Ant Trails

Engine Architecture & Design

Technical Documentation

1. File Version

This document describes the current state of the ant trails project and its technical features.

Version	Date	Author	Changes
0.1	10.06.2022	Eric Eckstein	Setup document filled with basic structure and topics
0.2	10.06	Richard Schultheis	User Interfaces added
0.3	11.06	Daniel Rajs	AI added
0.4	11.06	Marie-Lena Müller	Animations & Audio added

2. Participants

Name	**E-Mail**	**Responsibilities**
Eric Eckstein	ai21m041@technikum-wien.at	Git, Environment, Particles
Richard Schultheis	ai21m031@technikum-wien.at	User Interfaces
Daniel Rajs	Ai21m042@technikum-wien.at	AI
Marie-Lena Müller	ai21m017@technikum-wien.at	Character creation & animation, Audio

3. Technical requirements

Name	**Requirement**
Unreal Engine	Version 4.26.3
Git Repository	Any git working with Github

4. Scope of the documentation

The documentation does include a description of all parts of the developed ant trails project ll as the basic structure of the project in the unreal engine.

Also, there is an explanation of the time spent on the project. Since no time logging tool was used, the times are rough estimations to the best knowledge.

5. Project structure

5.1. File structure

- Content
    - AntTrails
        - *Core*
          Contains Characters and Game Modes
        - *Maps*
          Contains the Default simulation Map and the MenuLevel Map
        - *MaterialLibrary*
          Contains the Landscape Materials for Dirt and Gras and der Instances
        - *Particle*
          Contains the particles for food, and the pheromones
        - *UI*
          Contains the HUD asset as well as Menu and Pause assets for the HUD
    - AI
        - *Ants*

          Contains the behaviour tree AI logic

          Contains the blueprints for ants, home (hive), food and pheremones
    - Leni
        - *Ant_Character_Animated*
          Contains the fully working animated actor for the AI controlled ants
        - *ant_character_IK_V6_final.fbx*

          First version of the ant character – animation export did not work for that version

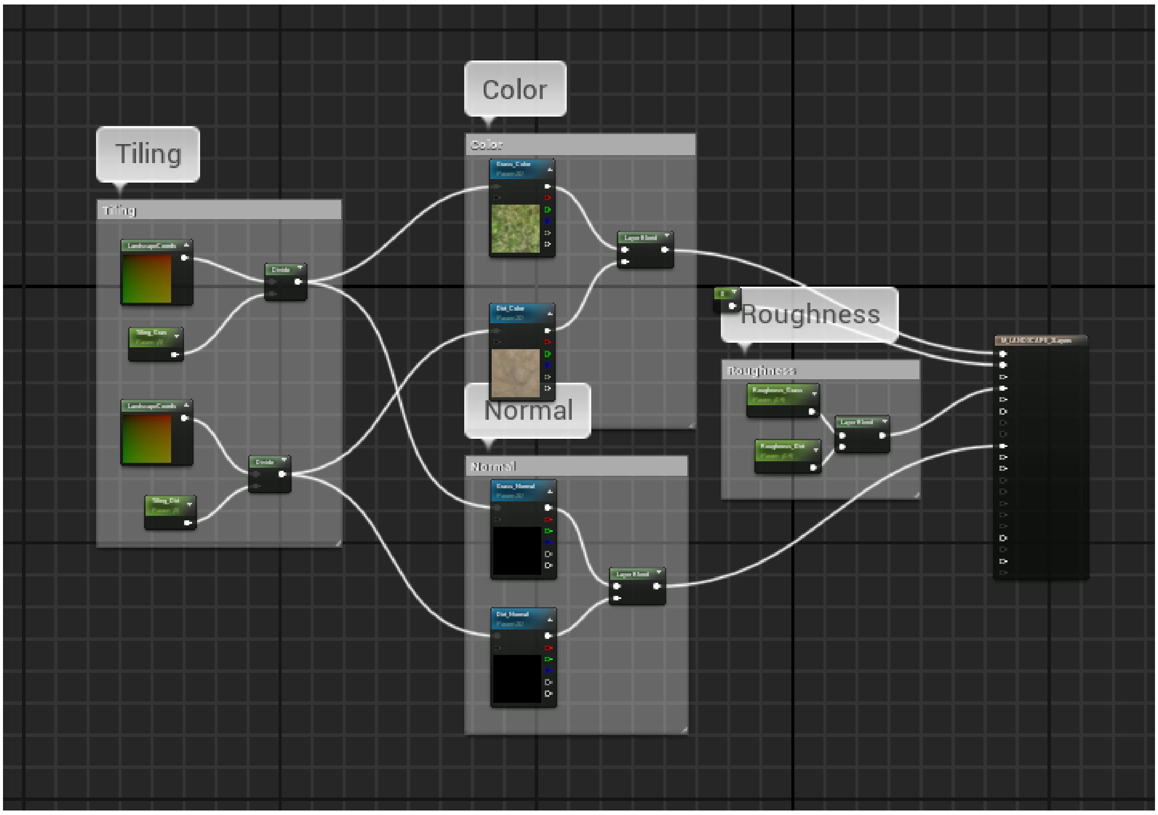
        - *Audio*
          Contains Audio files for all used audio queues
    - Mannequin
      Old mannequin example left over for comparison and testing
    - PolygonFarm
      Assets for landscape are used by foliage tools to enrich the landscape. Contains meshes, materials, and textures
    - StarterContent
      Default starter content. Some materials from here are used for landscape drawing.

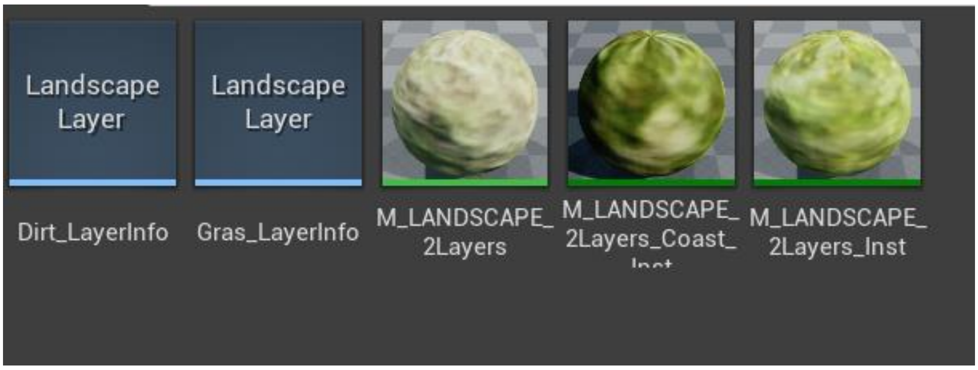
5.2. Materials

5.2.1. For landscape foliage

Two materials are created from scratch. Tiling, Color Normal, and Roughness are included. And reused in Material instances for dirt and grass. Those materials are used for painting the landscape.

project path: Content/AntTrails/MaterialLibrary/Landscape

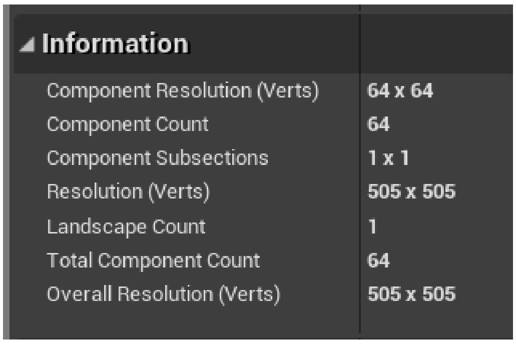




5.3. Landscape

5.3.1. Creation

The ant trails environment is built with the landscape tool of UE4. The landscape is created with the sculpting tool. Random noise and smoothing are used in different strengths and Radius to create a not predefined landscape.



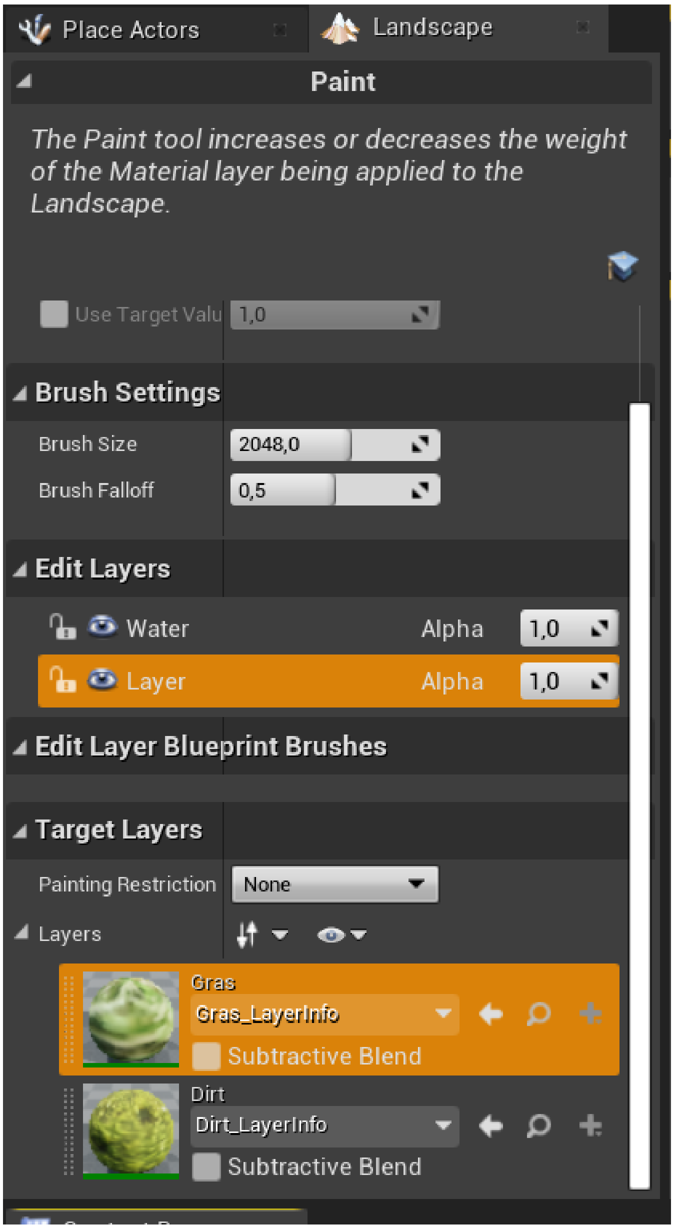
Information

Component Resolution (Verts)	64 x 64
Component Count	64
Component Subsections	1 x 1
Resolution (Verts)	505 x 505
Landscape Count	1
Total Component Count	64
Overall Resolution (Verts)	505 x 505

5.3.2. Material / Painting

The landscape is set to the default grass/landscape material instance described in topic 5.2.1.

The painting tool is used to create better-looking landscapes with two different landscape materials/layer information.



Paint

The Paint tool increases or decreases the weight of the Material layer being applied to the Landscape.

□ Use Target Valu 1,0

Brush Settings

Brush Size 2048,0

Brush Falloff 0,5

Edit Layers

Water Alpha 1,0

Layer Alpha 1,0

Edit Layer Blueprint Brushes

Target Layers

Painting Restriction None

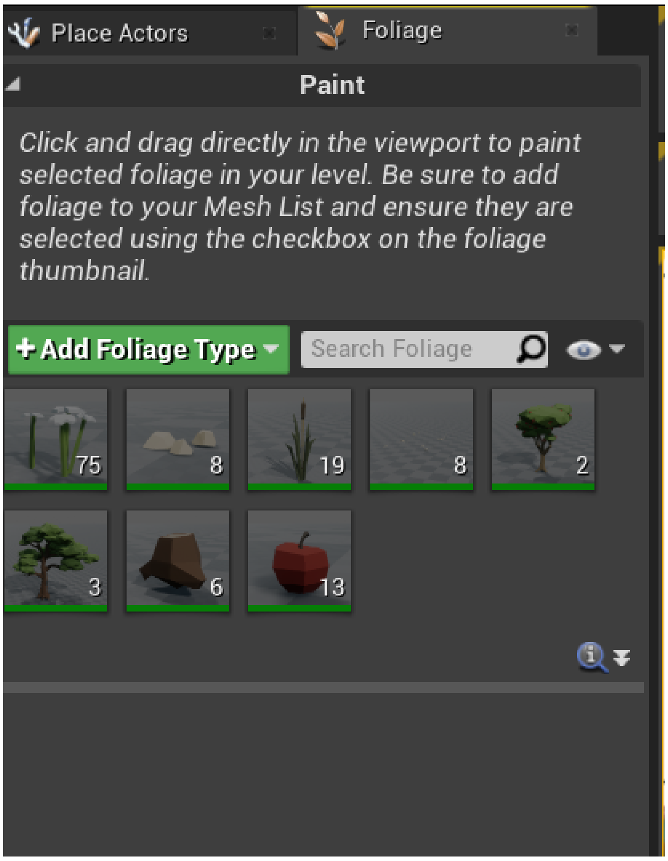
Layers

Gras
Gras_LayerInfo
□ Subtractive Blend

Dirt
Dirt_LayerInfo
□ Subtractive Blend

5.3.3. Foliage

With the foliage tooling from UE4, the landscape got augmented with plants, trees, flowers, stones as well as tree stumps. Also, some fruit (apples) are placed around matching trees. Added foliage is scaled in a factor of „6“ to „12“ to bring them into some more realistic scaling compared to the ants.



Click and drag directly in the viewport to paint selected foliage in your level. Be sure to add foliage to your Mesh List and ensure they are selected using the checkbox on the foliage thumbnail.

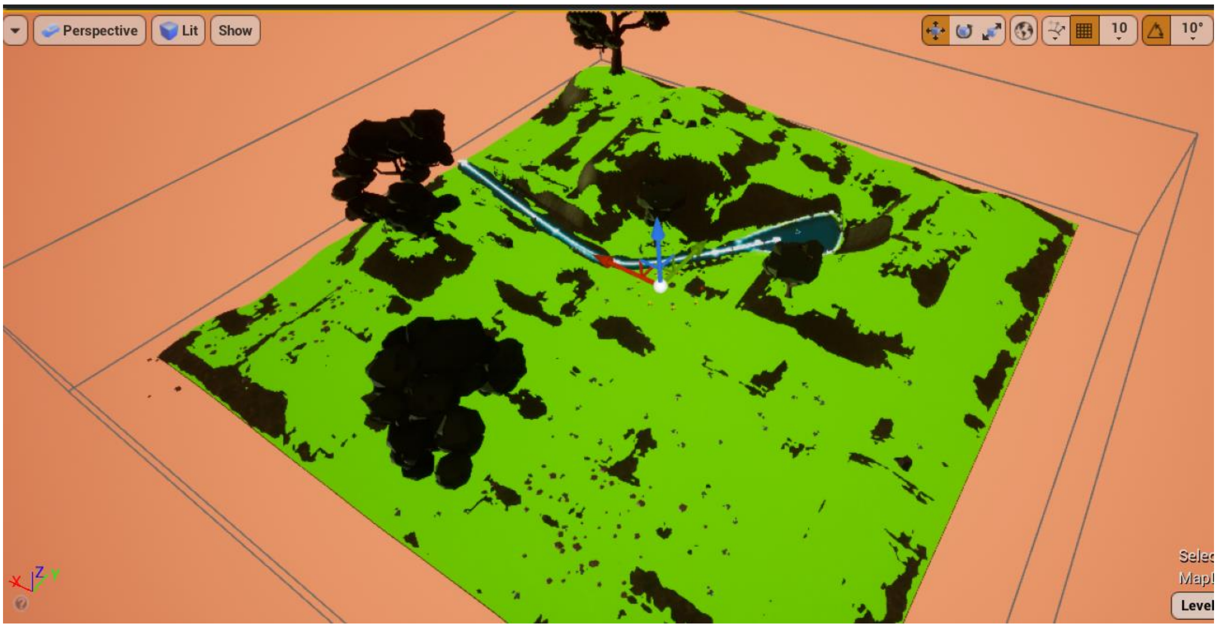
For a performance increase, only a few elements were placed.

5.3.4. River / Lake

The river is created using the waterbodylake and waterbodyriver actors from UE4. Those serve no purpose but are there to bring some depth into the landscape. Also, the river is a good place to bring some depth into the landscape with audio playing matching water sounds.

5.3.5. Nav Mesh

For navigation on the landscape, a navigation mesh is needed. The NavMeshBoundsVolume is used to create the mesh automatically. The mesh is set to not recalculate on every change because of performance. The mesh can be rebuilt by the „Build > Rebuild Paths“ menu option.



5.4. Light setup

5.4.1. Skylight

We use a skylight to match the atmosphere. Since the simulation of ants getting food is in an outside scenario.

5.4.2. Skysphere

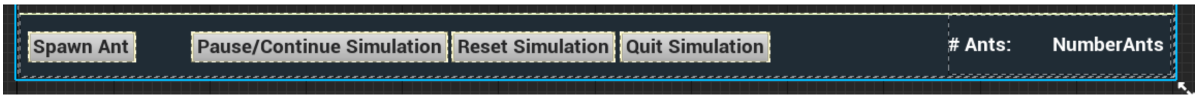
We use a sky sphere to make the horizon have a more realistic look to it. This also matches the outside Forrest realism.

5.5. User Interfaces

Initially a set of user interfaces were planned, but ultimately all but the Heads-Up-Display (HUD) were discarded.

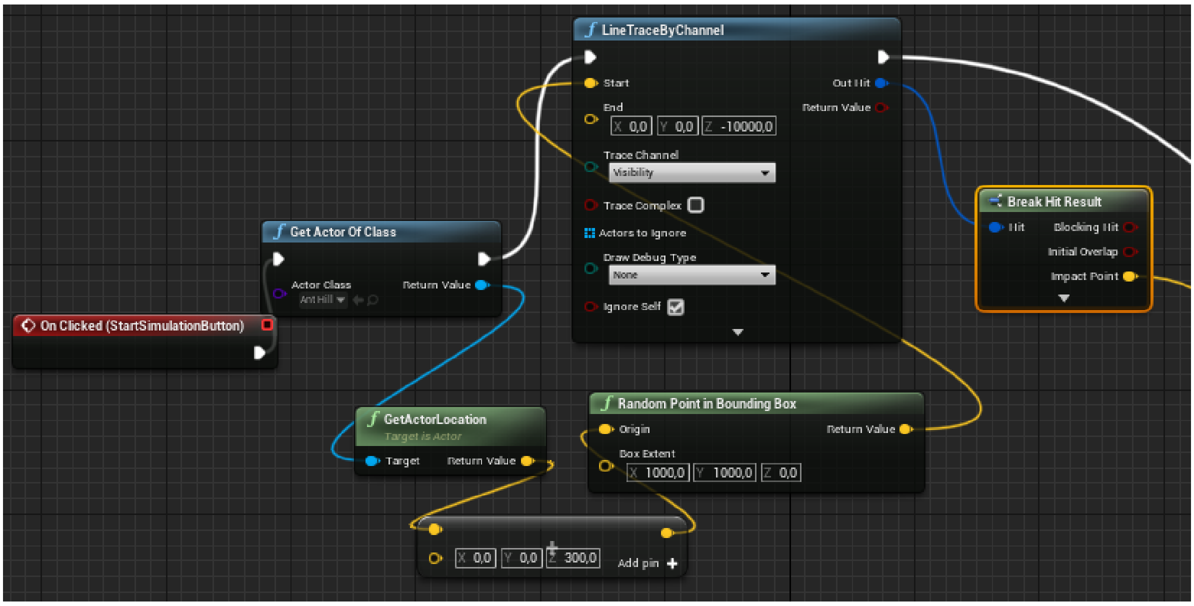
5.5.1. Heads-Up-Display

The HUD is the main interaction tool for the user. It implements functions to Spawn Ants, Pause and Unpause and Exit the Application. The HUD also holds a counter that displays the current number of ants.



Spawn Functionality

The spawn functionality spawns an Ant actor in the surroundings of the Anthill entity on the map.

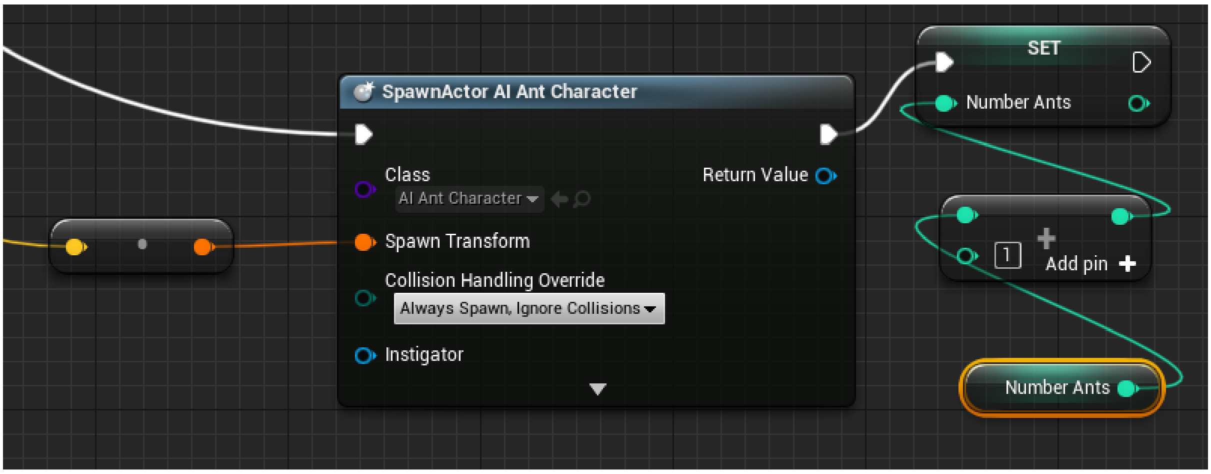


First the actor location of the Anthill is found by “Get Actor Of Class”. This has the intended result since there is only one Anthill entity on the map. Were there multiple, the function “Get All Actors Of Class” would be preferred in order to select the intended Anthill or multiple Anthills for spawning.

From the Anthill location, the position on the map must be determined in all axes. For this a two-part approach is implemented: The XY-location is determined by selecting a random point in a bounding box around the Anthill, modified by a net addition in the Z-axis in order to lift the point above ground.

The Z-Location is then calculated by performing a Line-Trace operation vertically down and finding the intersect point with the map.

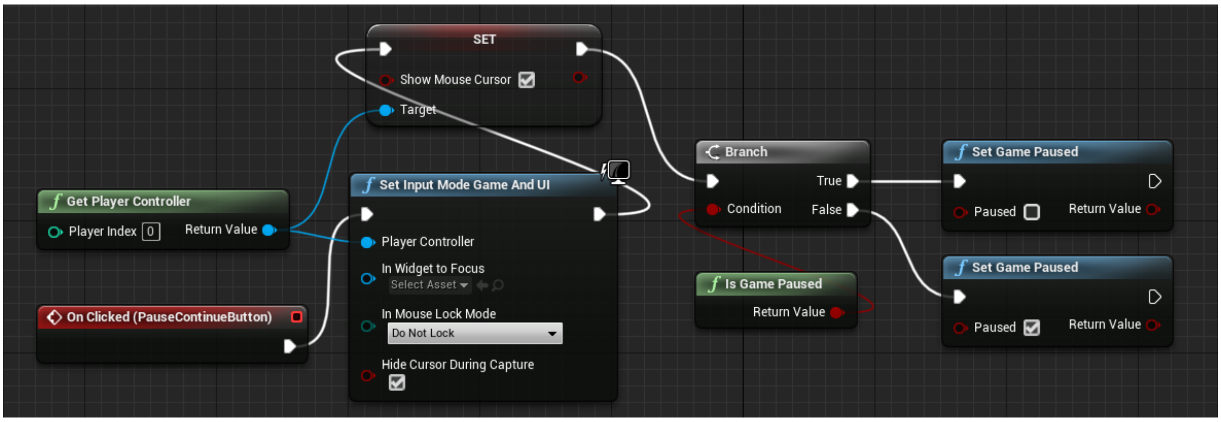
The resulting locations around the Anthill are each used to spawn one Ant actor per activation:



After spawning the Ant actor, a variable" Number Ants" is incremented by 1. This variable is used in the Ant counter functionality.

Pause/Continue Simulation

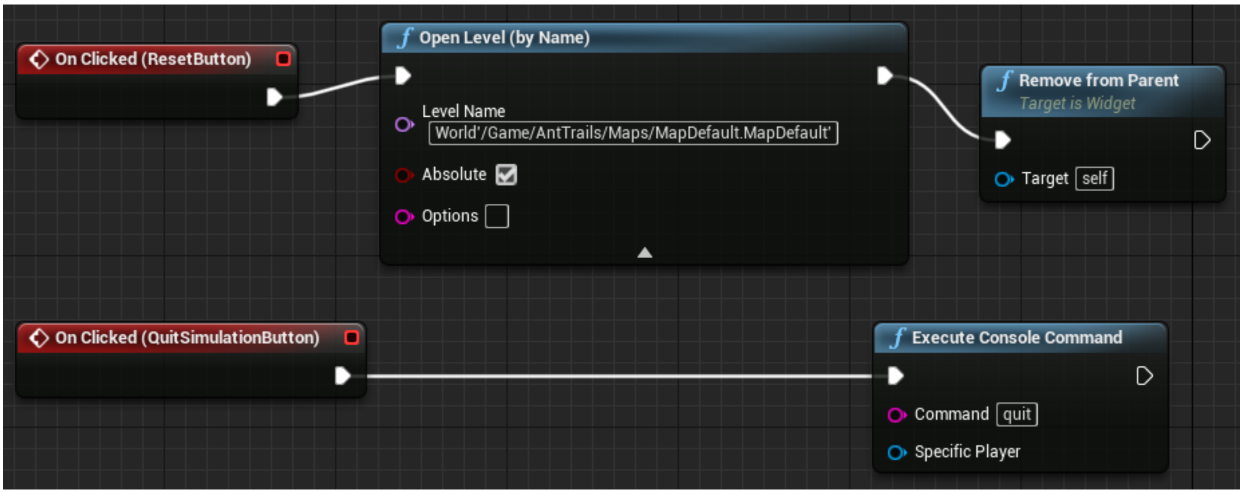
With the Pause/Continue functionality, all action is suspended or resumed, without exiting the application or resetting it.



Upon click, the input mode is set to UI by ”Set Input Mode: Game and UI” only (except in case the UI does not handle the inputs). From a functional standpoint, not suspending the keyboard inputs is a better option.
With a branch operation on the condition ”Is Game Paused”, the action will be paused when not previously paused and will be resumed when the action was previously paused.

Reset Simulation and Quit Simulation

The reset functionality reloads the map “MapDefault.MapDefault” and the Quit Simulation functionality exits the application using the “quit” console command.



Level Name World'/Game/AntTrails/Maps/MapDefault.MapDefault'

Absolute ☑

Options □

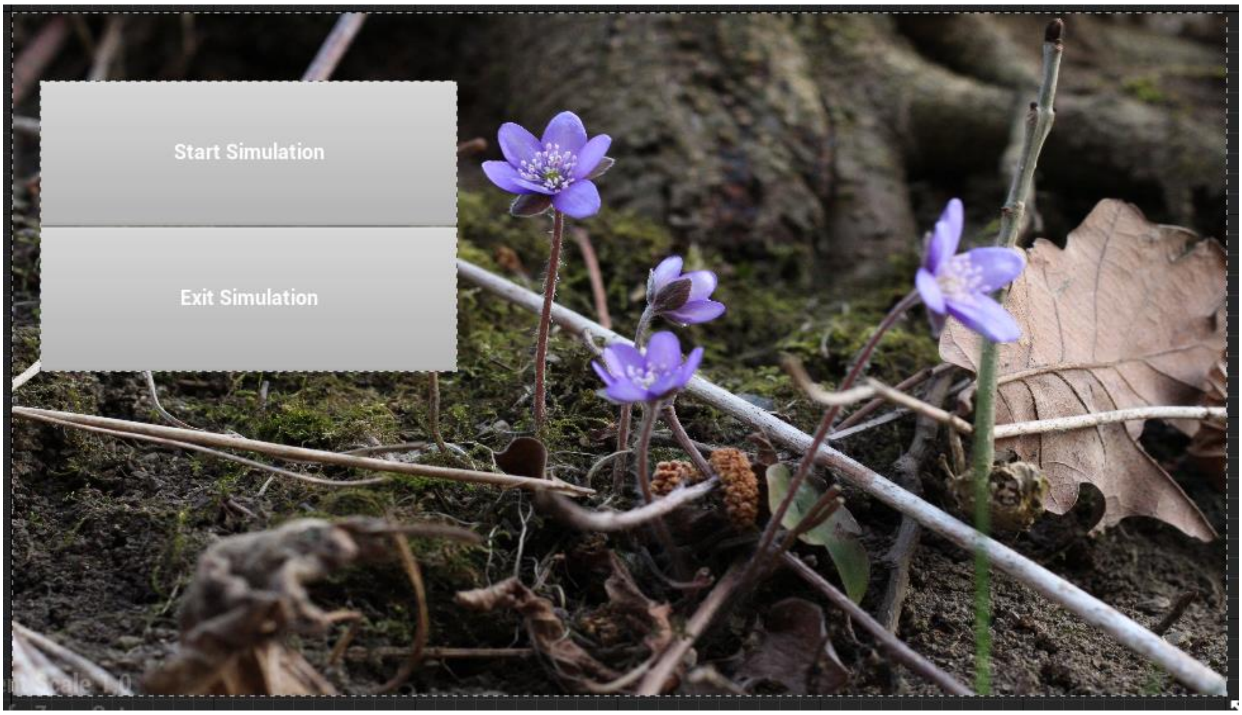
Target self

Command quit

5.5.2. Main Menu

The main menu was initially set up in order to navigate the user into the simulation and out of it. After several unsolved bugs with the water plugin in the Unreal Engine 4 version 4.26, the main menu was abandoned.

The main menu is implemented in a separate level, the “Menulevel”. This way, a consecution of changing the visibilities of menu and HUD elements in a single level is not necessary. In applications with multiple maps, this becomes more necessary.



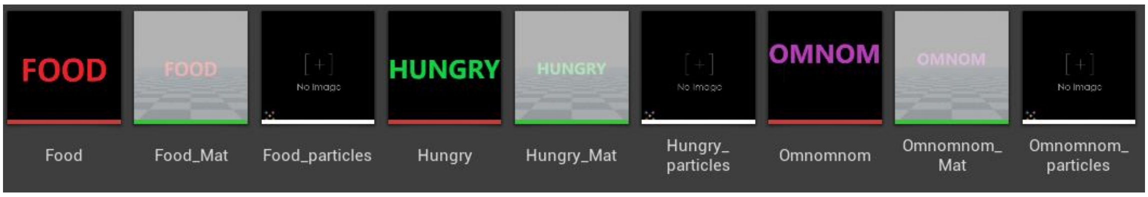
The ”Start Simulation” functionality was planned to be implemented using the ”Load Stream Level” function, but upon launching the simulation, the water bodies were not rendered.

In a cost/benefit analysis we decided to not spend further resources on the main menu, since it does not add significant functionality.

A Pause Menu was initially planned for in order to perform setting adjustments in game but was ultimately discarded also because of the Human Interfaces functionalities being over time budget.

5.6. Particles

Path: Content/AntTrails/Particle



5.6.1. Materials

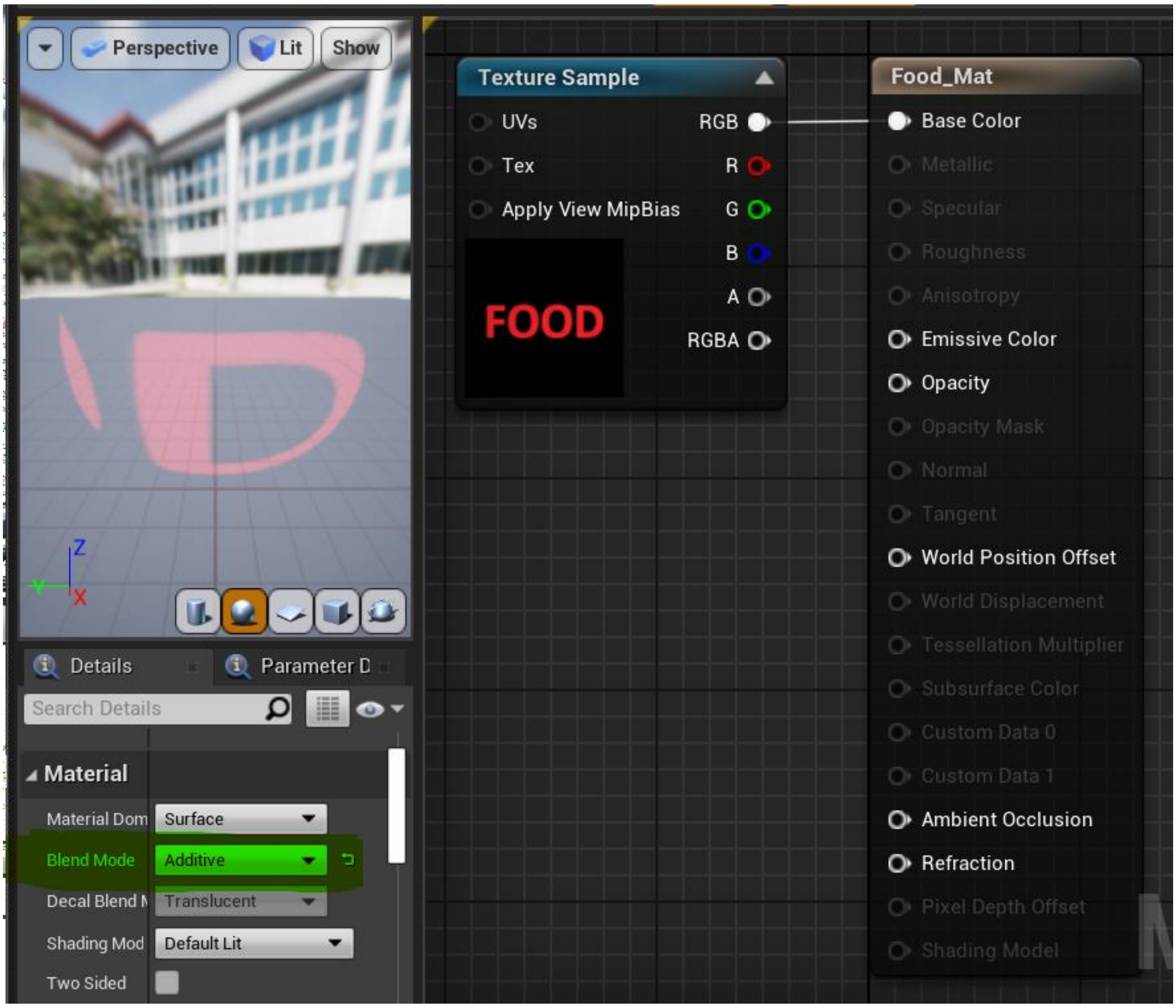
There are three materials created for the custom particle effects. The start point is 256*256 pixel .png images created in paint.net with black background (important!)

Those images are used to create materials. The blend mode has to be set to „additive“. This lead to the following effect: The image is added onto the material – but only pixels that are nonblack are added onto the material (additive adding pixel RGB values). This leads to transparent material-defined color pixels being visible.

From this point three particle systems are created, which use the pre-created materials.

The particle systems are used on the food actors as well as the pheromone actors.

In the pheromone actor, it is decided on runtime which particle system (hungry or omnomnom) is activated since there is only one actor for pheromones and those are distinguished by their boolean „food“ variable.



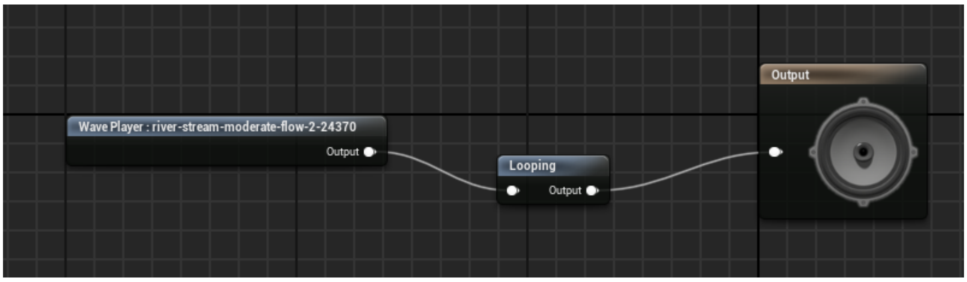
Material

Material Dom	Surface
Blend Mode	Additive
Decal Blend M	Translucent
Shading Mod	Default Lit
Two Sided	□

5.7. Audio

To create an immersive effect for the simulation, background music was added as well as sound effects to some assets. Generally, all sound files had to be converted to .wav files as the Unreal Engine only accepts said format.

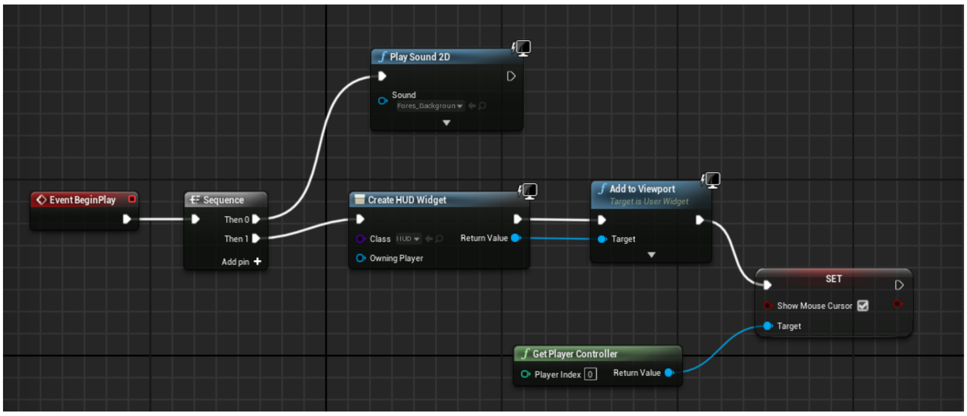
After importing the .wav audio file, a sound cue was created and the sound was looped in the *Sound Cue* blueprint too, to ensure correct looping.



Setup of Sound Cue

5.7.1. Background music

Background music was added via the Level Blue Print by connecting the *Event Begin Play* with the corresponding audio output.

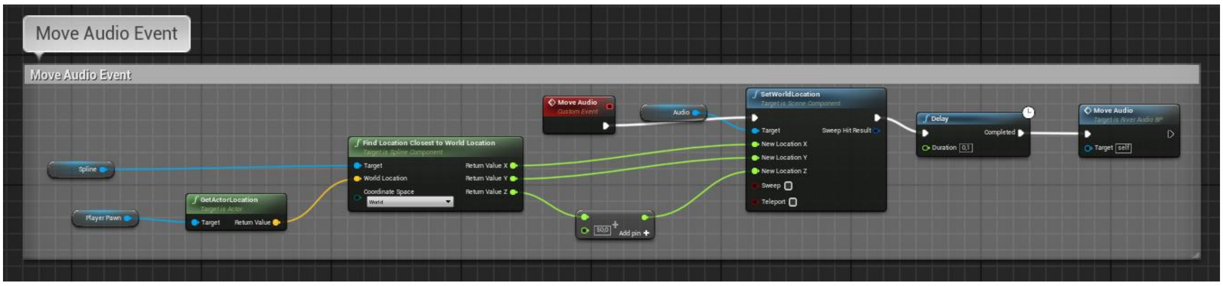


Background music added in level blueprint

5.7.2. Sound for assets

To add sound to assets, the sound cue was directly placed on the corresponding asset. *Sound Attenuation* was switched on to create an immersive 3D effect.

To add sound to the river a spline based audio was used, to ensure that the sound follows the shape of the river and is dependent on the character's location.



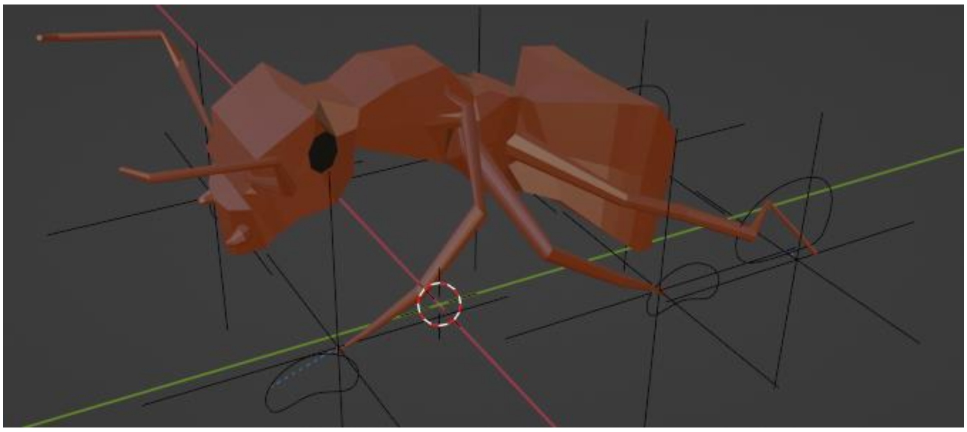
A detail fom the splain based river sound blueprint.

5.8. Animations

5.8.1. Creating an Ant character in Blender

In the first step, a mesh for the ant character was created that was designed based on ant reference images loaded into Blender. Starting from a cube the whole mesh of an ant body was designed and in the final step colored and smoothed by using *Subdivision Surface*. This step was followed by rigging the mesh: In a first attempt, the ant mesh was cut in half and after being rigged, it was mirrored using the mirror modifier – this was done to speed up the rigging process, but proved to be not working properly. Next, inverse kinetic was added to the last bone of each leg and attached to Bezier circles to simulate a walking motion by setting key frames at different locations on the Bezier circle. Unfortunately, exporting the animation in an .fbx file to import into the unreal engine proved to be not possible, as the animation was not exported.

Therefore, the whole process – starting from rigging the mesh had to be repeated and done in a different way. Big attention was payed to the correct parenting and weight painting of the rig – as this might have posed a problem in the previous attempt. The second version of the ant character was done without Bezier circles and simply by setting keyframes of the legs at various positions.



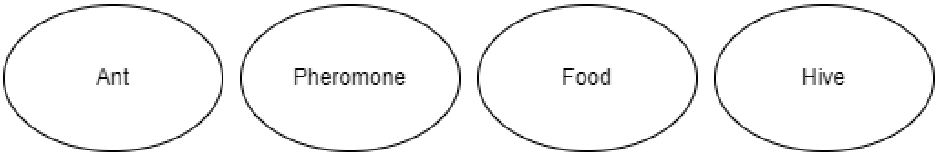
First attempt of creating an animated ant character.

5.9. AI-controlled Ants

The main blueprint is Content/AI/Ant, which is connected to the behaviour tree Content/AI/Ant_BehaviourTree with the Ai-Controller Content/AI/Ant_AiController.

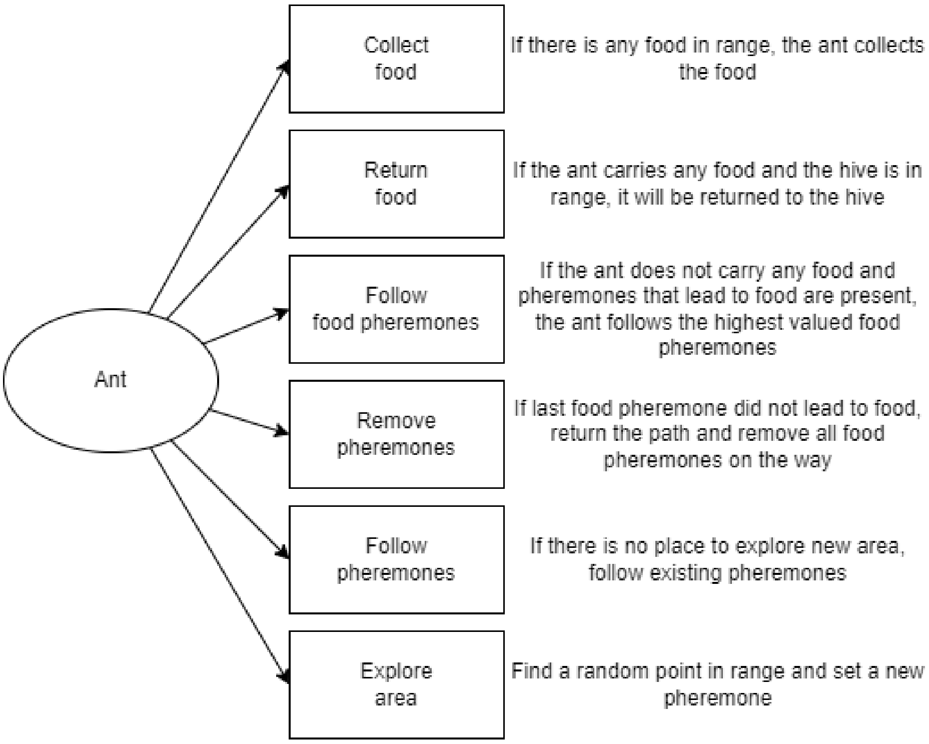
5.9.1. Blueprint placeholders

There are a few blueprints we need to work with. Firstly, the ant, which is controlled by the ai controller and our main protagonist. Secondly, the pheremones which are the main holder of information regarding pathing. Thirdly, food, which is the main target for our ants. Lastly, the hive where ants return to deliver the collected food.



5.9.2. Behavior tree

The behavior tree is composed by 6 main activities (like described below). Each one of those are composed by at least two tasks. The main goal is to find food and return it to the hive, without knowing any final destinations. Communication between ants happens with the help of pheromones.



If there is any food in range, the ant collects the food

If the ant carries any food and the hive is in range, it will be returned to the hive

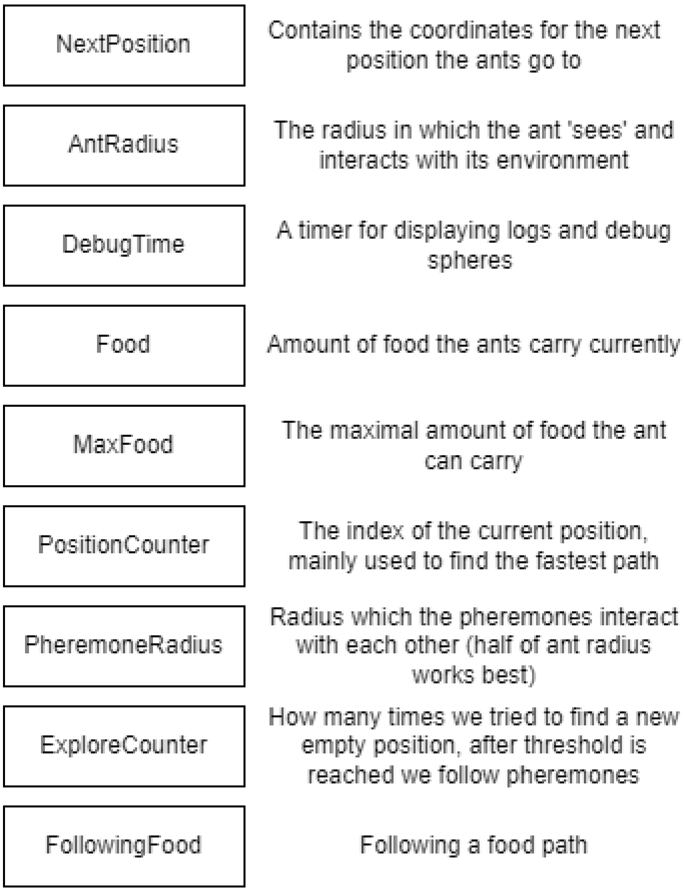
If the ant does not carry any food and pheremones that lead to food are present, the ant follows the highest valued food pheremones

If last food pheremone did not lead to food, return the path and remove all food pheremones on the way

If there is no place to explore new area, follow existing pheremones

Find a random point in range and set a new pheremone

5.9.3. Blackboard



NextPosition	Contains the coordinates for the next position the ants go to
AntRadius	The radius in which the ant 'sees' and interacts with its environment
DebugTime	A timer for displaying logs and debug spheres
Food	Amount of food the ants carry currently
MaxFood	The maximal amount of food the ant can carry
PositionCounter	The index of the current position, mainly used to find the fastest path
PheremoneRadius	Radius which the pheremones interact with each other (half of ant radius works best)
ExploreCounter	How many times we tried to find a new empty position, after threshold is reached we follow pheremones
FollowingFood	Following a food path

6. Timetable

Name	Topic	Time (estimation!)	Period

Eric Eckstein	Git & project setup	4h	06.04.2022 – 08.04.2022
Eric Eckstein	Landscape, Landscape materials, foliage, river, lake,	24h	08.04.2022 – 20.04.2022 (see git repo log)
Eric Eckstein	NavMesh for landscape, (foliage replacement because of asset size)	6h	30.05.2022 – 01.06.2022 (refactoring assets)
Eric Eckstein	Particle systems, material creation	5h	06.07.2022 – 07.07.2022
Eric Eckstein	Technical documentation setup	4h	10.06.2022
Richard Schultheis	Main Menu Tutorials, separate level, resolving problems with water bodies in UE4.26, Bug fixing	18h	13.05.2022-31.05.2022
Richard Schultheis	HUD Tutorials and custom implementation, Spawn, continue/resume Reset, Quit, Ant Counter, Bug fixing	23h	13.05.2022-03.06.2022
Daniel Rajs	Research and prototype ai movement (including behaviour tree, blackboard and tasks learning)	8h	05.-12.05.2022
Daniel Rajs	Set up AI basics (ai controller, behaviour tree, blackboard, blueprint placeholder; reverted due to using wrong engine version)	4h	12.05.2022
Daniel Rajs	Set up AI basics with 4.26	2h	13.05.2022
Daniel Rajs	Add tasks to behaviour tree; Add blueprint placeholder (Food, BreadCrumb = Pheremones)	4h	19.05.2022
Daniel Rajs	Research and replace ai movement with new logic / concept	8h	20.05.-01.06.2022

Daniel Rajs	Add latest ai movement implementation	5h	01.06.2022
Daniel Rajs	Fix and improve ai movement	4h	02.06.2022
Daniel Rajs	Fix and improve pheremone following	4h	04.-06.06.2022
Marie-Lena Müller	Creating ant character (mesh, rigging, animation) including tutorials	26h	20.4.-10.5.2022
Marie-Lena Müller	Trying to fix character, so it can be exporte with animations from Blender	4h	10.5.-17.5.2022
Marie-Lena Müller	Create new ant character in Blender in a different way to create exportable animations	7h	17.5. - 02.06.2022
Marie-Lena Müller	Search sounds, add background music and sound effects; tutorial about audio effects in Unreal Engine	6h	23.05. - 03.06.2022
Everybody	Meetings while project on different occasions	~12h (4 people, 3 meetings around 1h)	