# Realistic Plant Sim

Minor Immersive Media / Project IMT&S

Glyn Leine
Maurijn Besters
Robin Dittrich
Rowan Ramsey

# The Project

# The Team

**Daan de Leeuw**

**Glyn Leine**

CMGT
Engineer

**Maurijn Besters**

HBO-ICT
Software Engineer

**Robin Dittrich**

HBO-ICT
Software Engineer

**Rowan Ramsey**

CMGT
Engineer

# Team Workflow

# The Assignment

- Virtual field of realistic plants
- Virtual robot that can move through the environment
- Connection to ROS
- Interaction
- Modularity

# Stakeholders

## Wilco Bonestroo

Researcher Software for Robotics at Saxion

## Abeje Mersha

Professor of Unmanned Robotic Systems at Saxion University of Applied Sciences

## Saxion Mechatronics

Focuses on the development of robotic and mechatronic technologies

## The ROS Community

Target audience Robotics developers/enthusiasts (I.E. Pixel Farming Robotics)

# Design Process

# The Design Process

**Problem Statement**

What is the problem? What do we need to fix?

**Solution**

How do we fix it? What are the steps to take?

**Prototype**

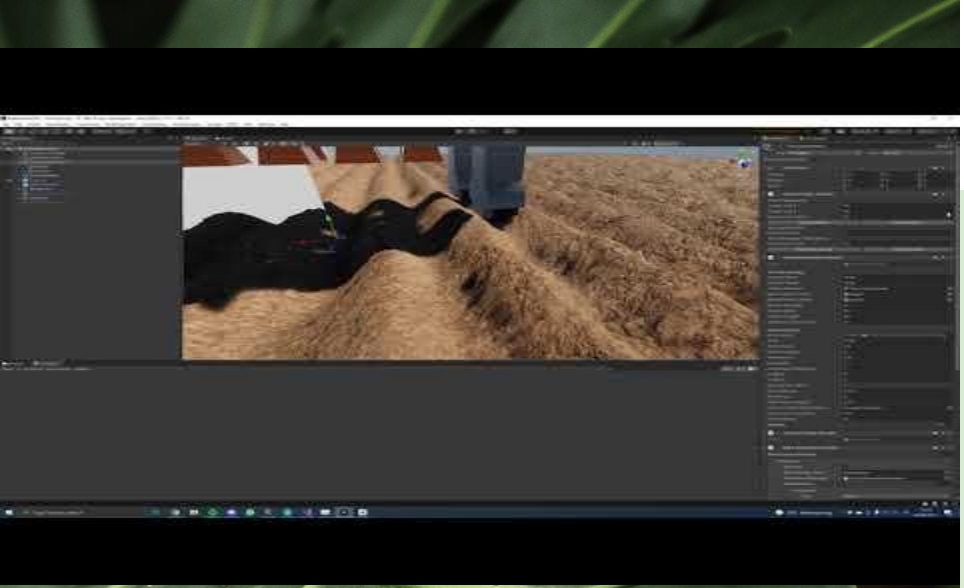
Creating a solution to the problem.

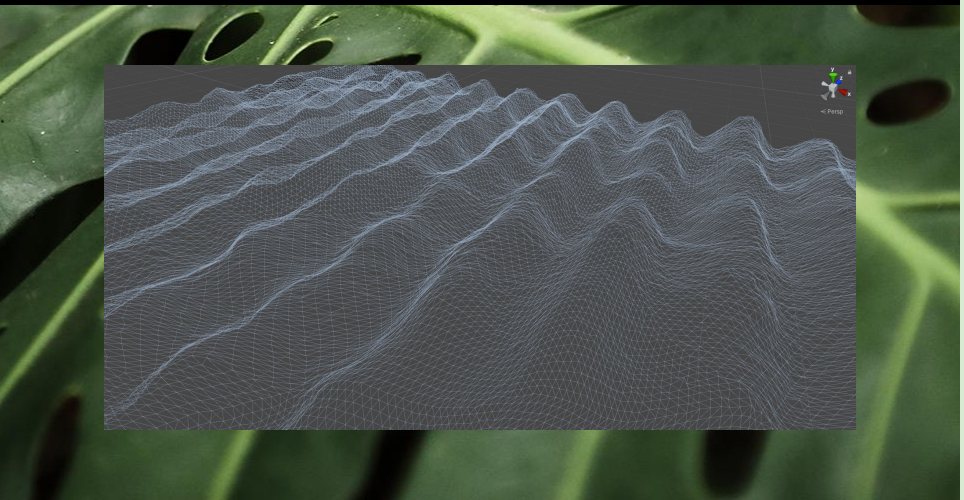
**Test**

Test the solution. Find the issues.

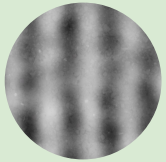
**Reiterate**


Final Showcase



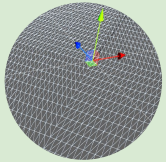


# Terrain generator



## Procedurally Generated Terrain

Is modular to make it possible to generate many types of terrain



## Tessellation

Procedural maps are used to tessellate the terrain

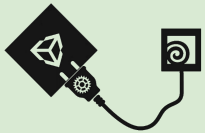
## Compute shaders

Generate mappings on compute shaders instead of the CPU

# Plant Generator

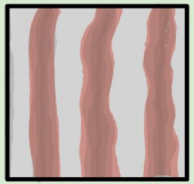
## Houdini

Houdini is used to procedurally generate plants using L-systems



## Houdini-Unity connector

The connector is scripted to change settings of plants to vary the way they look



## Placement Strategies

Placement strategies are used to vary the  spawning behavior of plants

# Camera Streaming

## FFmpegOut (Keijiro)

Video capture plugin for Unity that can record camera streams to a file. Modified for camera streaming to RTSP

## RTSP Server

RtspSimpleServer

Simple intermediate server that broadcasts our stream over a IP

## Custom Compute Shader

Cs

Reads the current frame rendered by the camera and parses the color and depth information into separate textures to be streamed

# ROS-Unity Communication Pipeline

## ROS Noetic

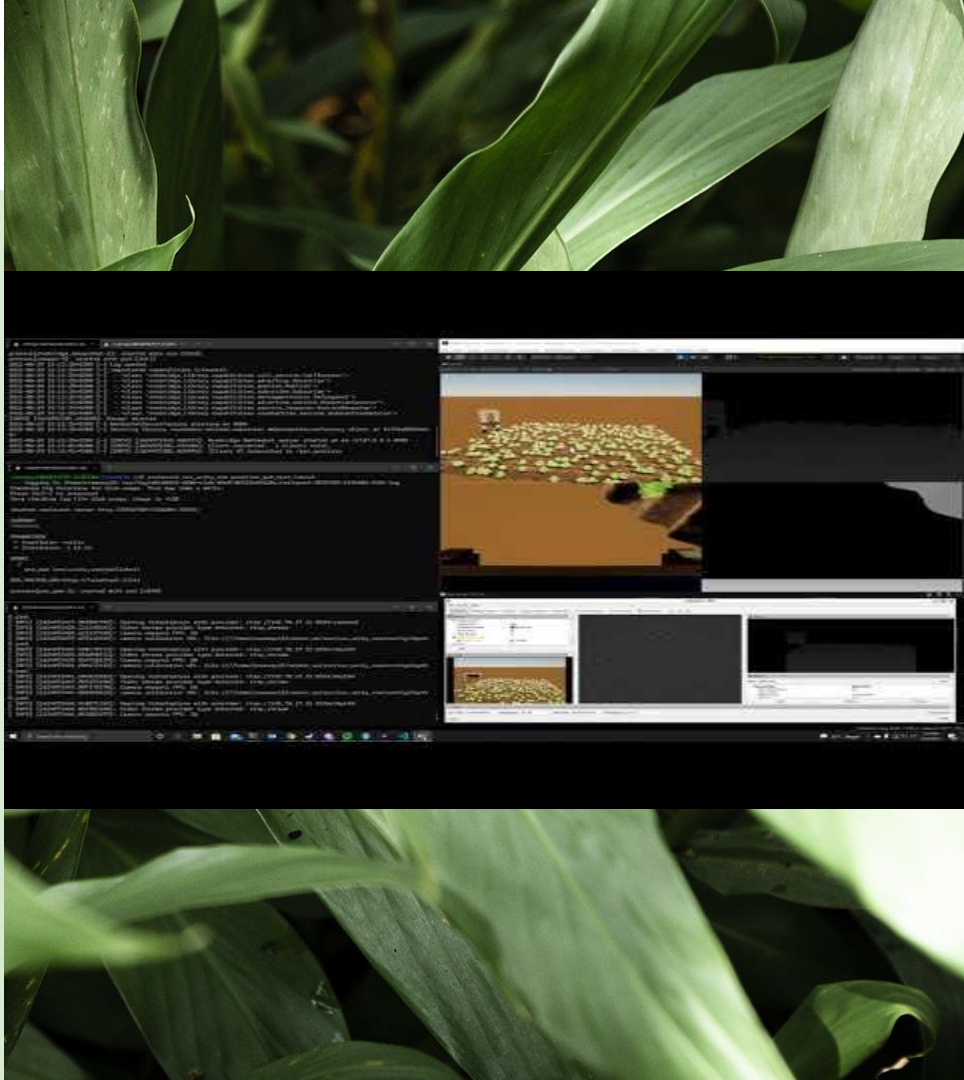
EOL for ROS1, but is still widely supported throughout the community. More stable packages for Noetic than ROS2

## ROS#

Unity Package that provides a C# API that can read and write ROS protocols

## rosbridge_suite

Broadcasts and receives ROS protocols over the WAN allowing anyone listening to send and received ROS protocols outside of the ROS LAN



# Graphics

## Post FX

Physically based effects that simulate various light interactions in the scene and with the camera sensor and camera lens.

## Raytracing

Ray traced alternatives for global illumination, reflections, fog, ambient occlusion, and subsurface scattering.

## Custom Simplified Shaders

Tooling for simplifying the process of creating custom shaders for the Unity HDRP.


Project Reflection
How the project went for us as a team

# Technical problems we faced

- ROS2 support is lacking in the community
- Developing on both Linux and Windows
- XR-Lab Computers
- Unity is lacking specific features (DLSS, Soft Bodies, etc)

# Project Reflection

## Problems we faced as a team

- Initial poor understanding of the project
- We would have liked to work on real robotic hardware
- Our client was unable to assist us due to personal reasons
- We had no artist to work with
- Some team members were consistently not on time

## Realisations we had

- Switching to Jira opened up a lot of possibilities for us
- Working at the XR lab sped up our development
- Defining milestones helped us feel more motivated and gave us a goal to reach for

# Recommendations

# Optimizations

- LOD’s
- Calculate terrain in shader
- Instanced rendering
- DLSS
- Implement Vegetation Engine
- Separation from the Unity Editor

# New Features

- Softbody plants
- Ground deformation
- Path-tracing
- Environmental Effects (dust, rain, etc)


Thank you!