RogueGen

A PCG ADVENTURE GAME PROPOSAL

Andreas Leitner
Florian Buschek
Jean Paul Vieira Filho
Johannes Runkel

GamesLab WS 17/18

Game Description

RogueGen is a roguelike game with procedurally generated world map and puzzles. It features 4 player local coop, where all the players must work together to overcome obstacles. Each time a game starts, a new world is generated with unique puzzles.

Set in a fantasy medieval world, the players start at a small town and embark on a quest to defeat a great evil. During their journey, they will face enemies and solve complicated puzzles. The only feasible way to succeed is to work together!

WORLD GENERATION

The generated world features a variety of biomes and landscapes that should motivate the players to continue their journey to discover new areas and at the same time affect the "mood" of the players. If the players are currently on a flowery field with natural fountains, this indicates that it is an area of rest and recovery, without enemy activity. But if the group finds itself in a dark forest with skeletons laying around, they should be ready for a fight.

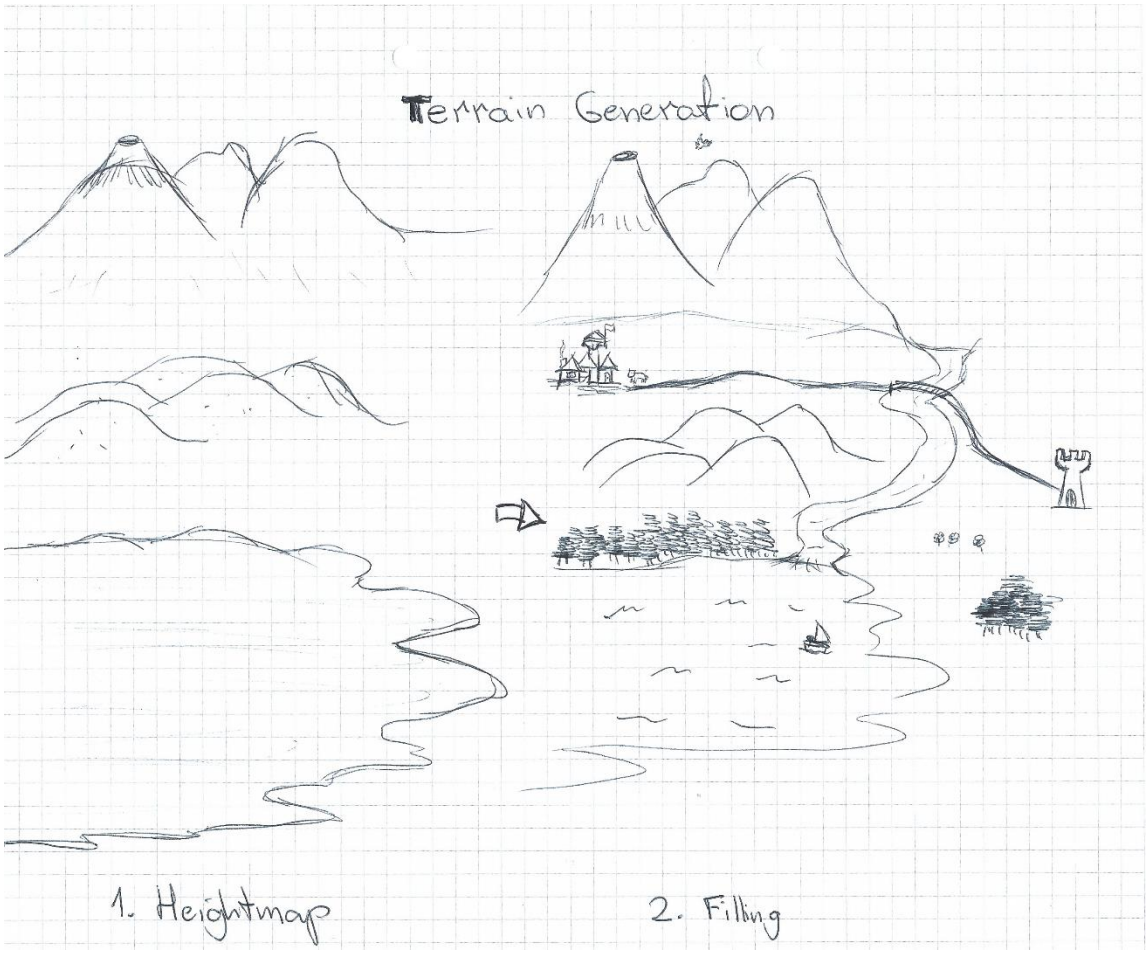


Figure 1 – World (terrain) Generation

Each biome features its own inhabitants and scenery elements, like trees, bushes and rocks. Some special buildings give the players effects, e.g. healing and increased damage, while others are there just for aesthetic purposes.

PUZZLE DESIGN

Distributed through the world, there are multiple puzzles that need to be completed to progress in the game. Currently there is only one type of puzzle planned, but if there is enough time we plan on including more.

Line drawing puzzle: players must navigate an area of the world and find statues pointing in the direction of other statues. At the exit of this puzzle, there is a map with all the statues. The players must draw a line connecting each statue to the ones they are pointing to. If drawn correctly, the door opens, and the group can proceed to the next area.

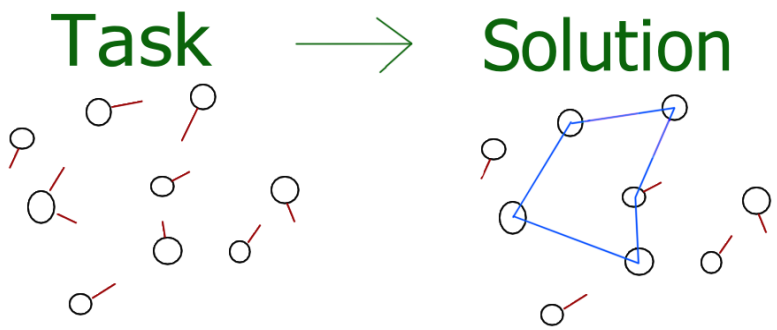


Figure 2: - Puzzle Concept

COMBAT SYSTEM

During their travels the players regularly encounter enemies they have to defeat in combat. This happens in real time and is based on the activation of skills. A skill represents a specific action the character can do in relation to a weapon they have equipped, such as swinging a sword, shooting an arrow with a bow or casting a magical spell. New weapons, and thus new skills, can be acquired throughout the game.

Overall, the combat system is supposed to encourage teamwork through synergies between skills as well as defined strengths and weaknesses of player characters and enemies.

While the skills, weapons and enemies are mostly hand designed and not randomly generated, their acquisition and encounter in the game can be, leading to a variety of different combat scenarios each playthrough.

The players can activate their skills by pressing the corresponding button on their controllers. For enemies, an A.I. system determines their combat behavior.

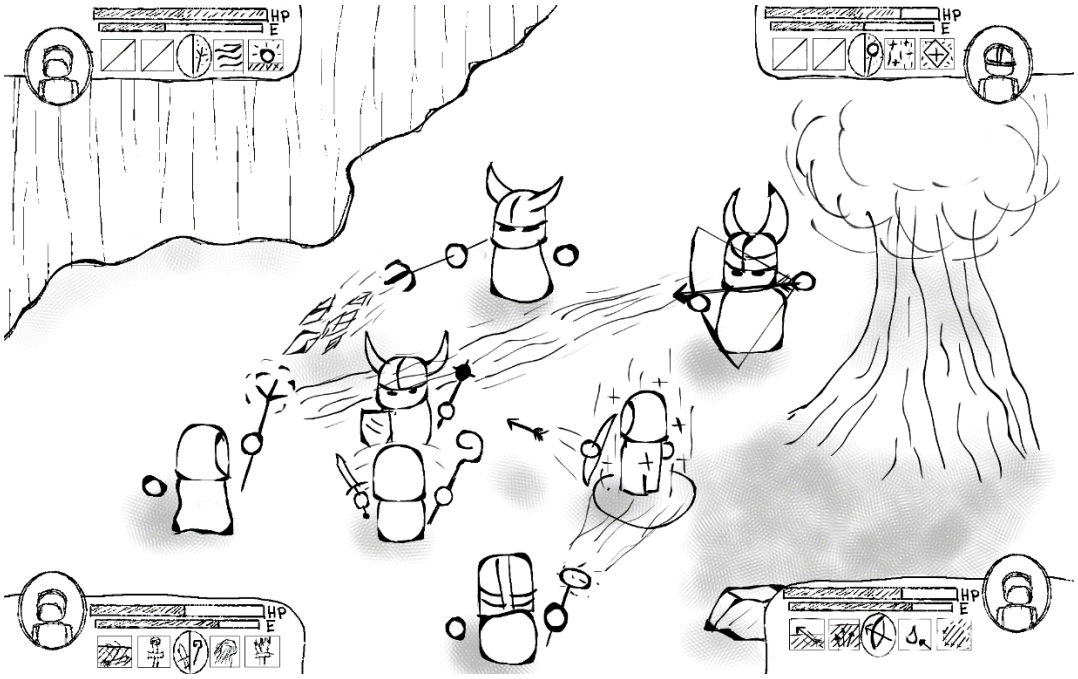


Figure 2 - Combat

CHARACTER DESIGN

To control the character a twin stick system is used. The left joystick controls the direction the character moves, and the right controls the rotation he faces. To ensure fitting interaction with other objects a physics based approach was chosen. This means that the character and the enemies are only moved by adding forces to them. The velocity and position are not changed so the physics engine can ensure a realistic and accurate collision response.

Figure 3 - Character Concept

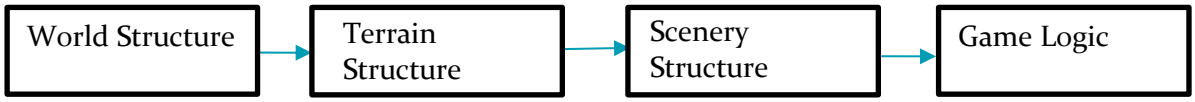
Technical Achievement

The main challenge in our project is the procedural generation of the game world and puzzles. Bringing knowledge from the lectures "3D Character Modelling and Animation", "Techniques in Artificial Intelligence" and "Game Physics", we aim to create an algorithm to generate worlds that are unique and interesting to play on. To increase gameplay diversity, we also want to generate the puzzles randomly at the start of each game.

Our strategy to generate these worlds is to use a combination of graphs, Voronoi diagrams and Delaunay triangulation.

GAME TECHNICAL DESCRIPTION

The technical aspect of the game can be organized in four main steps, each one feeding the next one information for the next step's generation algorithm.



World Structure:

In this part, the world "skeleton" is generated as a graph. It contains the abstract path, i.e. not physically bound to the world yet, that the players can take. To avoid a completely linear path, a planar graph is constructed by starting with "start" and "finish" connect nodes that represent the game's starting point and end. Afterwards, edges and nodes are added recursively to form loops while still maintaining planarity.

Nodes denote Points of Interest in the game. These are enemy spawn points, puzzle element locations, loot, player effect buildings and landmarks.

Edges in the graph represent whether players can navigate from one node to the other. This information is used in the later generation steps

Terrain Structure:

The terrain heightmap is generate at this step. Using a random point distribution function to place the biomes' pivotal points, a Voronoi diagram to represent biomes' area of influence and Delaunay triangulation to mark connectivity between biomes, the heightmap is created and textured. Additionally, large water bodies such as lakes and seas are generated.

Afterwards, the World Structure node positions are assigned to the terrain.

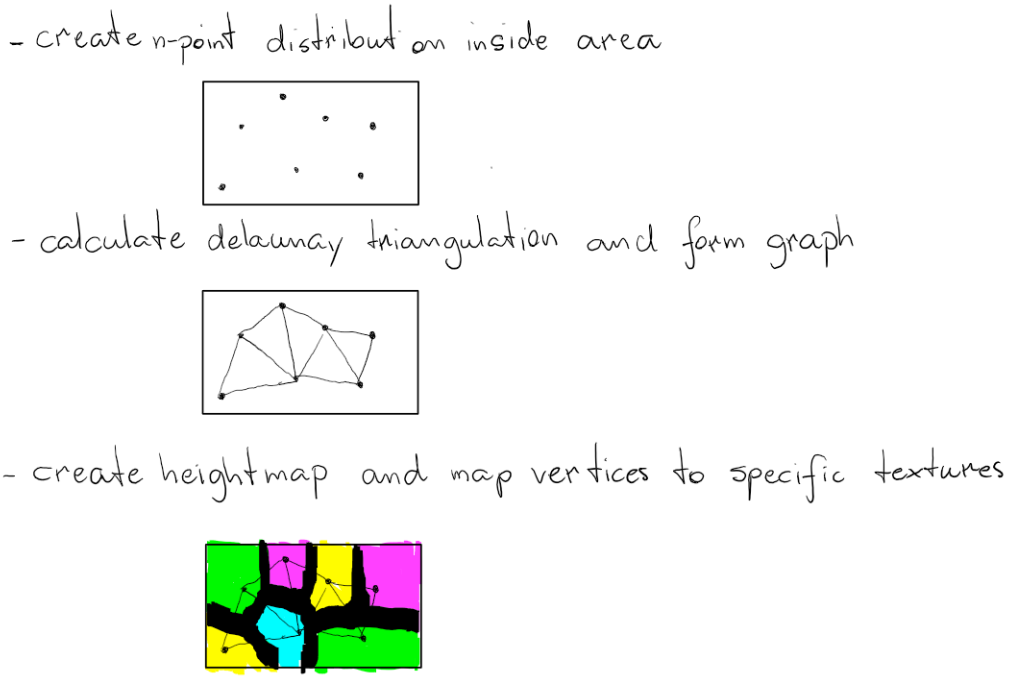


Figure 4 - Terrain formation

Scenery Structure:

Working with both the World and Terrain Structure, this stage fills the world with detail. It changes the previously generated terrain to guarantee the navigability stated in the World stage and adds assets like trees and rocks to the map based on the Terrain Structure biomes.

Furthermore, rivers and caves are generated in this step since modifications to the mesh are needed for these structures.

Game Logic Structure:

The players explore the world generated in the previous stages, searching for the Points of Interest, trying to solve the puzzles and fight enemies on the way.

Movement of characters is entirely physics based, allowing interesting interactions with the environment and the combat, for example knocking characters away by explosions.

Both player and enemy characters share a common core of attributes, such as health points, resistances, and equipped weapons and skills. Making no differentiation between players and enemies at this level minimalizes the need for exceptions in the combat system.

As combat is mostly defined by the skills, a highly modular approach to quickly create new complex skills is chosen. For this, skills have a type component determining the form the skill takes on, such as shooting a projectile, and a list of effects. These can affect

characters directly, such as inflicting damage, or contain another list of effects that they affect, such as activating further effects only if a certain condition is met.

While player characters are controlled directly by the players, enemies require an A.I. system determining their behavior. For this a utility based approach is chosen, where every possible action of a character, such as walking or activating a skill, is evaluated based on a list of considerations and the best option is chosen.

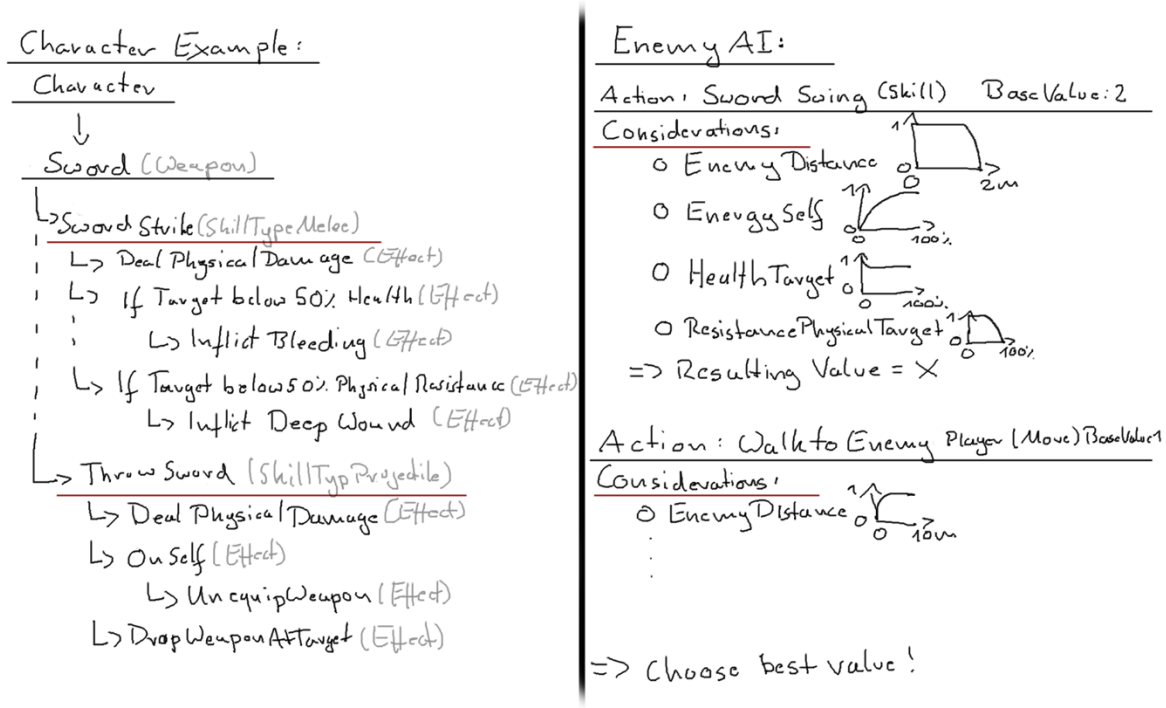
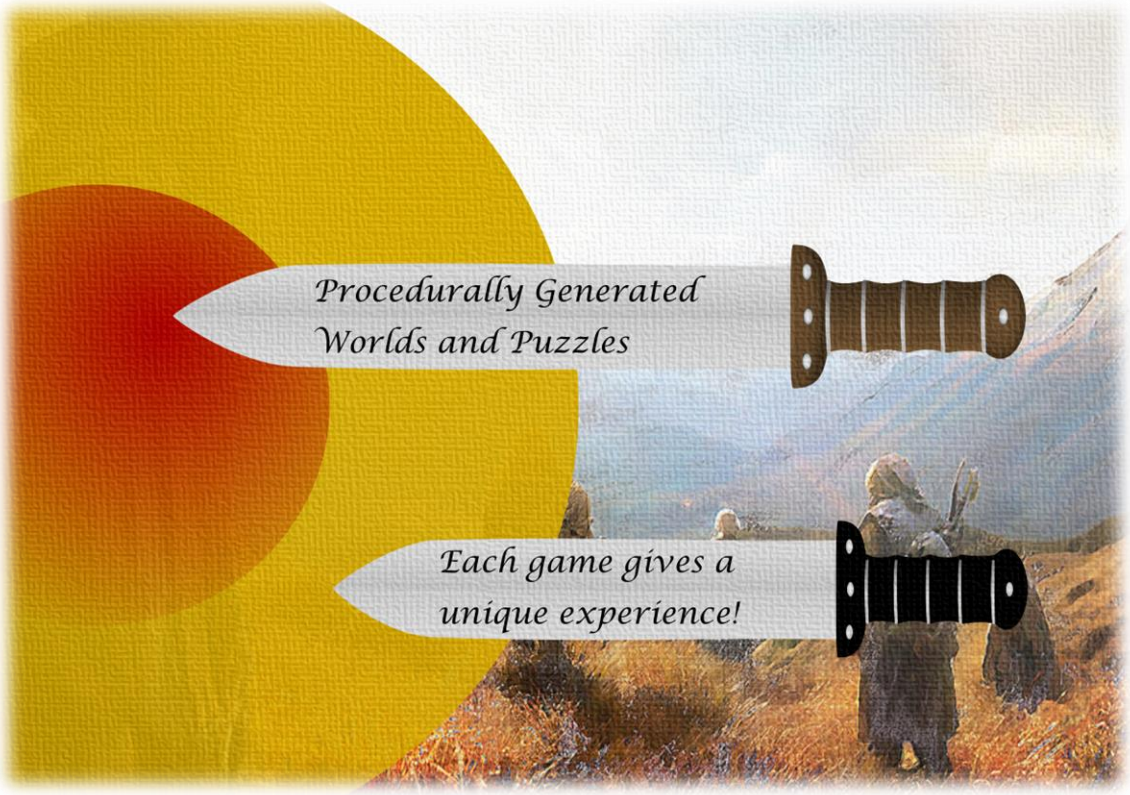


Figure 5 – Character Example and Enemy A.I.

“Big Idea” Bullseye



Assessment

The selling point of RogueGen is to live a unique adventure together with friends. Each time a game is started, the world will have changed, and new challenges appear to provide always a new experience. While playing, the users should help each other in combat, where each player can choose their character loadout, and puzzle solving. Therefore, cooperation is an essential part of RogueGen.

Development Schedule

Functional Minimum:

Task	Description	Who?	Hours	Actual
Handmade Map	3D modelled map for playtesting	Andreas	4	
Simple Puzzle	Handmade puzzle for playtesting	Andreas	1	
3D Assets	Character model and environment models	Andreas	12	
Terrain Generation Structure	Graph structure, Voronoi Diagram and Delaunay Triangulation	Jean	18	
Map Structure	Navigation and game graphs	Johannes	18	
Simple Combat System	Health points, attack, extremely simple A.I.	Florian	18	
Character Controller	Physic Based movement	Andreas	2	

Low Target:

Task	Description	Who?	Hours	Actual
Heightmap with texture blending	Create heightmaps based on noise functions and the puzzle graph	Jean	18	
Environment Filling	Add trees, rocks, bushes, etc... to the map	Johannes	18	
3D Model Textures	Texturing of the previously created models	Andreas	12	
GUI	Player icons, power/weapon selection, helpers, HP	Florian	10	

Desired Target:

Task	Description	Who?	Hours	Actual
GUI Improvement	Minimap, etc...	Any	4	
Heightmap changes	Change the heightmap by adding, removing and editing vertices	Jean	18	
PCG Puzzle	Generate puzzles based on seed using recursion	Johannes	18	
Biome based Environment Filling	Adjust environment assets based on biomes	Jean Johannes	20	
Character animation	Walking, attacking, etc...	Andreas	8	
More Environment Models	Trees, rocks, etc...	Andreas	12	
Advanced Combat System	Physics, more movement, multiple attacks, projectiles, team-work	Florian	26	

High target:

Task	Description	Who?	Hours	Actual
Biome Transition	Better texture blending	Jean	20	
Better Textures	Different biome textures	Andreas	12	
Shader Effects	VFX	Any	20	
Particle Systems	Cool effects	Any	20	
Environment affects characters and A.I.	Sample environment properties based on character position	Florian Johannes Jean	16	

Creative and Assignment Tasks:

Task	Description	Who?	Hours	Actual
Brainstorming	Game idea and elements decision	All	4	
Game Pitch Document	Write the document	All	4	
Game Pitch Presentation	Create PowerPoint presentation	All	1	
Prototype Game Creation	Art Attack time	All	12	
Prototype Notebook	Write document	All	4	
Prototype Presentation	PowerPoint	All	1	
Interim Report	Write	All	2	
Interim Presentation	PowerPoint	All	1	
Alpha Release Report	Write Document	All	2	
Alpha Release Presentation	PowerPoint	All	1	
Playtesting with others	Friends, family, dog and cat	All	1	
Playtesting Report	Write about playtests, collect screenshots	All	4	
Playtesting Presentation	PowerPoint	All	1	
Conclusion Report	Write document	All	2	
Conclusion Video	Gameplay recording and editing	All	4	
Poster Creation	DemoDay poster	All	1	
Conclusion Presentation	PowerPoint	All	1	