

About the Project

Deliverables

The foundation of the projects API with Farming Simulator.

- XRFS19 Mod
 - Includes a FS CLI interface for adding autopathing points and a basic solver.
 - Includes the GIS-Imported Cook's Farm

-> Link to Mindworks Wiki <-

Tools Required

- Steam Account from Dev
- Farming Simulator 19
- GIANTS Editor
- Lua 5.1 (Recommended)
- QGIS (For map data manipulation and creation)

Getting Started

Farming Simulator Account Access

The creds to the Steam account are:

Username: CyberFarmerDev
S J 9 t h Z A 8 X g Q H X A f 5 5 5

Spaces has been injected into the p a s s w o r d string for readability and to prevent basic Github scrapping.

Change ASAP.

Installing and Activating XRFS19.zip

C:\users\steamuser\My Documents\My Games\FarmingSimulator2019\mods - copy zip to mod path - start a new game on the XRFS19 map - check Mods in FS to make sure XFFS19.zip is activated

Using the Mod

Check the README.md in the XRFS19Mod

Make sure you enable the developer console.

Required for Mods

This is a template for the modDesc.xml that **must** be present at the root of the mode for anything to work. This file will tell Farming Simulator what artifcats inside the mod package need to be loaded. This file lists specializations for including Lua files, and maps.

```

<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<modDesc descVersion="43">
  <author>Authors</author>
  <version>Version</version>

  <title>
    <en>Project Title</en>
  </title>

  <description>
    <en>Description</en>
  </description>

  <iconFilename>icon.png</iconFilename>

  <specializations>
    <specialization name="specialization_name"
                    className="ClassName"
                    filename="file.lua"          />
  </specializations>
</modDesc>

```

The modDesc.xml file and all dependancies should be placed in a single folder, which can be zipped and added to Farming Simulator. Use this Makefile template in order to quickly test code changes.

```
DEST="C:\users\steamuser\My Documents\My Games\FarmingSimulator2019\mods"
```

```

default:
  zip ZIPNAME.zip -x ZIPNAME.zip -r .*
  cp ZIPNAME.zip $(DEST)/

```

The zip command will pick up every file in the mod directory (except for the zip itself).

Be sure to change DEST to the appropriate user path.

To test some Lua code, use this as a starting point:

```
TestScript = {}
```

```

function TestScript:cmdHello()
  print("Hello!");
  return "World!";
end;

```

```

--          FS CMD      HelpText      Func          Table
addConsoleCommand("hi", "View args.", "cmdHello", TestScript);

```

A Lua module requires a table, and functions associated with that table. Once these functions are built, you can use addConsoleCommand to create a callable function from within Farming Simulator.

Note: returning a string will print a string - this is showcased in this example.

Farming Simulator Next Steps TODO

- Path generation and creation needs visual tooling
 - Autopaths need visualization
 - Transform our path points into AutoDrive compatible data
 - Interface for selecting autopathing optimizations for Fuel and Time
 - Remove artifacts from maps
 - Add player vehicles to the map
 - Integrate farm plots
-

For Future Consideration

Build this project in Unity. Grow the team around specific tasks like - Create a VR driveable ‘vehicle’ - Model a VR tractor - Create a pathfinding algorithm for ‘vehicles’ - Work with Precision Ag students to understand the data structure required for computing ‘farms’

Understand that the potential of this project has a lot of merit, but because many future topics cannot be explored because Farming Simulator is a very restrictive environment.

Questions for future developers:

- What is precision agriculture and why is it important?
- What experiences can be built for teaching precision agriculture in XR?
- How do you appropriately model an environment for learning which can be run on cheap equipment?

These are questions that should guide your decisions through the project. Add questions as necessary.

If a mentor says that the ‘first semester shouldn’t include any coding’ do not listen to them. Coding == Research. If we had attempted to integrate our research in the first semester, we would have made much more substantial progress.

- Precision Fertilizer
 - The actual goal of this project is to create simulations to train people on how to use new technologies for agriculture. As such, certain devices need to be implemented into the game because they’re used extensively in this new field. iPad controlled drones with specialized cameras, GPS technology, and sensors would need to be added.
 - * Implementing “GPS” into Farming Simulator has already proven to be a monumental task in terms of a semester’s worth of work.
 - * Drones *can* be implemented into Farming Simulator, but would need to be completely reworked (with a Lua Escape) in order to provide automated controls, and advanced camera feedback (IR at the very least would be incompatible) (Confer: <https://www.youtube.com/watch?v=XVa7ewjSLsE>)
 - * Sensors would need to ‘fake’ any and all data about crops. This could be done, but it would be tricky to make the data usable.
- Soil Erosion

- The idea behind integrating some kind of soil erosion simulation is to try and understand how certain automated tractor paths may hurt the quality of a farm of long periods of time. As such, we would have liked to have an automated tractor path actual ‘detent’ a simulation field so that long term effects of this path could be visualized for students.
 - * Ruts in the mud will hold water, and possibly create channels that displace the topsoil. Confer with my previous semester’s worth of research for more information.