Networking Guidelines

The How the Fuck will I Network this Guide

# Protocol

TCP

## Why?

… Because I don’t have time to mess with building a thin TCP reliability layer on top of UDP. In addition, Bluetooth within Windows Phone operates under TCP and it will cut down time incorporating both protocols.

## Additional Info

Because Network will be Local, I expect there to be zero or near zero Packet Loss with speed being non-issue.

# Network Scope

Local Area Network with the ability to expand to Bluetooth

# Model

Client/Server model. As the game is being created, players will choose to be either a Host or a Client.

# How to Connect/Initial Handshake

## LAN

### Host

The player which will host selects Host from the LAN connection menu. From there they will be thrown into the Game Lobby

### Client

The player which is the client will connect to the host types in the Hosts IP Address which is shown on their Game Settings Sidebar.

## Bluetooth

# Menu Navigation

## Game/Map Settings

### Host

The host has full control of the Menus. Game Modes and Maps changed will be sent to the Clients

### Client

The Controls for all settings are locked to Clients. When a Host changes the setting, they will be sent to the Client who job it is to change the settings to show to the user.

## Chat

### Host/Client

The Host and Client can both type into the Chat Box. Messages sent will be received by the other and appended to the Message Log

## Game Start

### Host

#### Start

To start the game the Host selects the button titled “Start”. From there the Host will send a message to the Client letting it know to start the countdown.

#### Cancel

If the Host wishes to cancel the Game Start, they simply select the button titled “Cancel”. From there a message will be sent to the client to let it know the countdown has stopped and has been reset

#### Begin

Once the countdown is finished, the Host sends a Begin message to the Client to tell it to transition states into the game.

### Client

#### Start

If the Client receives a Start event from the host, then the countdown will begin.

#### Cancel

If the Client receives a Cancel event from the Host, then the countdown is stopped.

#### Begin

If the Client receives a Begin Event from the host, it will transition its states into the game.

# General Gameplay

## Initial Loading

### Host

When the host is done loading, it will wait for a prompt from all of the clients telling it that the game is ready to start. When all the clients are ready, the Host echoes back a GameLoaded event

### Client

When the client is done loading, it fires to the Host that it has completed loading. It then waits for a corresponding GameLoaded event before starting the game

## Gameplay

### High Level Summary

All Gameplay Actions will be taken through “Commands”. Input is taken from the user and converted into Useful commands which will be sent to the other player to keep the simulation Synced between official Game Syncs.

When the other player receives the commands, they will play them out on their side of the simulation

## Syncing

Every 15 Seconds, the Host will send a dump of the current playing field to sync up all the clients. All commands will embed a SyncID to keep track of which commands are associated with each sync. Commands received outside of the sync will be thrown away.

## Commands

Not all commands capable by the game will be sent over the network, only the commands required to run the game.

### Universal Commands

Network Commands which anyone can issue

* Move
  + Required:
    - Unit ID
    - Calculated Path
    - Position To Move To
  + Optional:
    - Patrol Flag
      * If flagged will have the unit patrol between its current location and its new location
* Cancel
  + Required:
    - Unit ID/Factory ID,
    - Target Type
* BuildFactory
  + Required
    - Factory ID
* BuildUnit
  + Required
    - Unit ID
    - UnitType
    - Factory ID
* SetFactoryRallyPoint
  + Required
    - Factory ID
    - Calculated Path
    - Position To Move To

### Host Only Commands

Commands which are processed and then issued by the Host to ensure a fair and accurate simulation

* Attack
  + Required:
    - Unit ID of attacking Unit
    - Unit ID/Factory ID of unit being attacked
    - Target Type
* Damage
  + Required
    - Unit ID/Factory ID taking Damage
    - Target Type
    - Amount of Damage
* Kill
  + Required
    - Unit ID/Target ID
    - Target Type