# Underwater Sim Checklist

## Unity Project (Sam)

Find a suitable resolution for images to prevent render stuttering (4k is overkill for single target tracking – 256px \* 256px is likely more suitable, N.B: May need to change aspect ratio of camera)

Restructure network/packet classes

Reformat data to JSON (will need to be linked with the suitable controls)

Create a suitable set on controls from which the server can return (x, y, z movement/orientation …probably don’t want all controls – turbines unlikely to allow for such versatility)

Create new movement model of greater likeness to a rover – including ‘hovering’ in the water (a default toggle is probably most appropriate)

Add server tick rate controls

Add Linux build support

Add orientation controls

Normalise/Scale (1- -1) for movement from server

UI Debug information (toggleable too)

Set up message types (including for game config)

Buoyancy, realistic water simulation

Create a more realistic AI that simulates an aquatic animal (3D nav mesh, interpolation etc.)

Endless ocean

## Network (Sam + Kirsten)

### Sam

Determine which data protocol is more suitable (TCP, UDP, RUDP …potentially a mix?)

Decide how often the server sends input data – Only when we want a change (assuming the rover should maintain velocity) OR constantly send controls every frame (per image we tell the rover what direction to head)?

Add disconnecting sockets so the server stops killing itself

Set up message types (wait for new server from Kirsten)

Set up config setting for unity + server

### Kirsten

Integrate current server script to interact with RL algorithm

## Reinforcement Learning (Kristen)