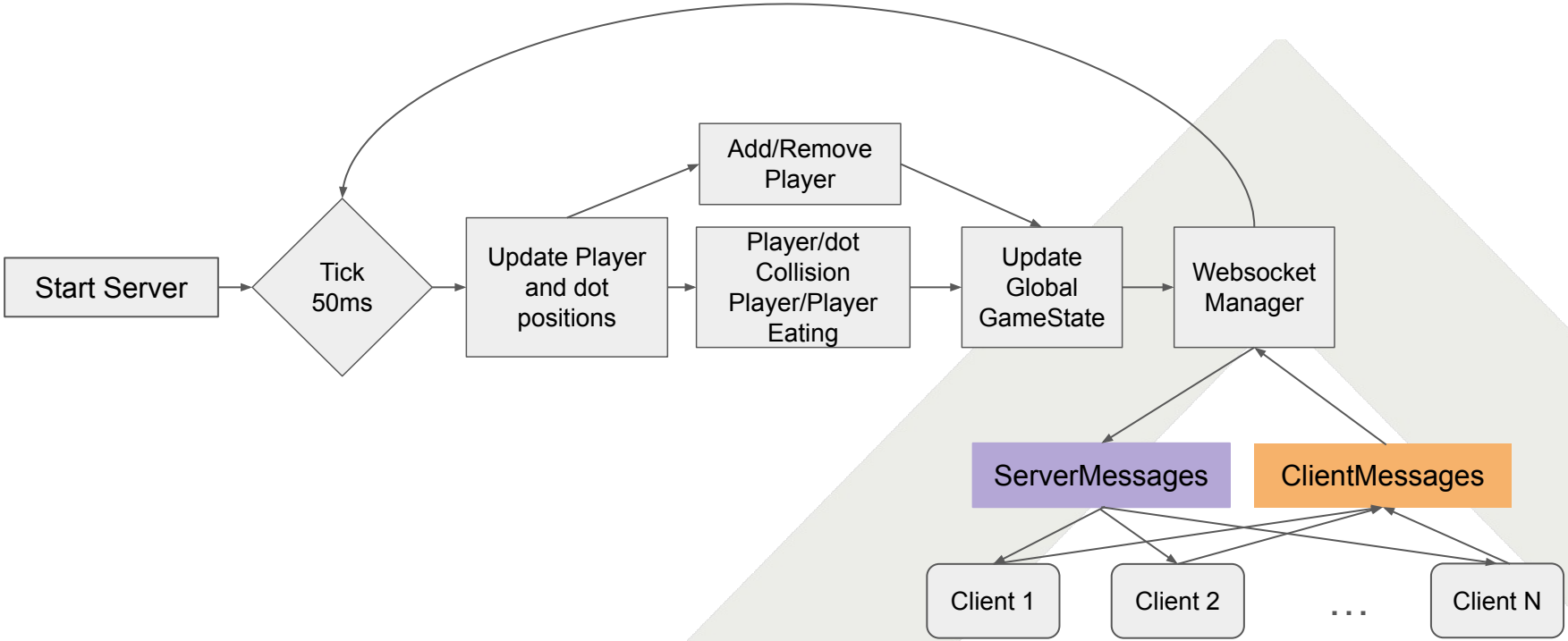
Project Architecture



UNIVERSITY OF TORONTO

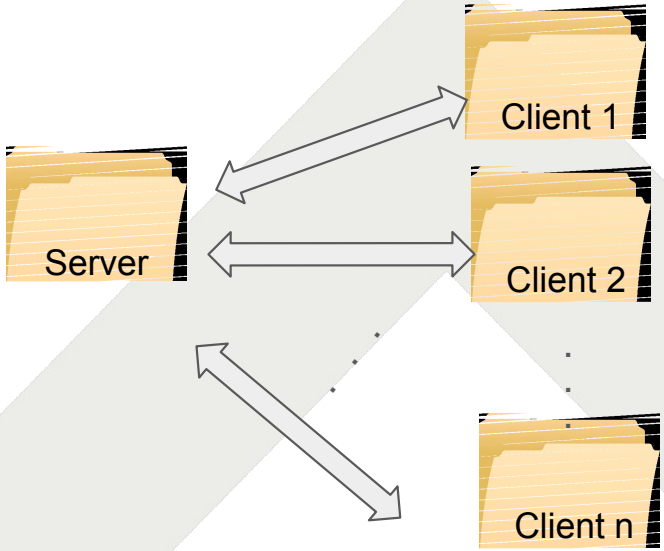
DEFY GRAVITY

Server: Authoritative game engine and networking backend



Server: Authoritative game engine and networking backend

- Uses **tokio + tokio-tungstenite** to handle multiple concurrent WebSocket clients asynchronously.
- Deserializes incoming ClientMessage packets and queues them for the next tick.
- Runs a deterministic 20 Hz tick loop applying game mechanics (movement, collisions, scoring).
- Serializes and broadcasts ServerMessage::StateUpdate snapshots packet to all connected clients.
- Uses **Serde** for all message serialization and shared crate types to ensure consistency with the client.



Server: Authoritative game engine and networking backend

```
[DEBUG] GameLoop: About to broadcast state (tick: 1208)
[DEBUG] GameLoop: About to broadcast state (tick: 1209)
[DEBUG] GameLoop: About to broadcast state (tick: 1210)
[DEBUG] GameLoop: About to broadcast state (tick: 1211)
Player 2 connected!
GameState: Player 2 added at (194.30792, 1384.196)
[DEBUG] GameLoop: About to broadcast state (tick: 1212)
[DEBUG] GameLoop: About to broadcast state (tick: 1213)
[DEBUG] GameLoop: About to broadcast state (tick: 1214)
```

```
[DEBUG] GameLoop: About to broadcast state (tick: 1796)
[DEBUG] GameLoop: About to broadcast state (tick: 1797)
Raw text from 2: {"Join":{"name":""}}
Parsed ClientMessage from 2: Join { name: "" }
GameState: Player 2 final name = AnonymousWizard
[DEBUG] GameLoop: About to broadcast state (tick: 1798)
Raw text from 2: "Ready"
Parsed ClientMessage from 2: Ready
GameState: Player 2 is ready
[DEBUG] GameLoop: About to broadcast state (tick: 1799)
```

New player connecting and starting

```
[DEBUG] GameLoop: About to broadcast state (tick: 1307)
Player 1 ate Player 2
GameState: Player 2 respawned at (1821.3529, 1417.1627)
[DEBUG] GameLoop: About to broadcast state (tick: 1308)
```

Player eating logs

Server: Authoritative game engine and networking backend

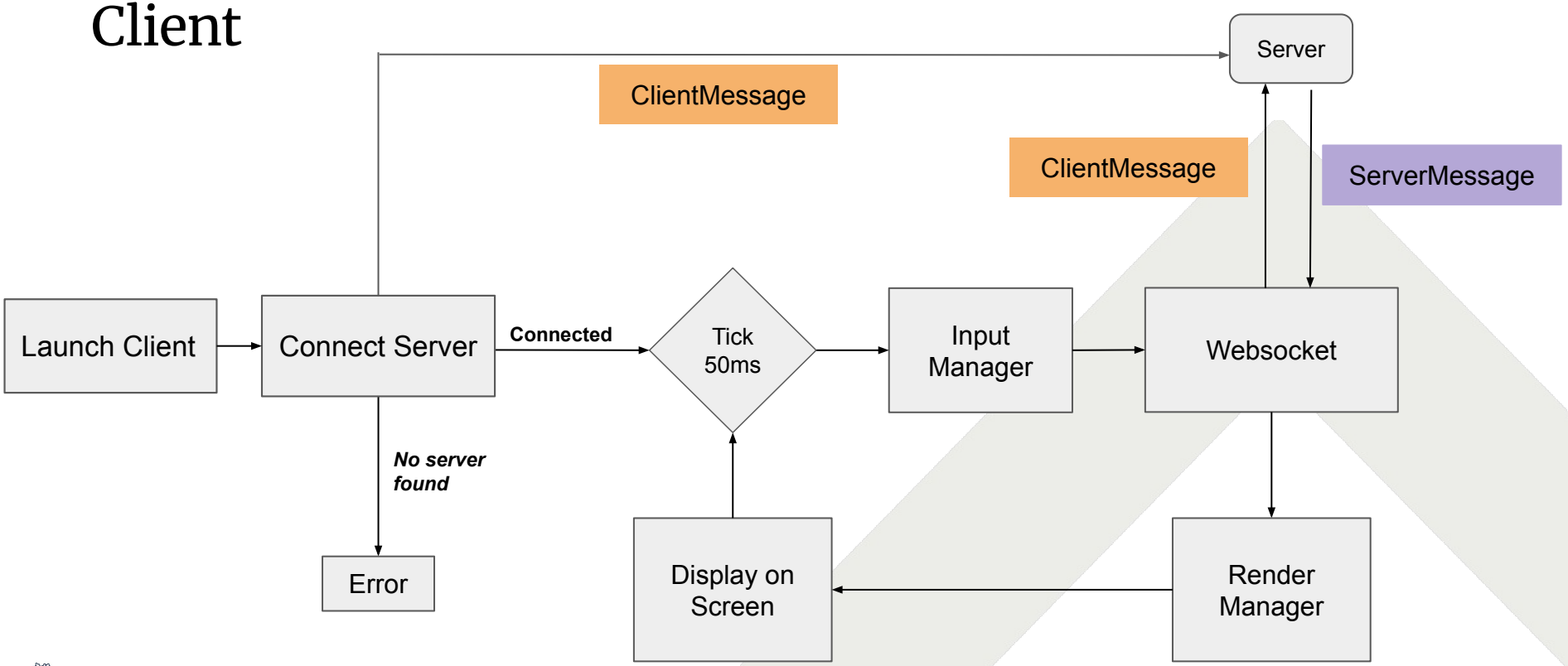
```
[DEBUG] GameLoop: About to broadcast state (tick: 4527)
Raw text from 1: {"Move":{"dx":0.0,"dy":-1.0,"distance":32.165527}}
Parsed ClientMessage from 1: Move { dx: 0.0, dy: -1.0, distance: 32.165527 }
GameState: Player 1 queued move: dx=0, dy=-1, distance=32.165527
[DEBUG] GameLoop: About to broadcast state (tick: 4528)
```

Player moving logs

```
[DEBUG] GameLoop: About to broadcast state (tick: 17828)
WebSocket error from 1: Protocol(ResetWithoutClosingHandshake)
Cleaning up player 1 from GameState
GameState: Player 1 removed
[DEBUG] GameLoop: About to broadcast state (tick: 17829)
```

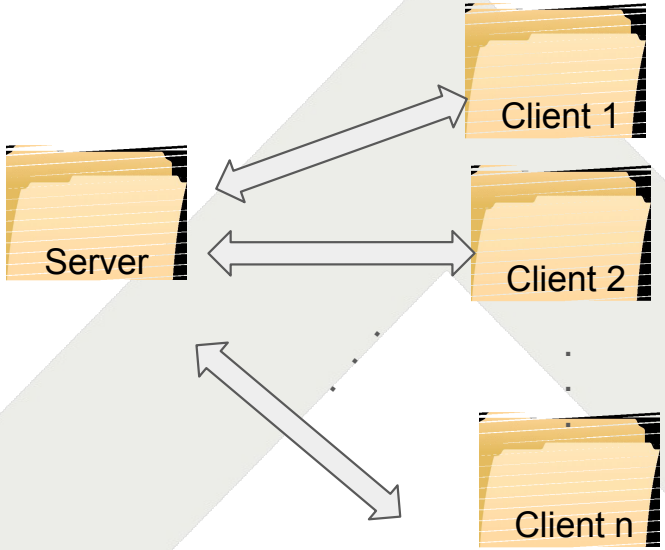
Player quit/disconnected

Client

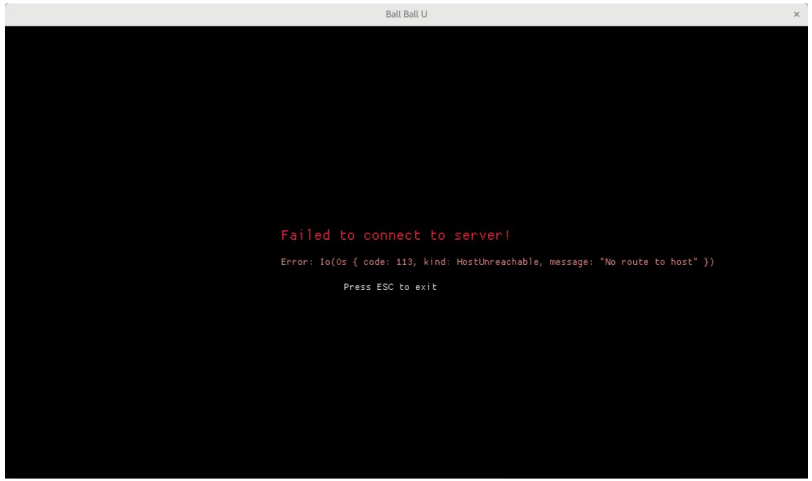

Server
ClientMessage
ClientMessage
ServerMessage
Launch Client
Connect Server
Connected
Tick
50ms
Input
Manager
Websocket
No server
found
Error
Display on
Screen
Render
Manager

Client

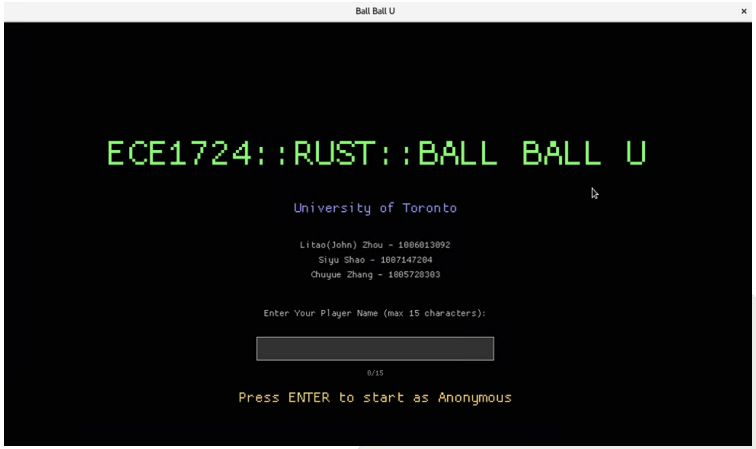
- Uses **tokio-tungstenite (WebSocket)** to send player inputs and receive game snapshots from the server.
- Inputs are serialized using **Serde** and sent as ClientMessage (movement direction + sequence number).
- Receives authoritative ServerMessage::StateUpdate snapshots packet and updates local rendering state.
- Rendering is done using **macroquad**, which draws players, dots, names, leaderboard, and timer.
- Operates in a loop: capture input → send → receive snapshot → render.



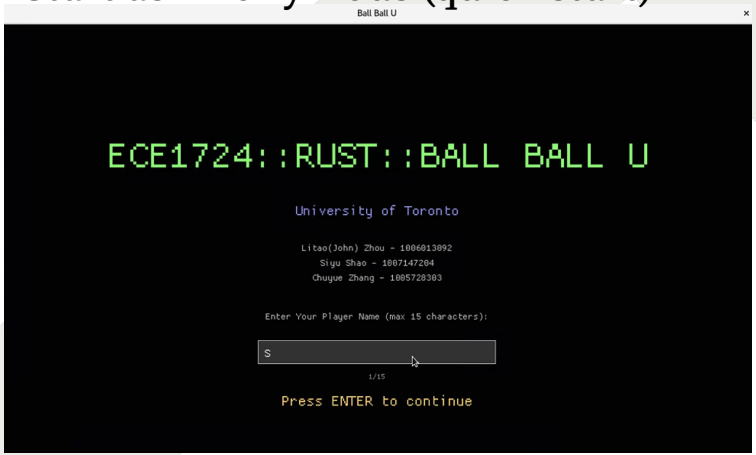
Client: Start game



No server running →
client shows connection error



Start as Anonymous (quick start)

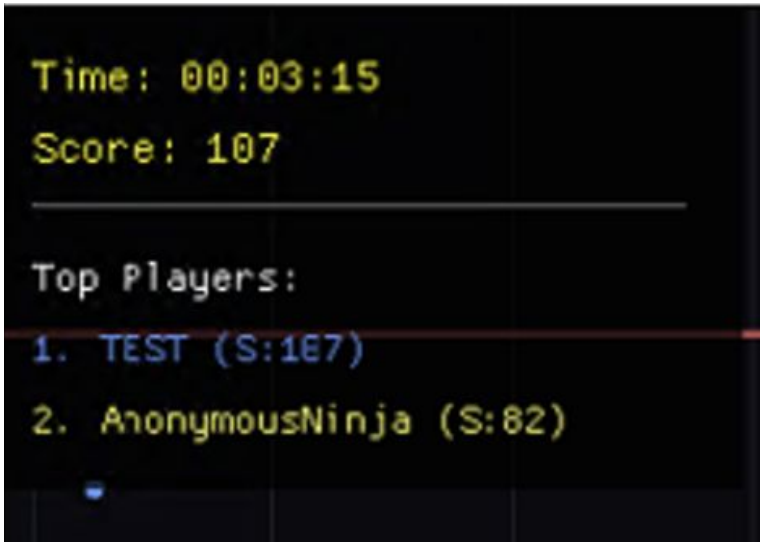


Start with player name (identified user)

UNIVERSITY OF TORONTO

DEFY GRAVITY

Client: In game

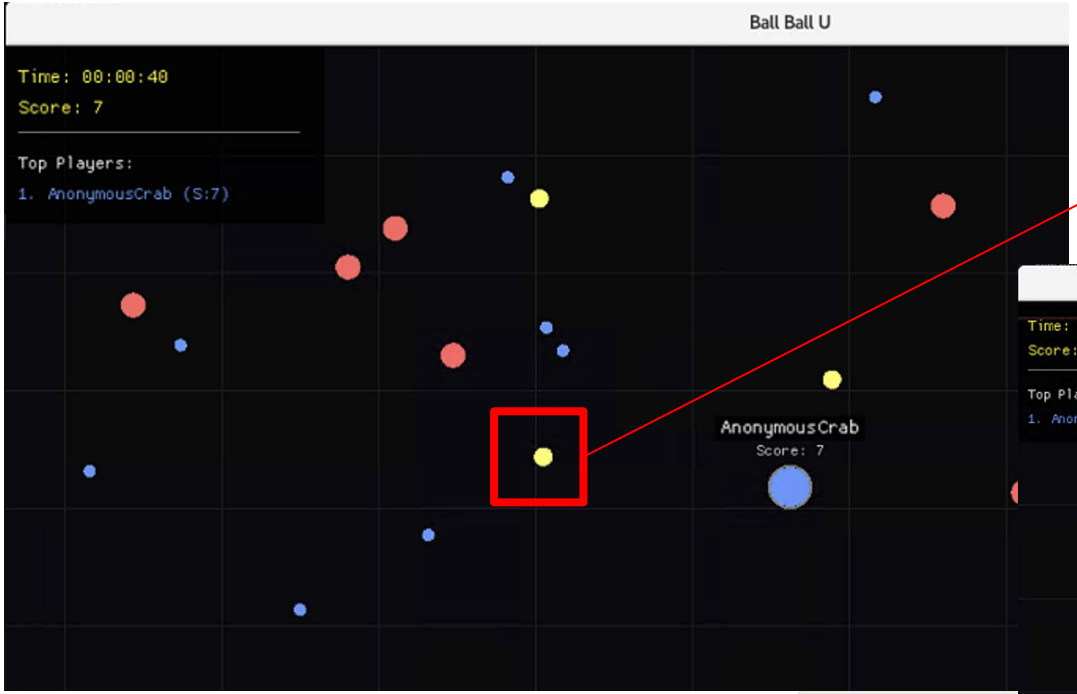


UNIVERSITY OF TORONTO

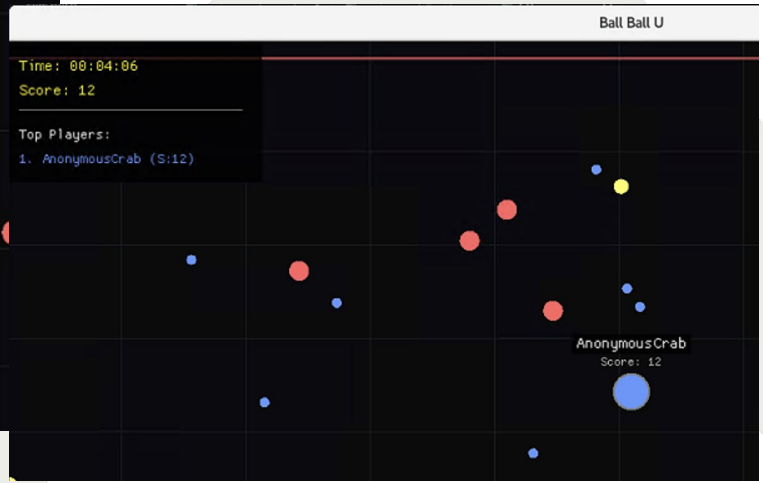
DEFY GRAVITY

Client: Game mechanics

Red dot: 10 score
Yellow dot: 5 score
Blue dot: 2 score



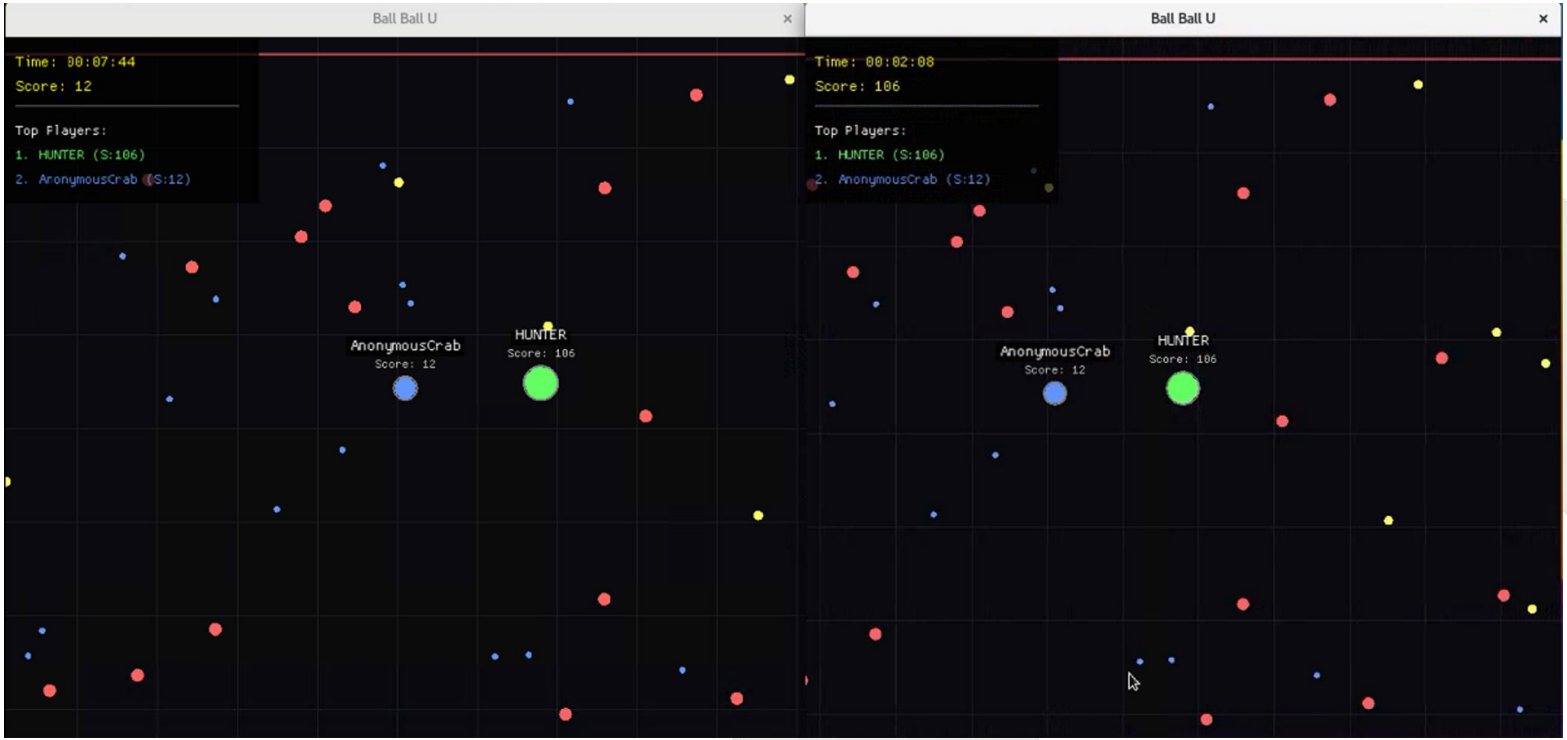
After eating this yellow dot



UNIVERSITY OF TORONTO

DEFY GRAVITY

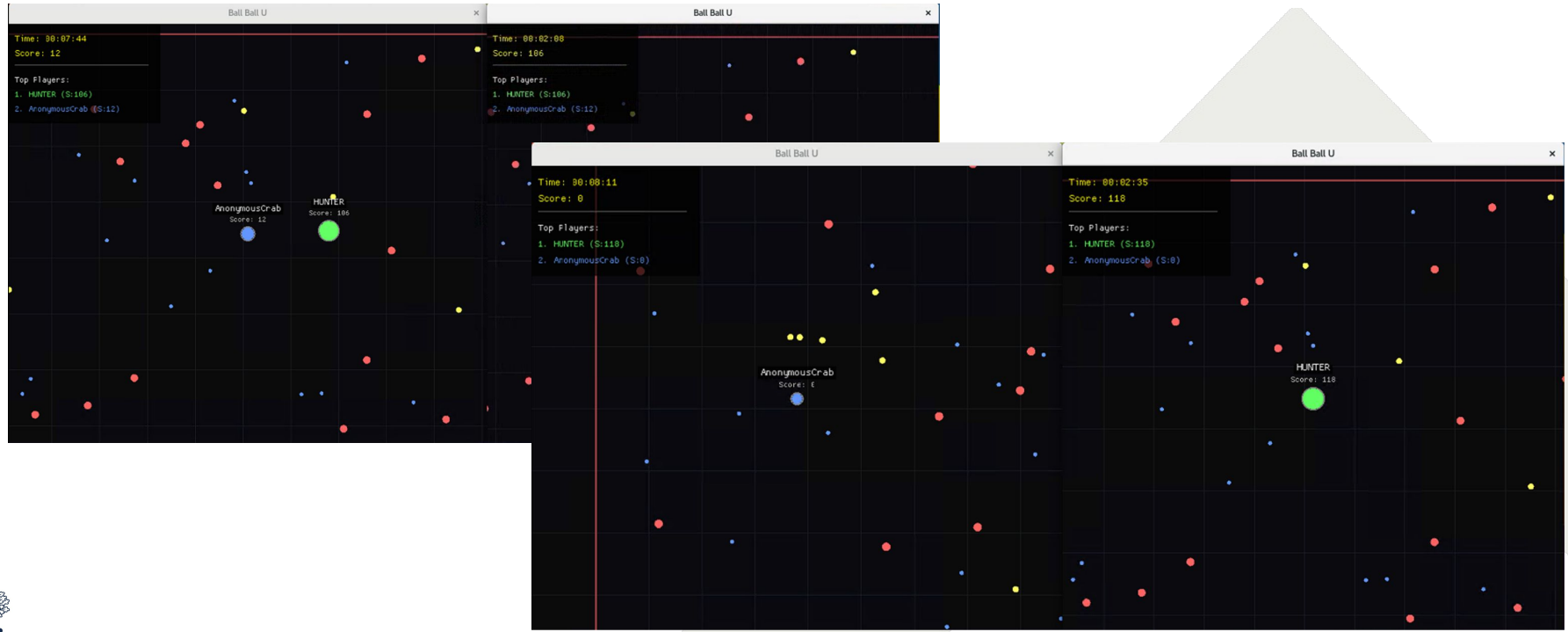
Client: Game mechanics



UNIVERSITY OF TORONTO

DEFY GRAVITY

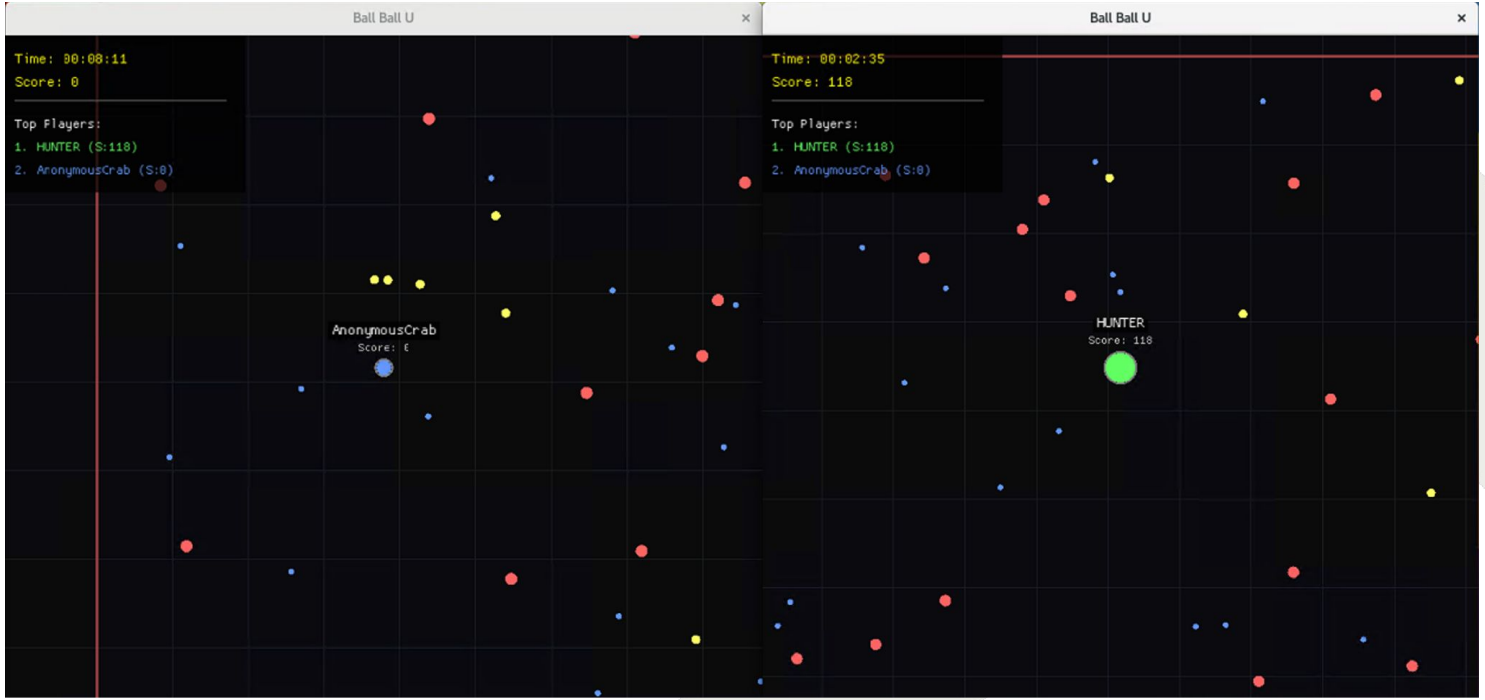
Client: Game mechanics



UNIVERSITY OF TORONTO

DEFY GRAVITY

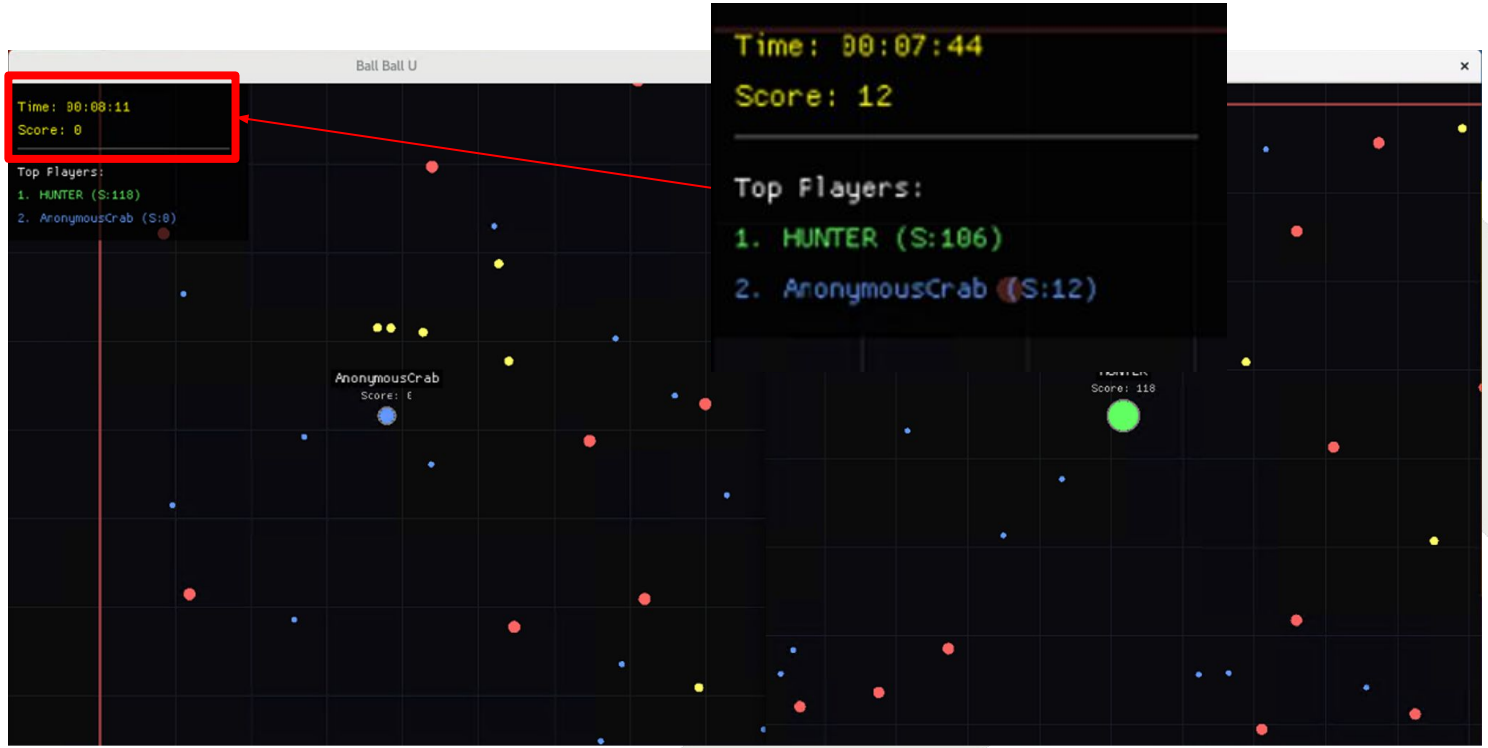
Client: Game mechanics



UNIVERSITY OF TORONTO

DEFY GRAVITY

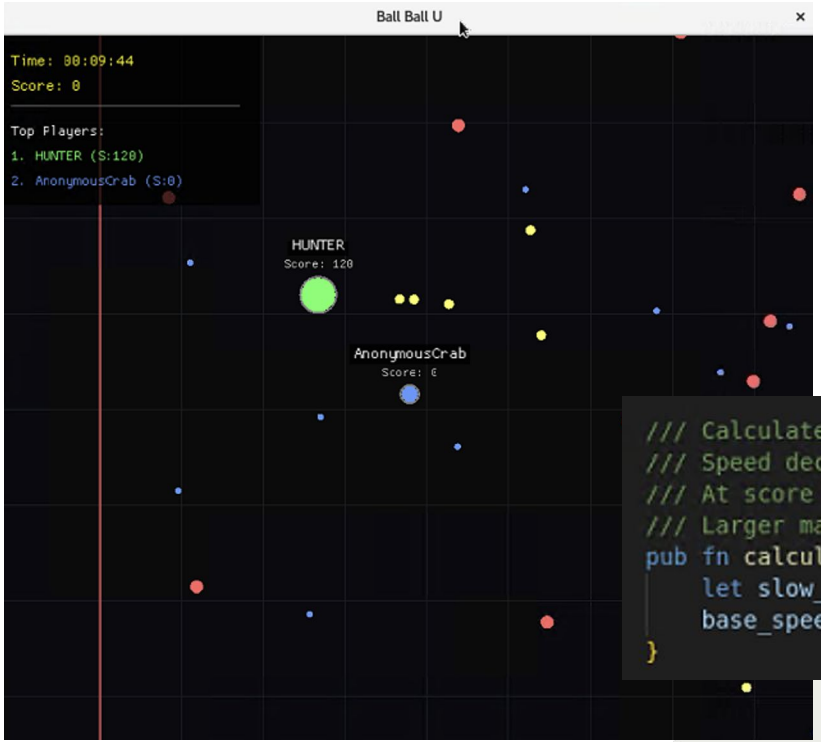
Client: Game mechanics

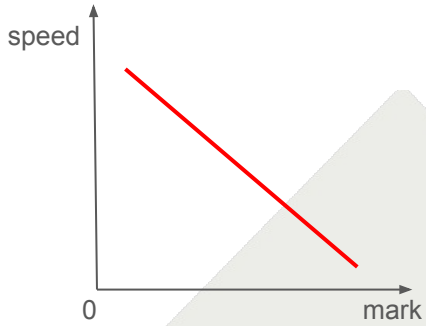


UNIVERSITY OF TORONTO

DEFY GRAVITY

Client: Game mechanics






```rust
/// CalculateSpeedFromScore
/// Speed decreases slowly as player grows larger.
/// At score 0 => base_speed
/// Larger mass => slightly slower movement
pub fn calculate_speed_from_score(score: u32, base_speed: f32) -> f32 {
    let slow_factor = 1.0 / (1.0 + (score as f32) * 0.005);
    base_speed * slow_factor
}
```

UNIVERSITY OF TORONTO

DEFY GRAVITY

Conclusion

- Ball Ball U is a real-time PvP game fully implemented in Rust.
- We combine a shared protocol crate, an async authoritative server, and a macroquad-based graphical client.
- Our design shows that we can build a responsive and fair real-time multiplayer game entirely in Rust.

UNIVERSITY OF TORONTO

DEFY GRAVITY

Thank you for watching!

UNIVERSITY OF TORONTO

DEFY GRAVITY