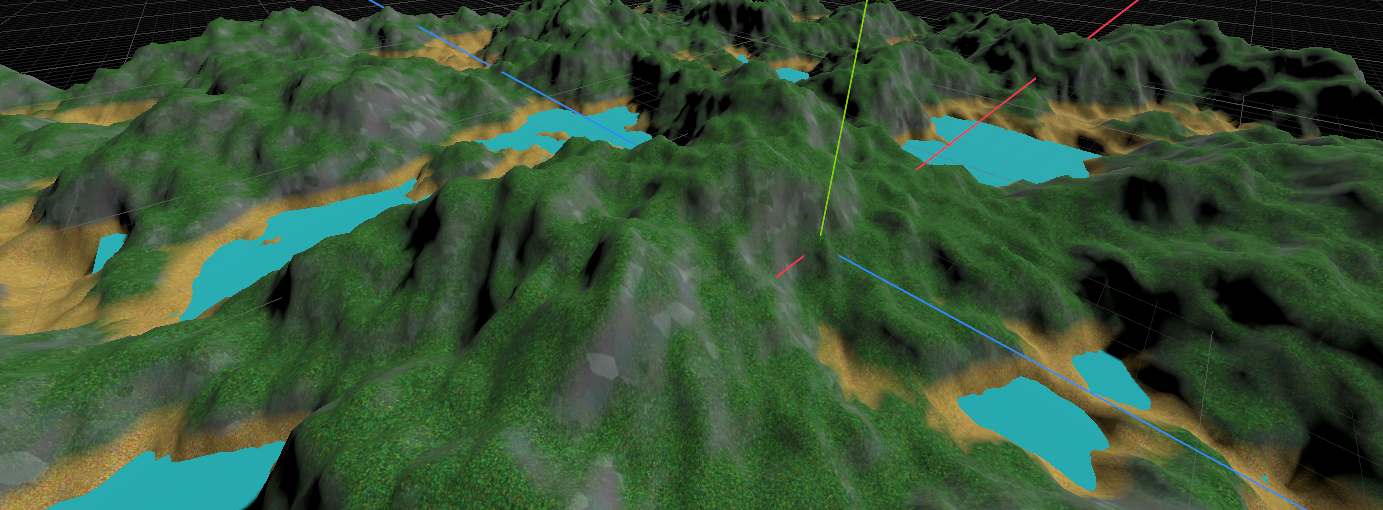
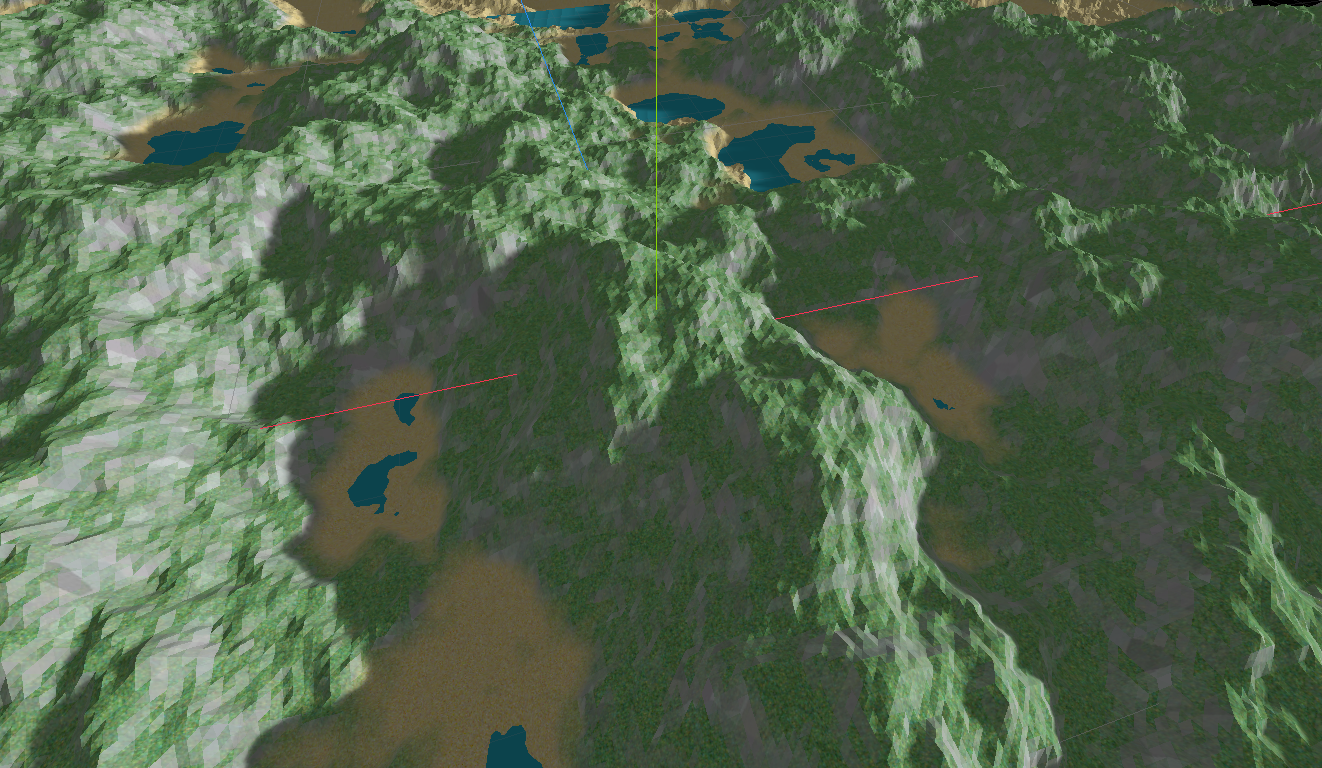
**Terraingenerator**

Gruppe 8:

Julian Matheis, Julius Schilling, Alexander Bornhauser





Projektziel

Das Ziel des Projektes war es einen Terraingenerator in Godot zu erstellen. Am Ende soll ein 3D-Modell einer Landschaft dargestellt werden welches anhand verschiedenster Faktoren verändert werden kann. Dafür verwenden wir das Simplex-Noise was auch in Minecraft zum Einsatz kommt.

Schwierigkeiten und Abweichungen

Da es zu Godot 4 noch nicht besonders viel Dokumentation sowie Tutorials im Bezug auf Terrain Generation gab, haben wir uns zunächst dazu entschieden Godot 3 zu nutzen, da es einige syntaktische Unterschiede und Probleme gab die eine direkte Übertragung auf Godot 4 nur schwer möglich machten. Am Ende hatten wir es allerdings dank einiger Anpassungen und Änderungen geschafft das Projekt nach Godot 4 zu exportieren.   
(siehe Abb. Godot 3, Godot 4)

Nach dem das Projekt vollendet war, schien es uns allerdings noch nicht ganz vollständig und auch nicht interessant genug zu sein. Daher haben wir uns dazu entschlossen einen spielbaren Player in die Szene einzubauen, um das Terrain aus nächster Nähe betrachten und erkunden zu können. Besonders passend fanden wir dies da unsere ursprüngliche Inspiration aus Games wie Minecraft kam.

Unsere „nice to have Ziele“ wie z.B. Bäume haben wir nicht geschafft aber dafür haben wir Wasser eingefügt, wenn auch nur in vereinfachter Form.

Abbildung: Godot 3

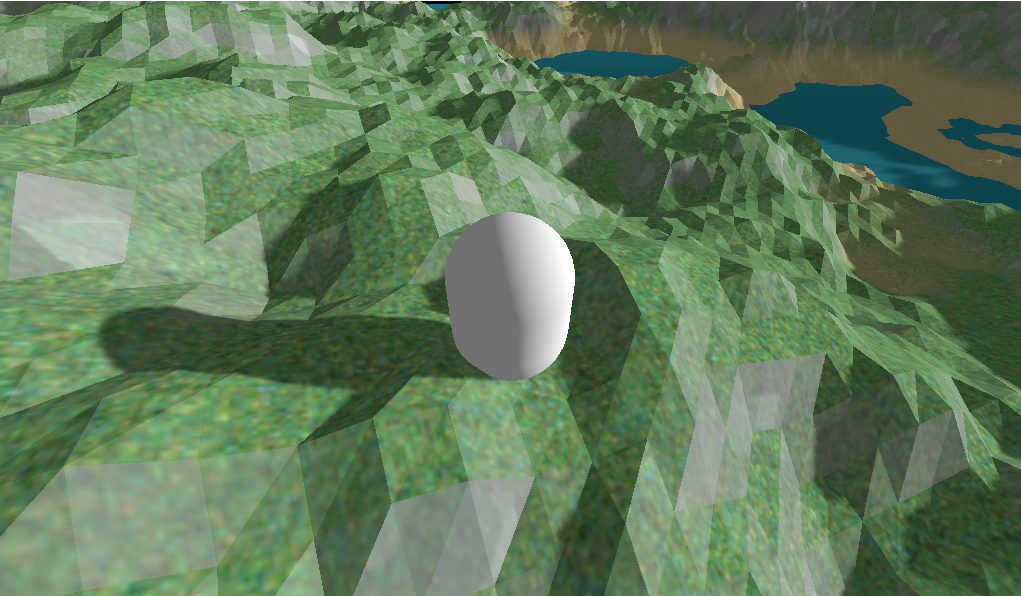
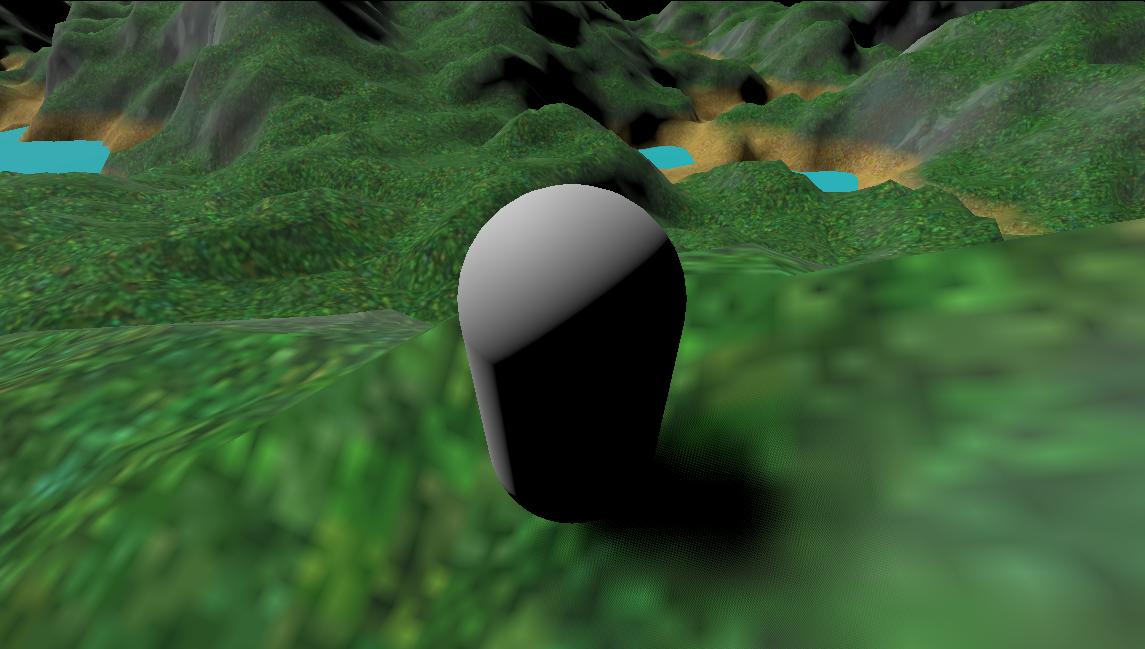


Abbildung: Godot 4

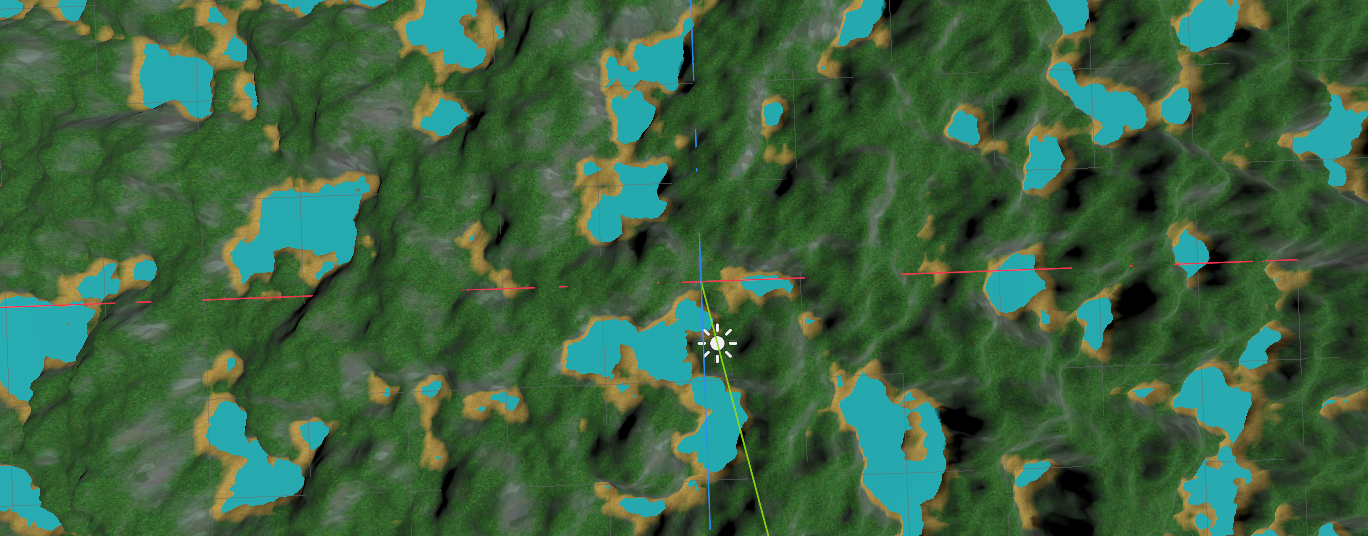


Projektaufteilung

Projekt Leader – Julius hat den Plan gemacht, sich überlegt wie wir vorgehen sollten und Informationen zum Projekt gesammelt. Natürlich hat er auch beim Programmieren geholfen und recherchiert.

Coding - Julian und Alexander waren die Main-Coder. Sie haben die Idee in Code umgesetzt wofür einiges an Recherche und Einarbeitung in bereits vorhandene Vorgehensweisen nötig war. Sie haben viel zusammen an einem Gerät gearbeitet da es so um einiges einfacher war auf einem Stand zu bleiben. Aber auch gemeinsam online.

Dokumentation und Präsentation wurden von Julius Schilling und Alexander Bornhauser Fertiggestellt, da Julian die meiste Code Arbeit übernahm.



Technische Umsetzung

Zuerst wird ein Mesh generiert, welches die grundlegende Struktur des Terrains darstellt.  
Dieses PlaneMesh wird entsprechend den angegebenen Faktoren wie Größe und Unterteilung angepasst.

Mit dem SurfaceTool wird anhand des Noises die Höhenvariation mit y-Koordinaten im Terrain generiert.   
Wir verwenden hier FastNoiseLite, was über eine SimplexNoise Variante verfügt.

Das neu generierte Mesh wird dem MeshInstance3D-Objekt zugewiesen, wodurch es im Spiel sichtbar wird und mit Normalen die Beleuchtung korrekt darstellt.

Ebenfalls wird eine CollisionShape erstellt um physikalische Kollisionen mit dem Player zu ermöglichen.

Das Wasser ist eine einfache blaue Plane, welche am tiefsten Punkt des Meshes liegt.

Der Shader um die Landschaft zu texturieren unterscheidet zwischen drei Texturen:  
Gras, Felsen und Sand. Die Sandtextur wird in niedrigeren Höhen (dort wo die Wasserebene liegt) verwendet, während Gras in den Bereichen mit geringer Steigung verwendet wird. Die Felsentextur wir an den steileren Bereichen angebracht.