

Milestone #2 Presentation

Group 1 (“Cave labyrinth group”)

Daniel Fürst, Pablo Martínez, Manuel Prinz, Nikolas Schwarz, Philip Zimmermann

July 5, 2019

Contents

- State of the Art
 - Recap
 - Map Image Rendering
 - Map Generation

Contents

- State of the Art
 - Recap
 - Map Image Rendering
 - Map Generation
- Work in Progress
 - Map Generation (continued)
 - Light Baking
 - Enhancing P_2 's Participation

Contents

- State of the Art
 - Recap
 - Map Image Rendering
 - Map Generation
- Work in Progress
 - Map Generation (continued)
 - Light Baking
 - Enhancing P_2 's Participation
- Planned
 - Accessing the final room
 - Movable rocks
 - More sound effects

State of the Art

Recap

- Map
 - consists of a 7×7 hexagon grid
 - There are 13 types of hexagons to cover all possible intersections.
 - Each type can have multiple variants
- Exploration (P_1)
 - Move freely within playing area.
 - Use teleportation to move to places outside playing area.
 - Use torch to illuminate surroundings.
 - Fire emits sound.
 - Screen fades to black if P_1 sticks his/her head into a solid object, e.g. a wall.

Recap

- Markings
 - Draw with chalk to mark locations (P_1).
 - Prominent features are scattered on the map like glowing crystals and mushrooms.
- Co-op
 - Players communicate verbally.
 - P_1 can describe the cave to P_2 .
 - P_2 can identify tiles and puzzle them together.
 - P_2 can help P_1 to navigate.

Map Image Rendering

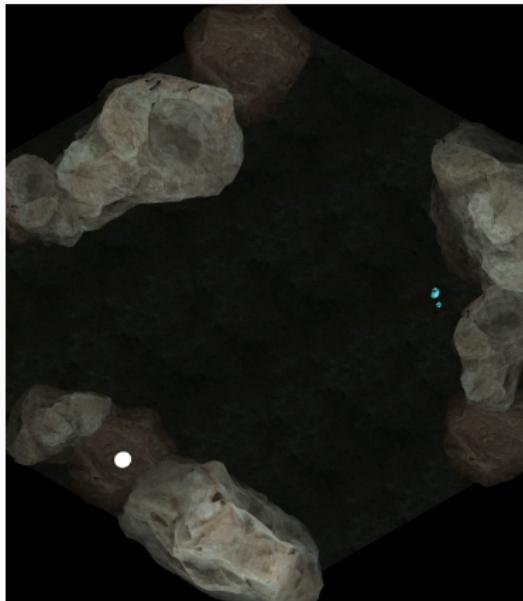
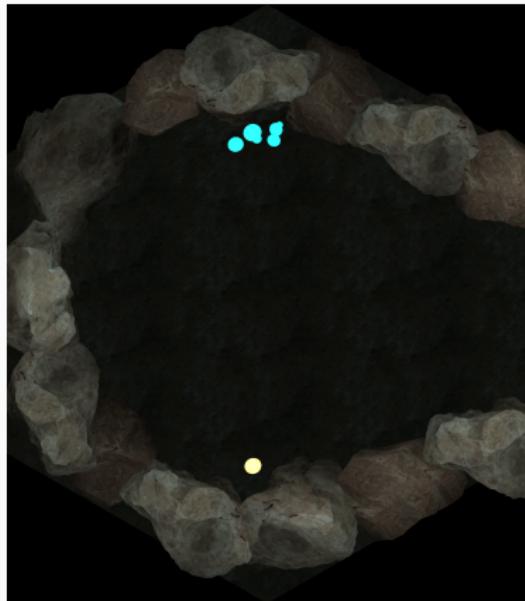


Figure 1: The hexagon image render we decided on.

Map generation

Task

- Create a connected map where every hexagon is accessible to the player.
- Create an interesting layout.
- Create at most a certain amount of instances of each prefab because P2 will have a limited amount of cardboard tiles.

Map generation

Task

- Create a connected map where every hexagon is accessible to the player.
- Create an interesting layout.
- Create at most a certain amount of instances of each prefab because P2 will have a limited amount of cardboard tiles.

Idea: Overlay two spanning trees to create a map

- Initialize two full grids with random edge weights
- Calculate minimum spanning tree of each grid
- Merge the two trees

Map generation

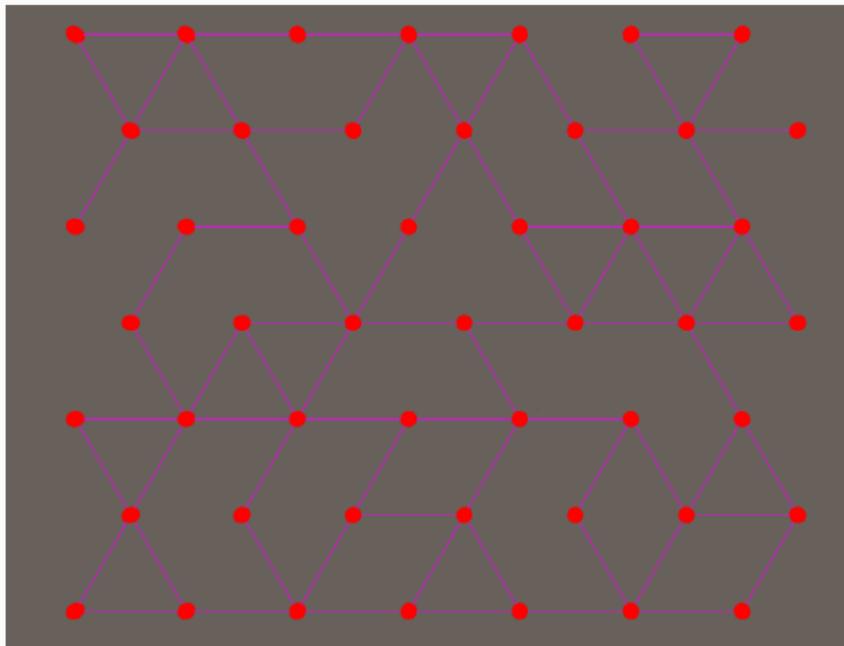


Figure 2: Current result of the map generation (debug mode, 1)

Map generation

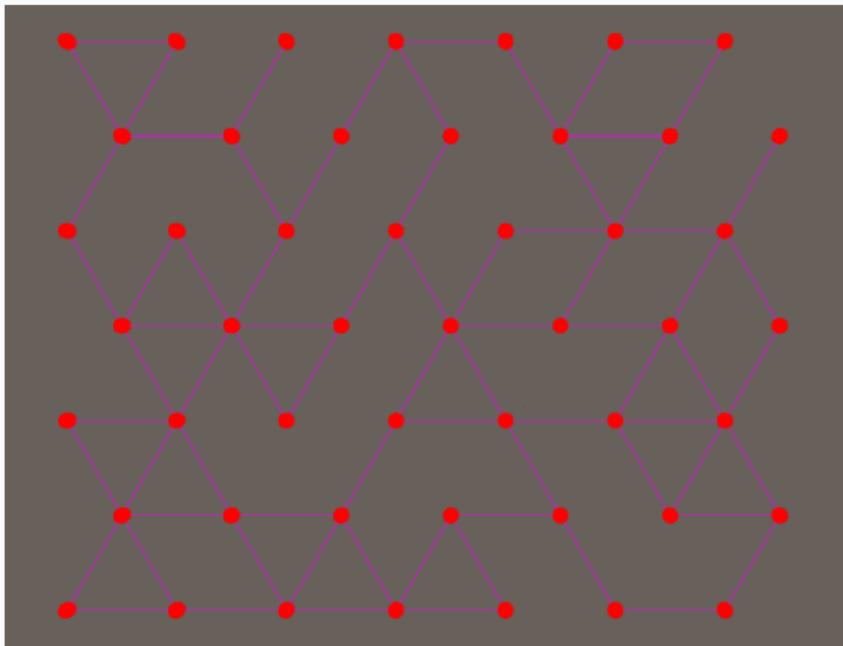


Figure 3: Current result of the map generation (debug mode, 2)

Map generation

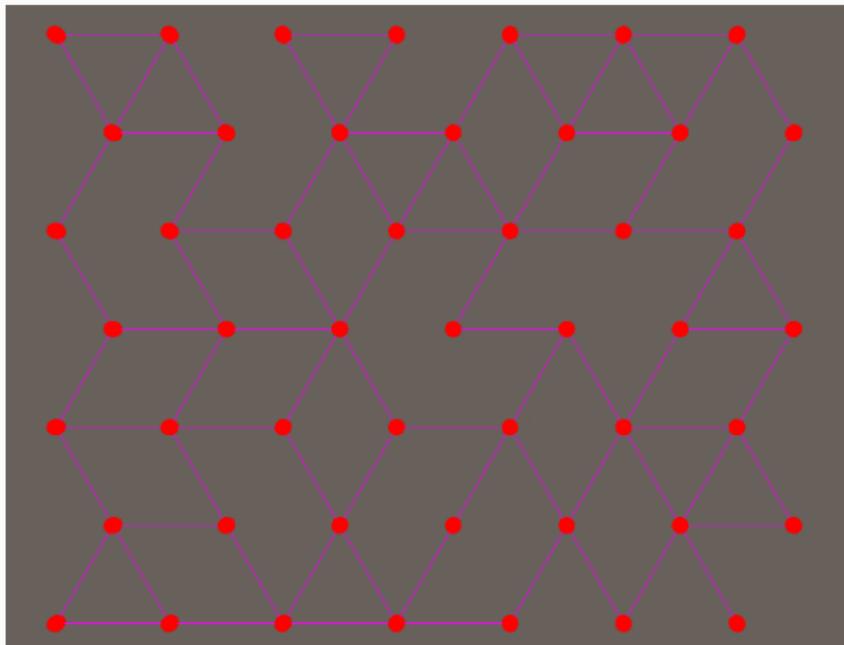


Figure 4: Current result of the map generation (debug mode, 3)

Map generation

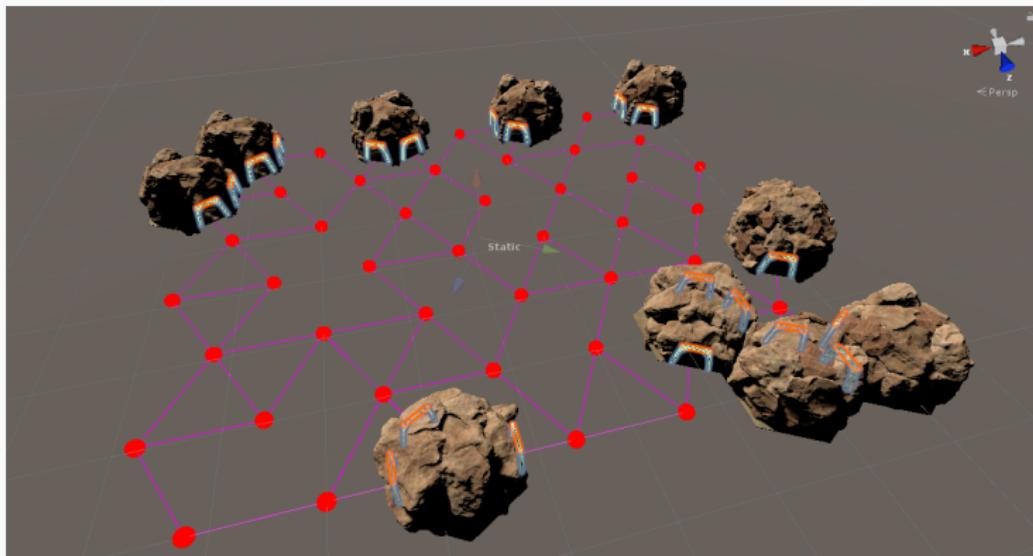


Figure 5: Current result of the map generation with some prefabs loaded

Map generation



Figure 6: Current result of the map generation – inside

Work in Progress

Map generation

Problem: The last constraint of not exceeding a maximum amount is not met.

Map generation

Problem: The last constraint of not exceeding a maximum amount is not met.

Proposal for solution: Augment result by inserting and removing half-edges. (WIP)

Map generation

Problem: The last constraint of not exceeding a maximum amount is not met.

Proposal for solution: Augment result by inserting and removing half-edges. (WIP)

- First, try to insert half-edges to fulfill constraints until no longer possible
 - easy to do
 - will usually not change the labyrinth
 - does not compromise connectedness

Map generation

Problem: The last constraint of not exceeding a maximum amount is not met.

Proposal for solution: Augment result by inserting and removing half-edges. (WIP)

- First, try to insert half-edges to fulfill constraints until no longer possible
 - easy to do
 - will usually not change the labyrinth
 - does not compromise connectedness
- If necessary, remove half-edges
 - might compromise connectedness
 - hard to fix if connectedness is compromised

Light Baking



Figure 7: Current state of lighting from generated maps – inside

Light Baking

Problem: Light baking does only work if the hexagons are already in the scene.

Light Baking

Problem: Light baking does only work if the hexagons are already in the scene.

Solution:

- Do not spawn hexagons at runtime.
- For each hexagon place the maximum amount in the scene.
- Relocate hexagons after level generation has finished.

Enhancing P_2 's Participation

- Ensure game's completion depends on P_2
- Add unique symbol to each hexagon
- Lock labyrinth's final room

Enhancing P_2 's Participation

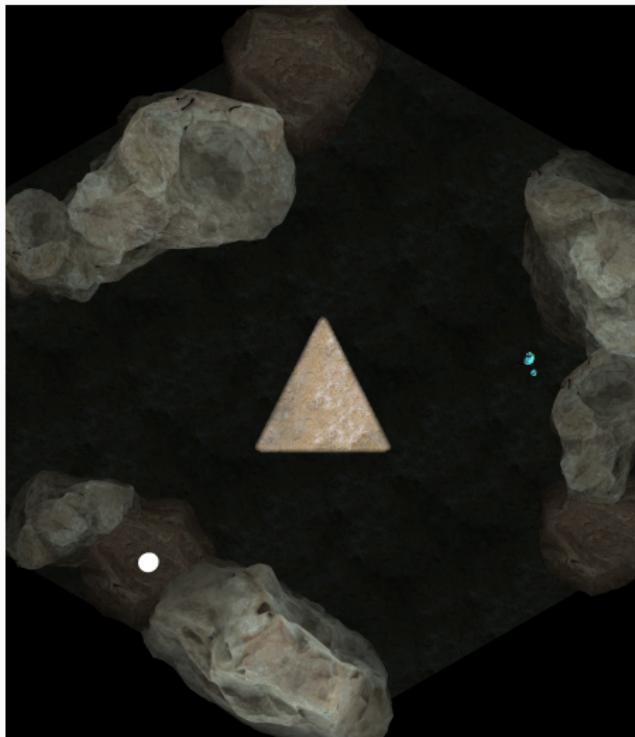


Figure 8: The hexagon image render with a unique symbol.

Planned

- Accessing the final room
 - Think of and implement a way for P_2 to unlock the final room.
 - Hard to trigger by accident
 - Hard to find by brute force
 - Should fit the setting
 - **Proposal:** Unlock by placing rocks at certain locations.
- Movable rocks
 - Give the player another option to mark locations.
- More sound effects

Thank you for your kind attention!

Any comments or questions?