

# Online Co-Op Bot Synchronization

Heather Barclay

3/14/08

# Co-op Bots

The goal is to 'loosely' sync the Bots (enemies) so that all of the players in a co-op game have approximately the same experience.

# Position / Rotation

Right



Wrong



# Fire Weapon / Projectile

Right



Wrong



# Take Cover

Right

Wrong



# Terms

- Client - machine connected to the game
- Sync – short for (network) synchronization (to make happen at the same time). The past tense of ‘sync’ is not ‘sunk’
- Local - first-person
- Remote - third-person



# 3 Main Points

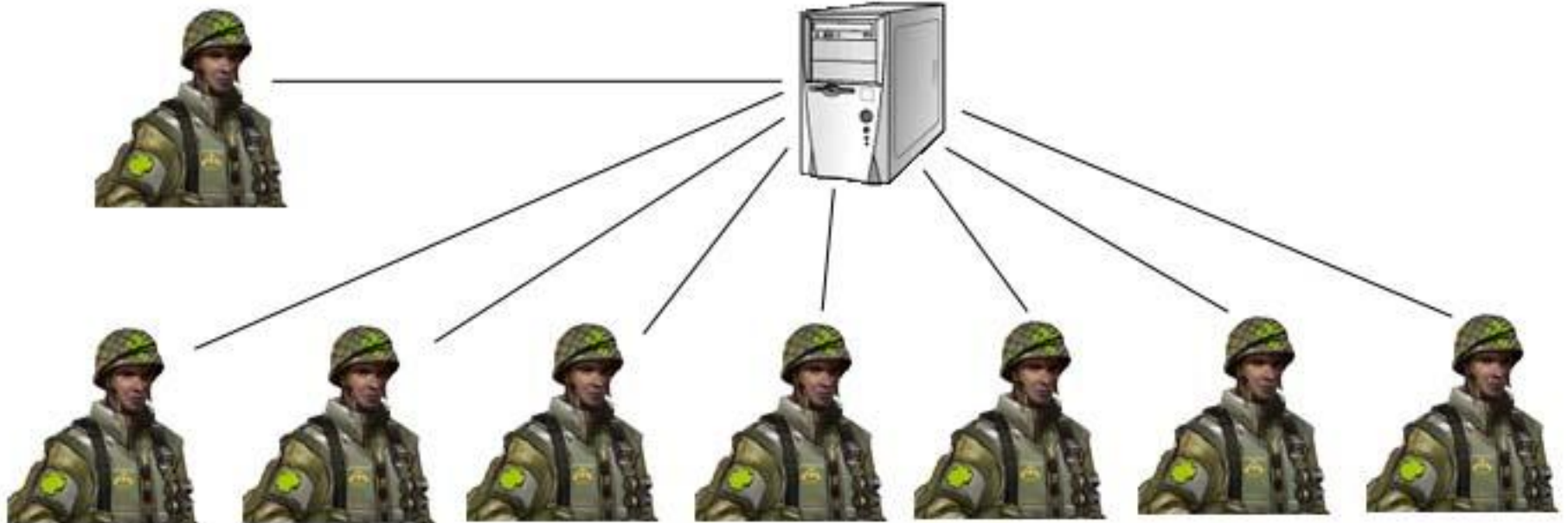
- Which client will be the decision maker for a Bot?
- What decisions or events need to be synced?
- How do we communicate the decisions?

# Authority vs. Non-Authority

Which client will be the decision maker for the Bot?

Authority Client

Server



Non-Authority Clients

# Authority

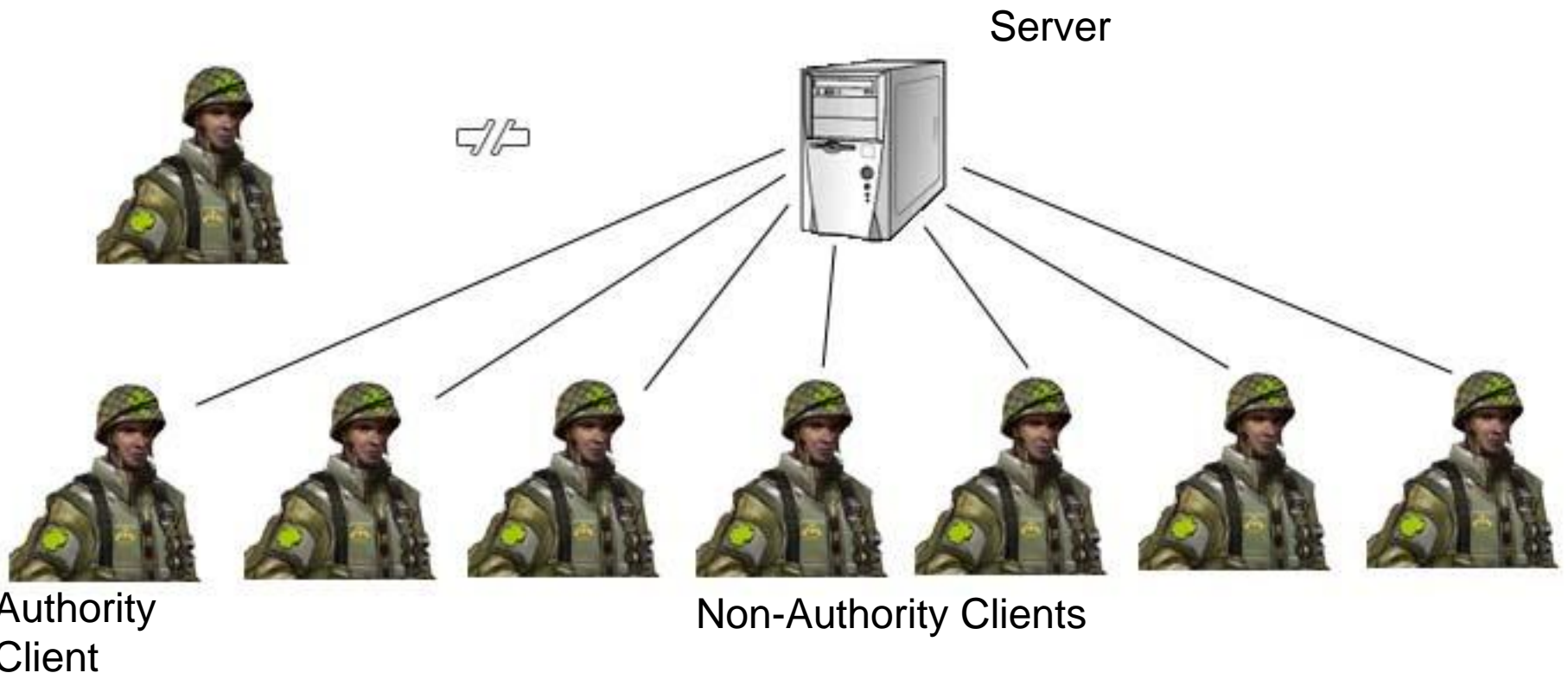
- Runs all AI states just like in a single-player game
- Messages to / from non-authority machines
- All targeting decisions
- All health decisions

# Non-Authority

- Non-authority clients should not be running any decision making code
- Runs 'Remote' AI state
- Receives messages and emulates state
- Sends messages about damage done locally

# Disconnection

- Authority will migrate to another client



# Decisions made by Authority Client

- Idle
- Position / Rotation
- Follow path
- Fire weapon / projectile
- Throw grenade
- Melee attack
- Dodge
- Take cover
- Targeting
- Health

# Decisions made by Authority Client

- Idle Goal
- Position / Rotation Goal
- Follow path
- Fire weapon / projectile
- Throw grenade
- Melee attack Goal
- Dodge Goal
- Take cover Goal
- Targeting
- Health

# Decisions made by Authority Client

- Idle
  - Position / Rotation
  - Follow path
  - Fire weapon / projectile
  - Throw grenade
  - Melee attack
  - Dodge
  - Take cover
  - Targeting
  - Health
- State  
Update  
State  
  
Component  
Component

# Damage

- We only care about what is fair to the player
  - Bots don't get to complain about lag!
- Damage done to a Bot by a player is determined on that player's local machine
- Damage done by a Bot to player is determined on that player's local machine

# Synced stuff we got for free

- Leaper roll attack – based off of movement speed that was already synced as part of the GotoPosition goal



# Synced stuff we had to code

- Titan Roar attack



# Communication



- Finding the corresponding Bot on another client
- MobySync class

# Corresponding Bots

- Placed Bots will have the same moby handle on all clients 😊
- Spawned Bots will not 😞
- SpawnFactoryController – spawns all of the Bots in co-op and assigns them an ID
- If you have the controller handle and an ID, you can find the corresponding Bot on another client

# MobySync class

- Pointer in GameMobyUpdate class
- MobySync objects are only allocated in co-op
- MobySync stores the SpawnFactoryController handle and ID assigned to the Bot, and can use this to find the corresponding Bot on all clients
- MobySync is used to send and receive network messages

# Conclusion

- The Authority client is the decision maker for a Bot
- Important decisions need to be synced (unimportant ones don't matter)
- Use the MobySync class to find corresponding Bots, and to communicate decisions