

PROJECT VRIJ III

MIDTERM

Carl Peters

INHOUD

- The Idea -

- Project Scope and Planning -

- Goals -

- Research -

- Coding -

- Art -

- Plan and Outlook -

What is **DUNGEON DISASTERS**

- Two to four player Co-op
- Four unique classes
- Procedurally generated dungeons
- Player competition through ranking

THE TASK

Project Scope

- + Procedural Dungeon Generator
- + Concept art for varying dungeon themes
- + One or two fully produced 3D tilesets

THE PLAN

Rough outline

- Refine design
- Repertoire research
- Designing dungeon system
- Building the system
- Concepting dungeon themes
- Concept art
- Asset production
- Implementation
- Polish

THE GOAL

Personal goals

- Coding Adventure
- Procedural content
- Autonomy
- Environment Art
- Concept & Production
- Production-ready

RESEARCH

REPERTOIRE RESEARCH

<i>Game</i>	<i>Diablo III</i>	<i>Binding of Isaac</i>	<i>Link's Awakening 2019</i>	<i>Spelunky</i>
<i>Level</i>	Large pre-built segments random events	Pre-built room layouts Painted backgrounds	Pre-built square rooms	room layouts fill fixed 5x5 grid texture tileset
<i>Features</i>	Set Entrance/Exit Large open spaces	Set Entrance/Exit Special Event Rooms Secret Rooms	Keys & locked doors	Set Entrance/Exit Dynamic open space Secret Rooms
<i>Limitations</i>		All rooms are the same size No interconnected paths	All rooms are the same size	Rooms may not properly connect
<i>Technical Complexity</i>	Low, set rooms laid out on grid	Low, set rooms laid out on grid		Mid, requires automated tileset
<i>Artistic Complexity</i>	High, each segment is pre-made and highly detailed	Low, premade backgrounds overlaid with object sprites	Mid, simple grid-based pre-made rooms	

CONCLUSION

How do i want my dungeons to look?

- Simpler, room-based system
 - Pre-designed rooms, no corridors
 - Reminiscent of 2D Zelda dungeons
 - Simpler algorithm
- “Classic” Tile system
 - But 3D
 - No automation
 - Reminiscent of limited tile palettes of Gameboy/NES games
 - Alt versions of tiles -> variation

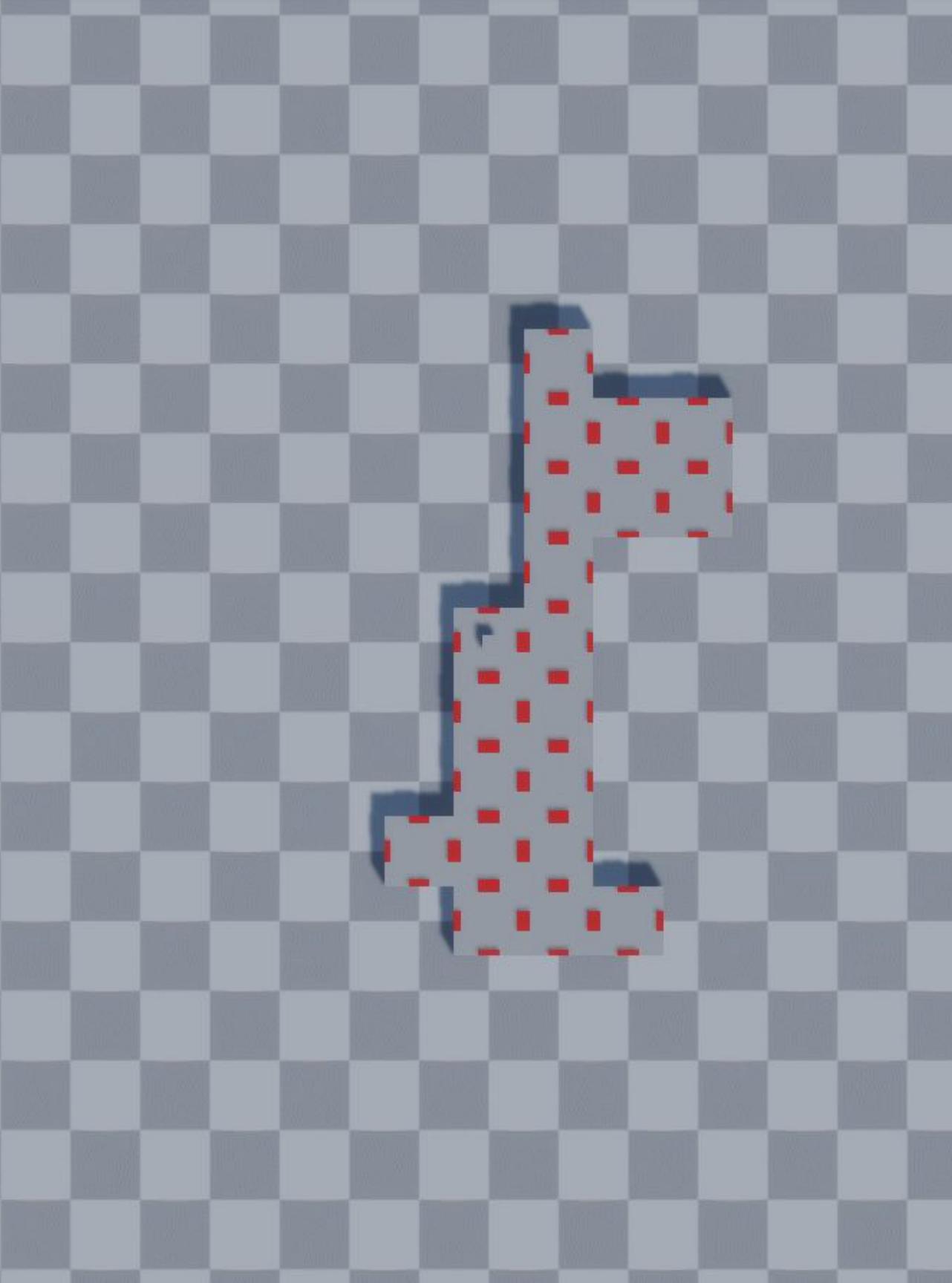


MAKING THE ALGORITHM

ALGORITHM 1.O

1. Runner sets point (“room”) in bool array as true
2. Chooses a new point in a random direction
3. New runner gets created at that new location
4. Repeat until maximum distance is achieved

```
3 Verweise
public static RoomGrid[,] RunnerBasic(RoomGrid[,] dungeon, Vector2 runnerPosCurrent, int runnerDistance) //Runner with random direction, can go back on itself
{
    // Set given position as true
    dungeon[(int)runnerPosCurrent.x, (int)runnerPosCurrent.y].Enable();
    print("Map Array Entry " + (int)runnerPosCurrent.x + "," + (int)runnerPosCurrent.y + " has been set to: " + dungeon[(int)runnerPosCurrent.x, (int)runnerP
    Vector2 runnerPosNext = runnerPosCurrent;
    Vector2 newDirection = RandomDirVector(); // Get new direction
    runnerPosNext += newDirection; // Set next runner's position in array
    // Start next runner if not at goal
    if ((runnerDistance - 1 > 0))
    {
        if (((runnerPosNext.x >= dungeon.GetLength(0) - 1) | (runnerPosNext.y >= dungeon.GetLength(1) - 1) | (runnerPosNext.x < 1) | (runnerPosNext.y < 1)))
        {
            print("Whoops! Out of bounds");
        }
        else
        {
            print((runnerDistance - 1) + " steps to go");
            dungeon = RunnerBasic(dungeon, runnerPosNext, runnerDistance - 1);
        }
    }
    return dungeon;
}
```



ALGORITHM 2.0

- Backtracking disabled
- Boolean replaced by custom class
 - Rooms now know their neighbors
- Entrance, Boss, and Chest room are placed
- Scriptable Object stores room set
- Dungeon Controller selects prefab based on neighbors

```
28 Verweise
public class RoomGrid
{
    public bool isEnabled = false;      //Is there a room in this spot?

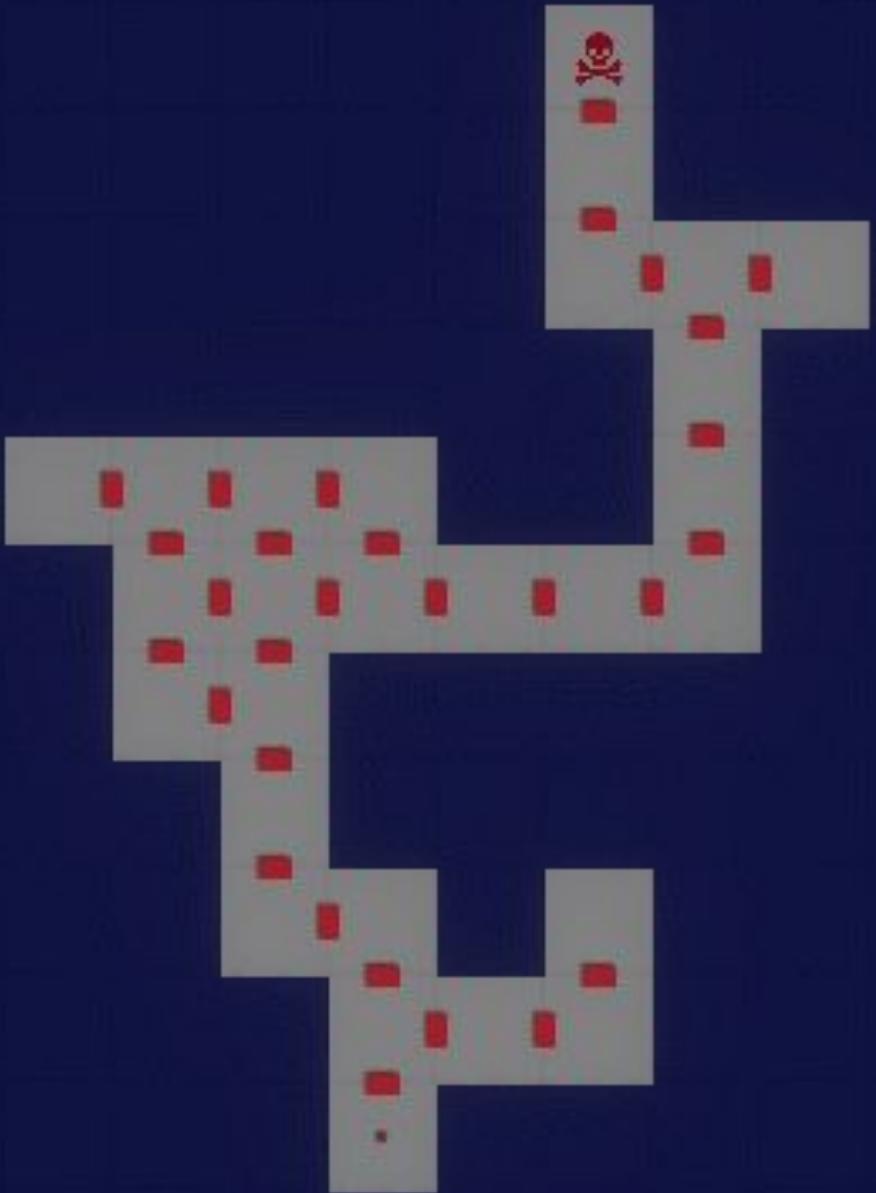
    public int ID = 0;                  //Room Type ID (Normal, Start, Boss)

    public GameObject roomAsset = null; //Assigned Room Prefab

    public bool hasRoomUp = false;      //Neighboring Rooms
    public bool hasRoomRight = false;
    public bool hasRoomDown = false;
    public bool hasRoomLeft = false;

    7 Verweise
    public void Enable()
    {
        isEnabled = true;
    }
}
```

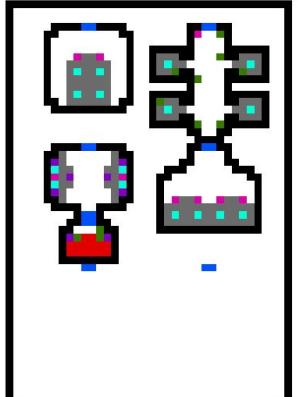
```
public class DungeonDict : ScriptableObject
{
    [Header("Entrances")]
    public GameObject[] Entrance;
    [Header("Boss Rooms")]
    public GameObject[] BossUp;
    public GameObject[] BossRight;
    public GameObject[] BossDown;
    public GameObject[] BossLeft;
    [Header("End Rooms")]
    public GameObject[] EndUp;
    public GameObject[] EndRight;
    public GameObject[] EndDown;
    public GameObject[] EndLeft;
    [Header("Corridors")]
    public GameObject[] CorridorHorizontal;
    public GameObject[] CorridorVertical;
    [Header("Corners")]
    public GameObject[] CornerUpRight;
    public GameObject[] CornerUpLeft;
    public GameObject[] CornerDownRight;
    public GameObject[] CornerDownLeft;
    [Header("T Sections")]
    public GameObject[] TUp;
    public GameObject[] TRight;
    public GameObject[] TDown;
    public GameObject[] TLeft;
    [Header("Center Rooms")]
    public GameObject[] Center;
    [Header("Error")]
    public GameObject ErrorRoom;
```



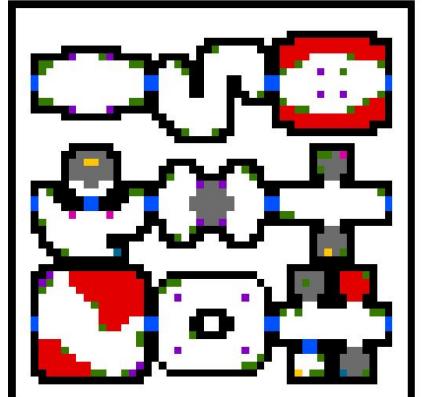
ROOMS & TILESET

- Blockout tileset for testing purposes
- Scriptable Objects store tilesets
- Room prefabs made of tile controllers
- Tiles have type IDs
- Controller spawns tile asset pools
- Tile controllers pull assets from pools when room is entered

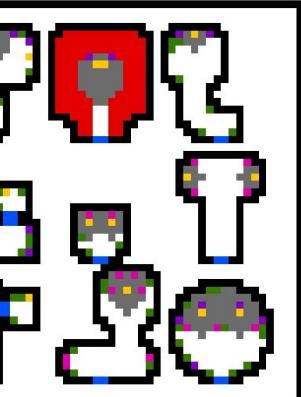
START



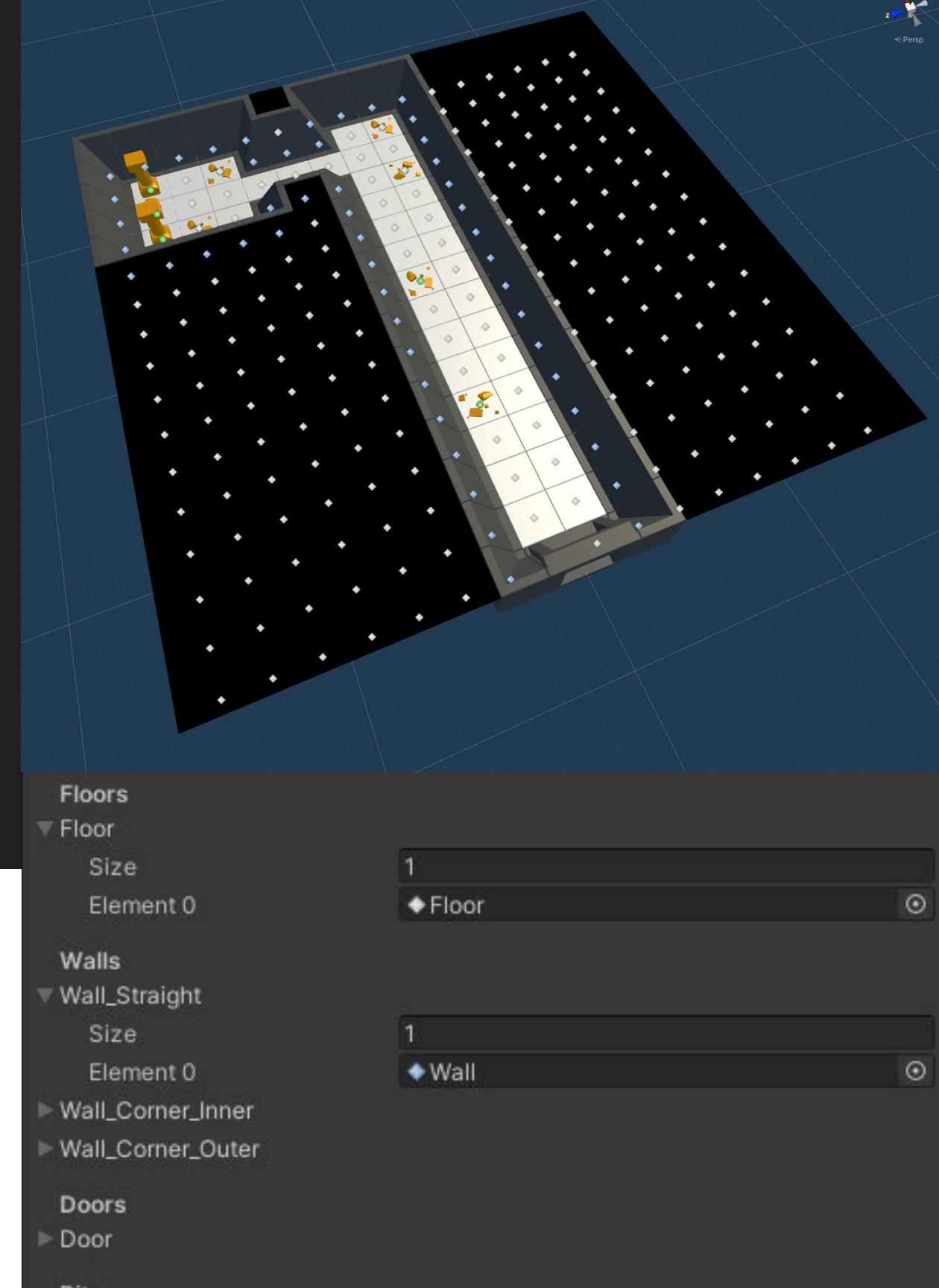
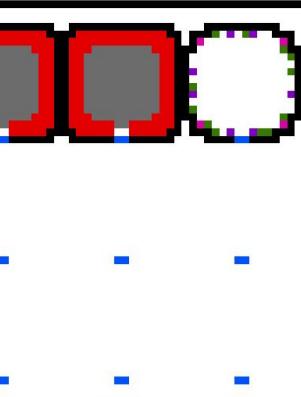
CORRIDOR



END



BOSS





CONCEPT ART

Dungeon Themes

MOODBOARDS

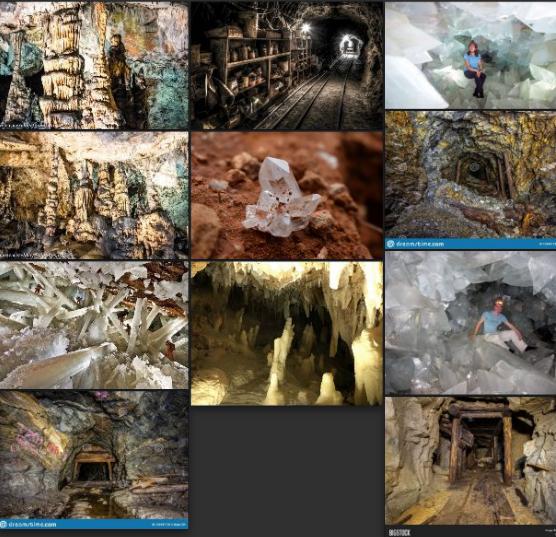
Jungle Temple

- Hindu/Buddhist ruins
- Overgrown temple structure
- Deep, dark rainforest feed
- Roots taking over walls/floor
- Water in pits



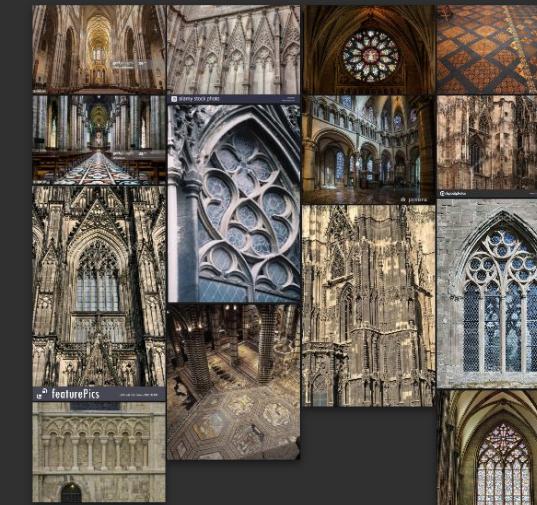
Crystal Cave Mines

- Natural crystalline caves with mine elements
- Crystals illuminate environment
- Oversized, imposing crystals
- Spiky, dangerous crystals & stalagmites in pit
- Holes ate stalagmites



Gothic Cathedral/Abbey

- Light-filled but almost too pristine gothic architecture
- Patterns everywhere
- High walls
- Stained glass windows



Necrotic Tombs

- Dark, damp, and oppressive
- Slimy dark mold dripping from the walls
- Large rough stones
- Stacks of bones in tomb holes
- the occasional skull or bone
- cobwebs



Enchanted Forest

- Deep, luminous European style forest
- Forest floor with moss, leaves, and stones
- Neolithic structures scattered about
- Walls, floors, built-in stone walls
- Enchanted feel, oversized vegetation like mushrooms
- small rivers, waterfalls, maybe?
- pits are ponds



Temple of the Fire God

- orange & red tints
- Lots of lava & volcanic rock
- Mesopotamian style temple elements
- angular smooth structures contrast flowing natural lava
- lava streams from walls
- pits are bubbling lava



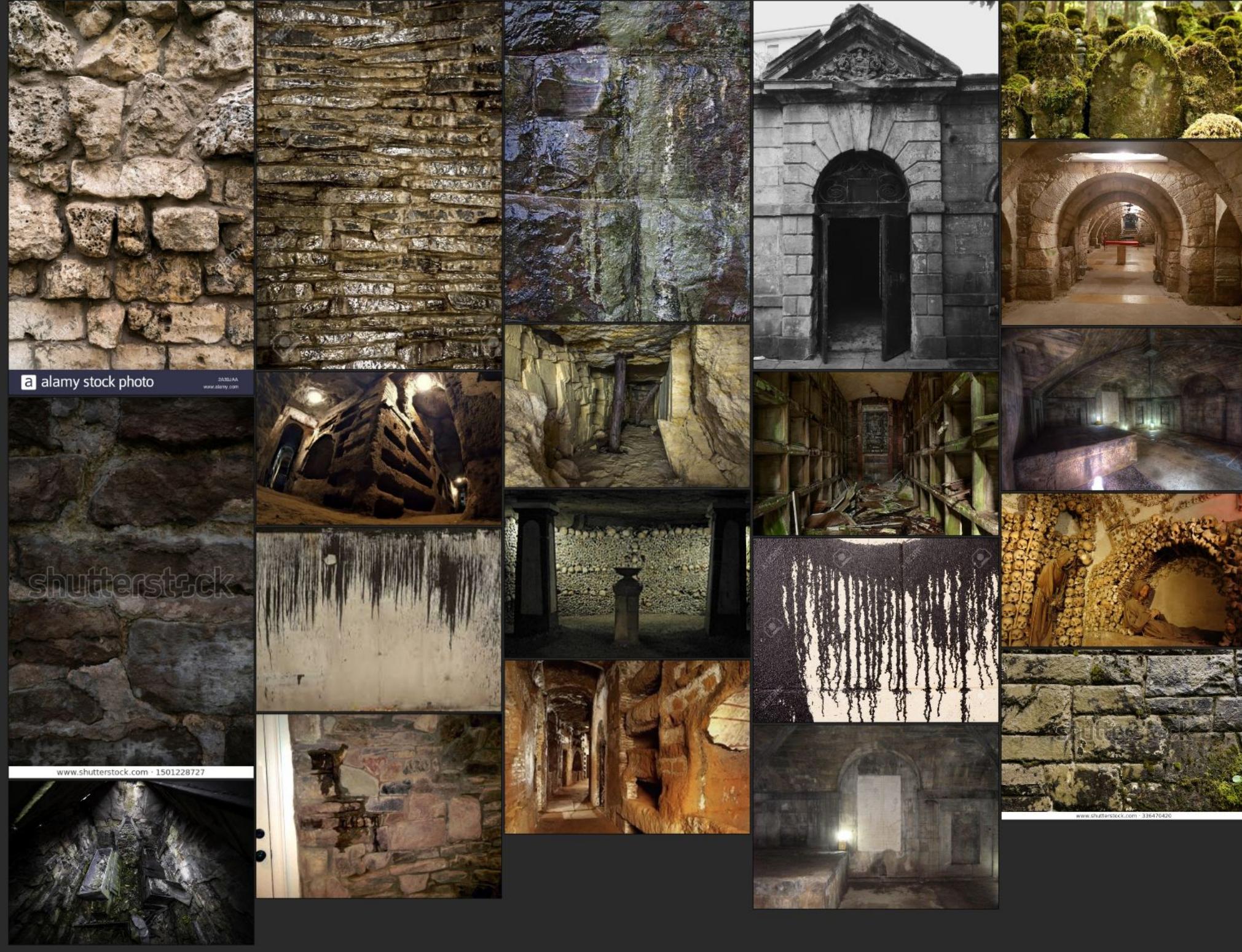
JUNGLE TEMPLE

- Rich vegetation
- Hindu & Buddhist ruins
- Overgrown temple structure
- Deep, dark rainforest feel
- Floor is broken stone tiles with roots and plants overgrowing
- Pits are water channels



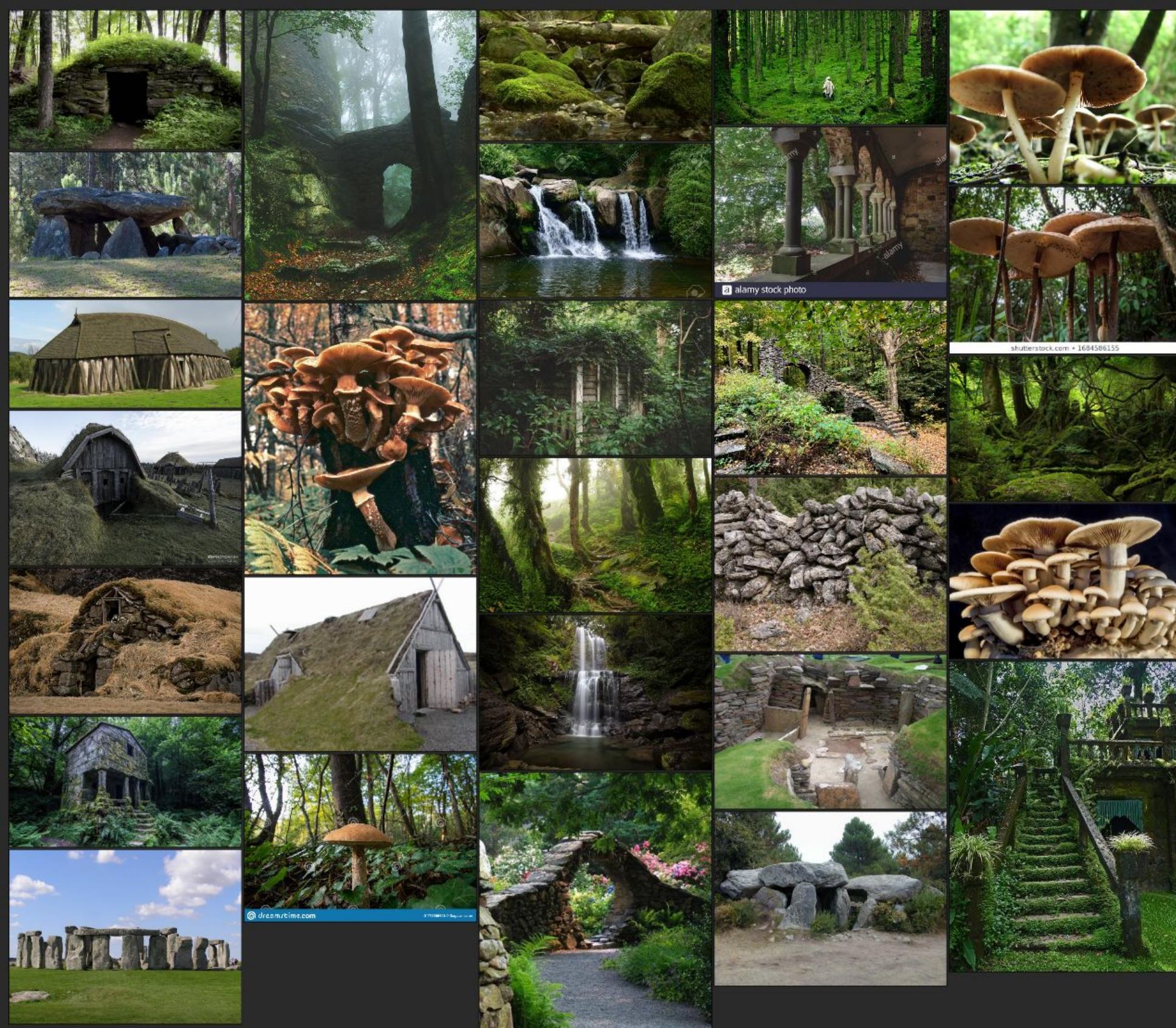
NECROTIC TOMBS

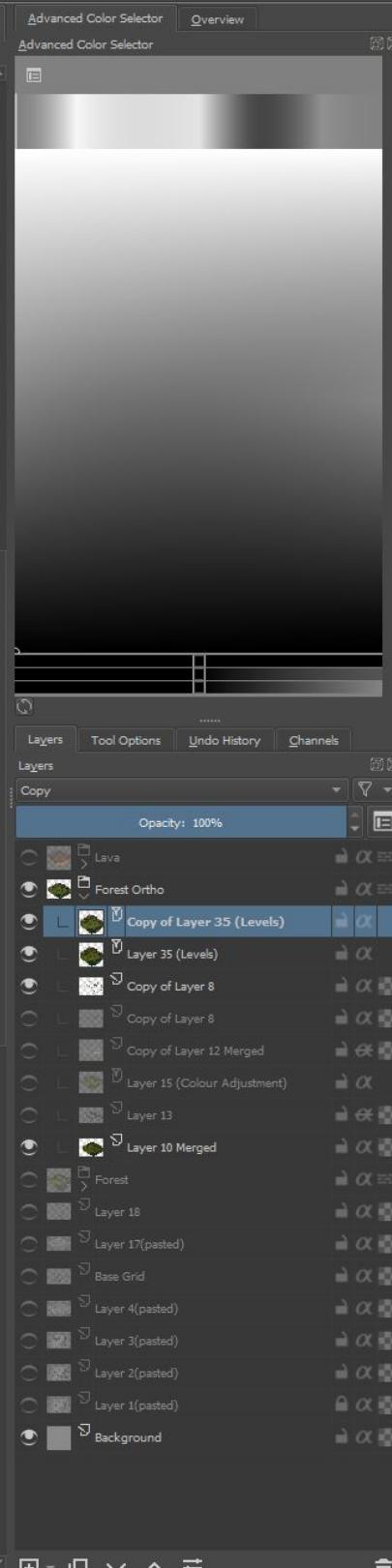
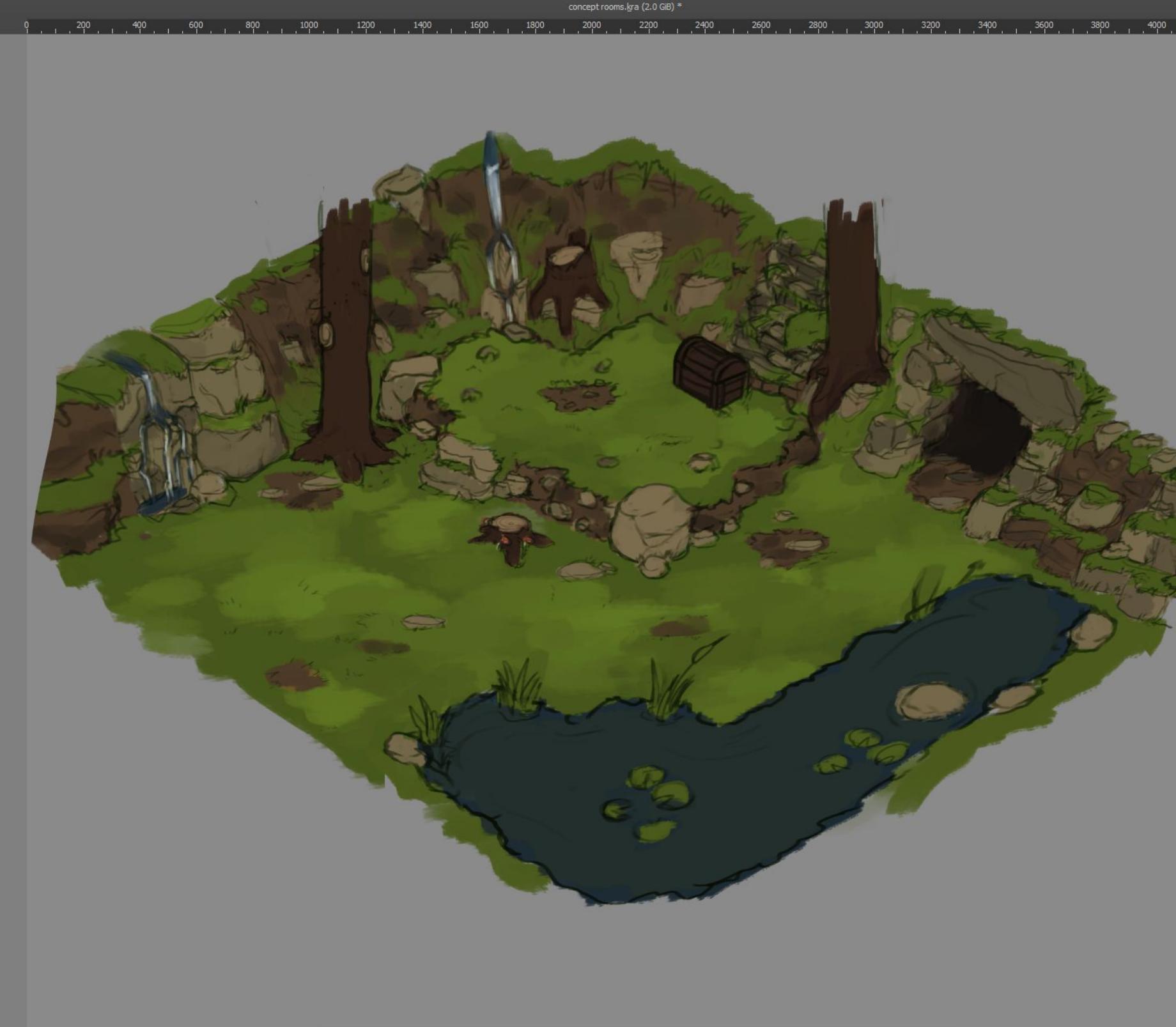
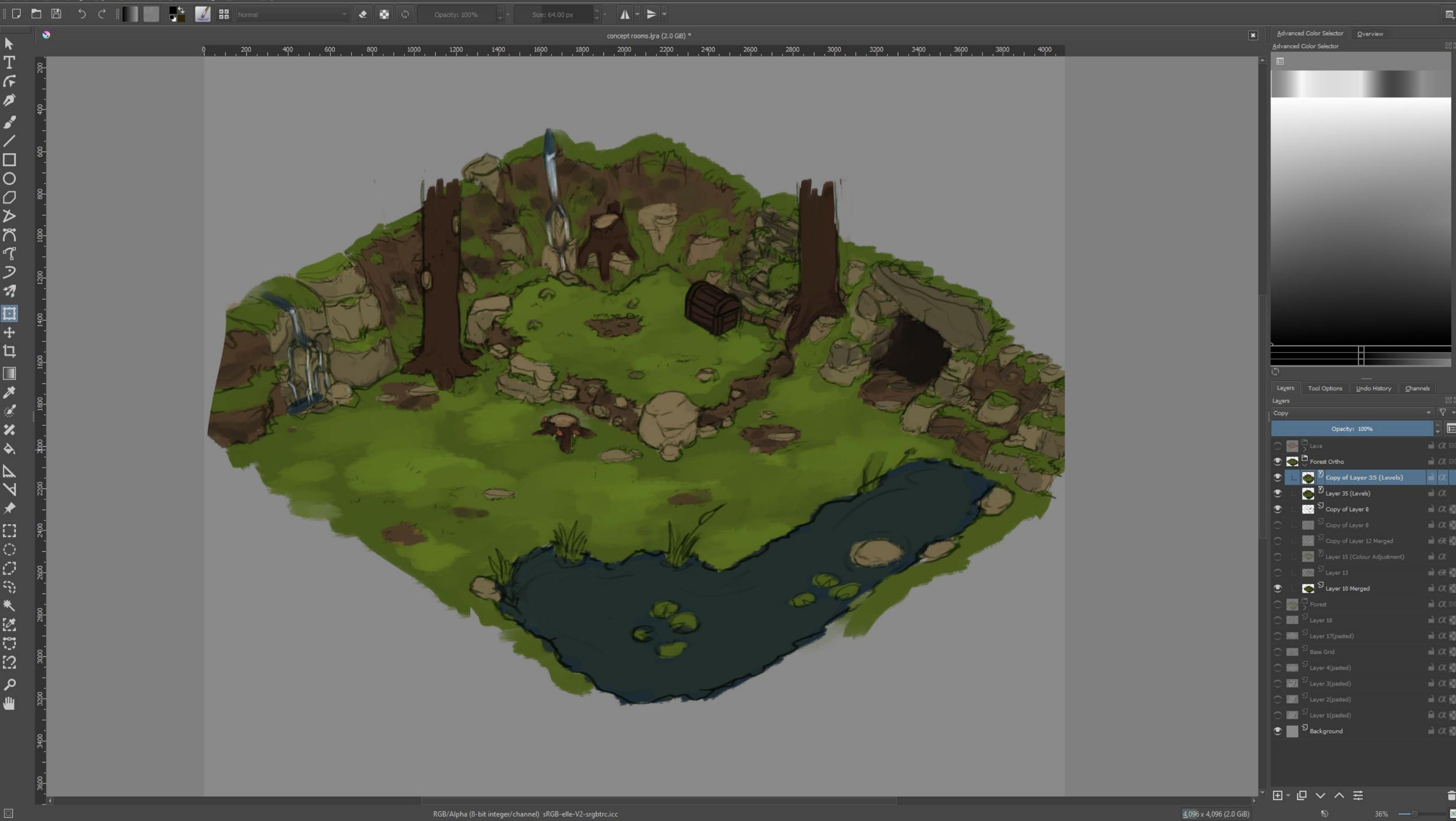
- Dark, damp, and oppressive
- slimy dark mold dripping from the walls
- Large rough stones
- Stone walls with niches
- the occasional skull or bone
- cobwebs

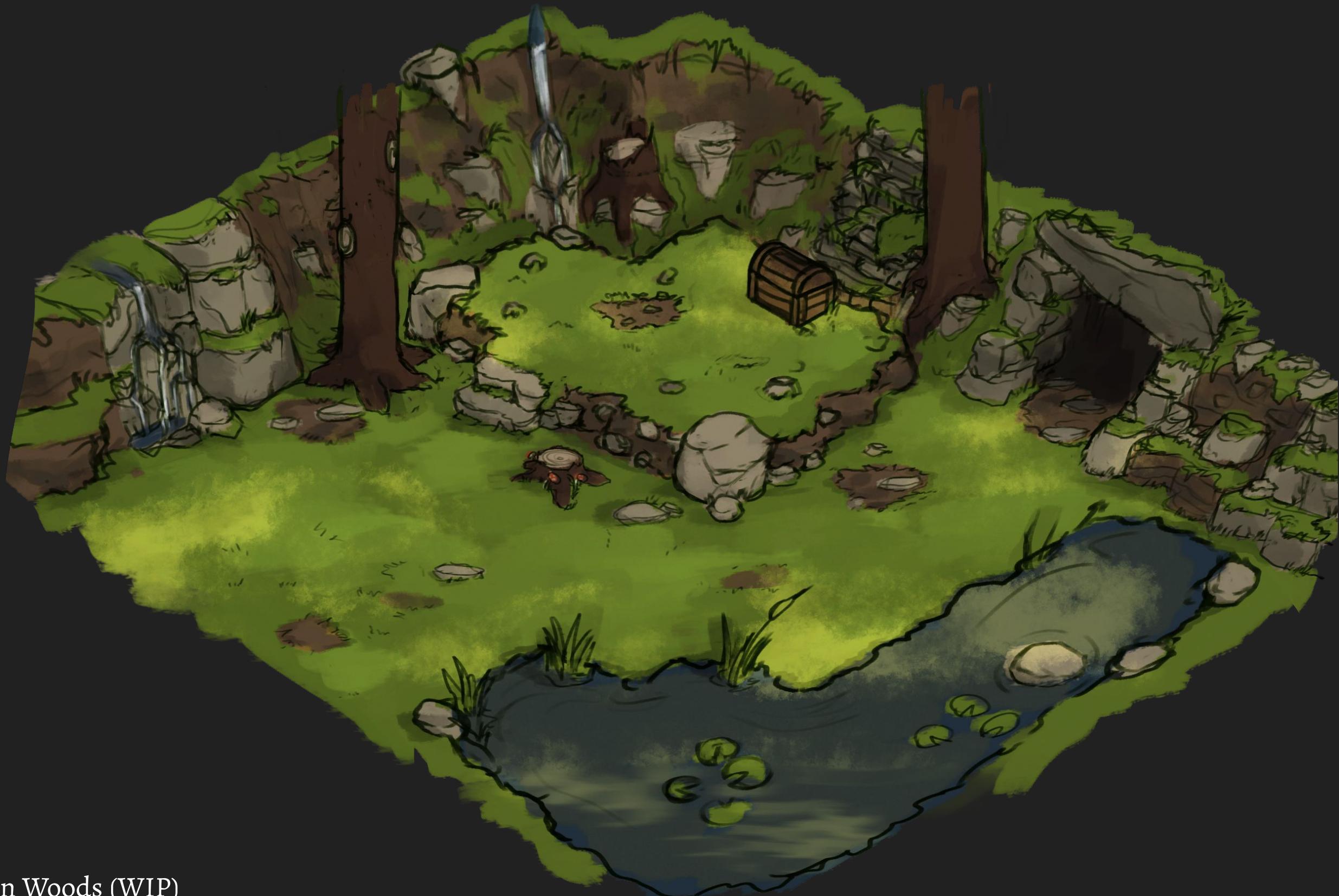


FORGOTTEN Woods

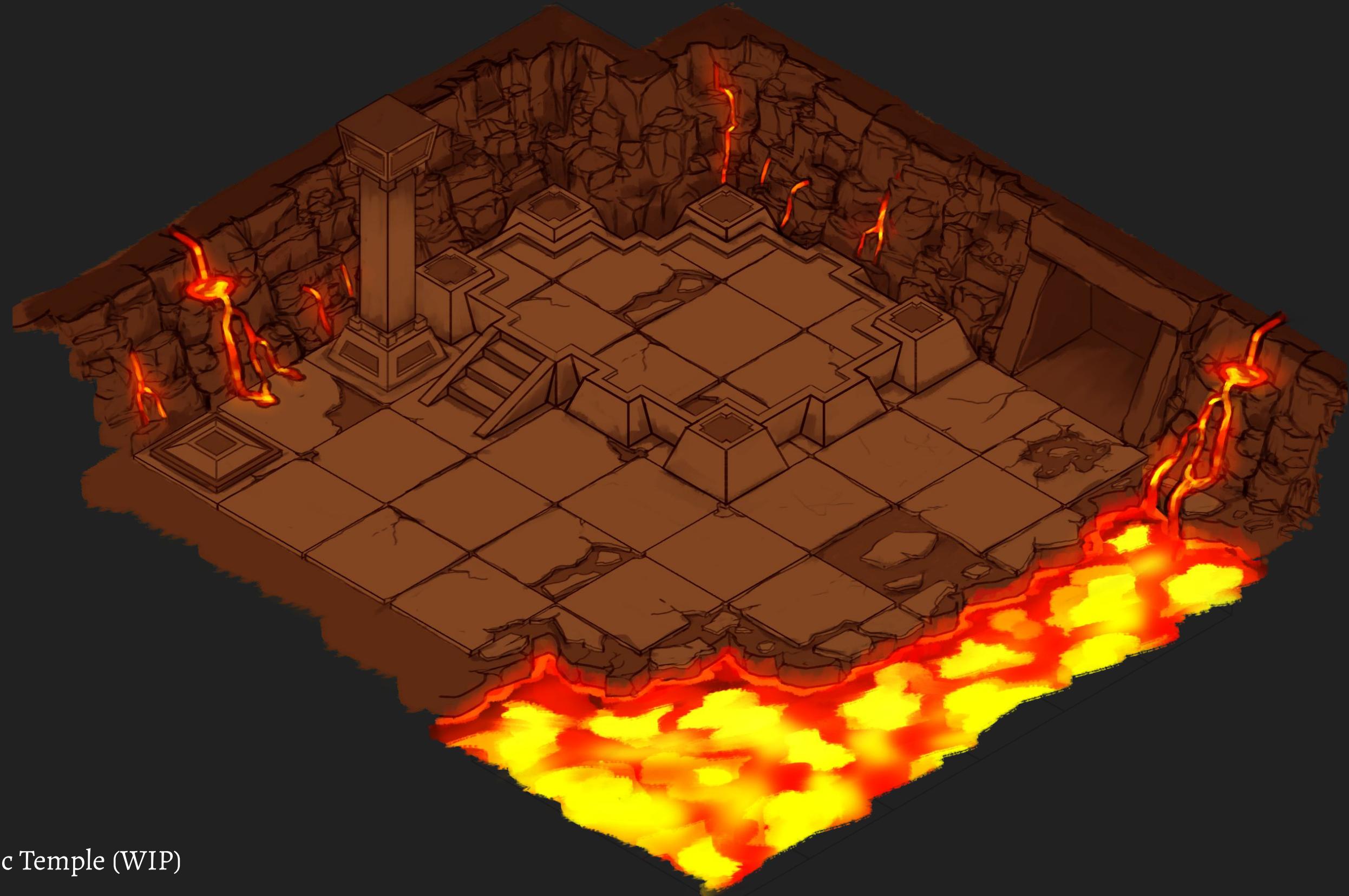
- Deep, luscious woods
- forest floor with moss, leaves, and stones
- neolithic architecture
- Walls are hills, boulders, stacked stone walls
- Enchanted feel, oversized vegetation like mushrooms
- small rivers/ waterfalls maybe?
- pits are ponds



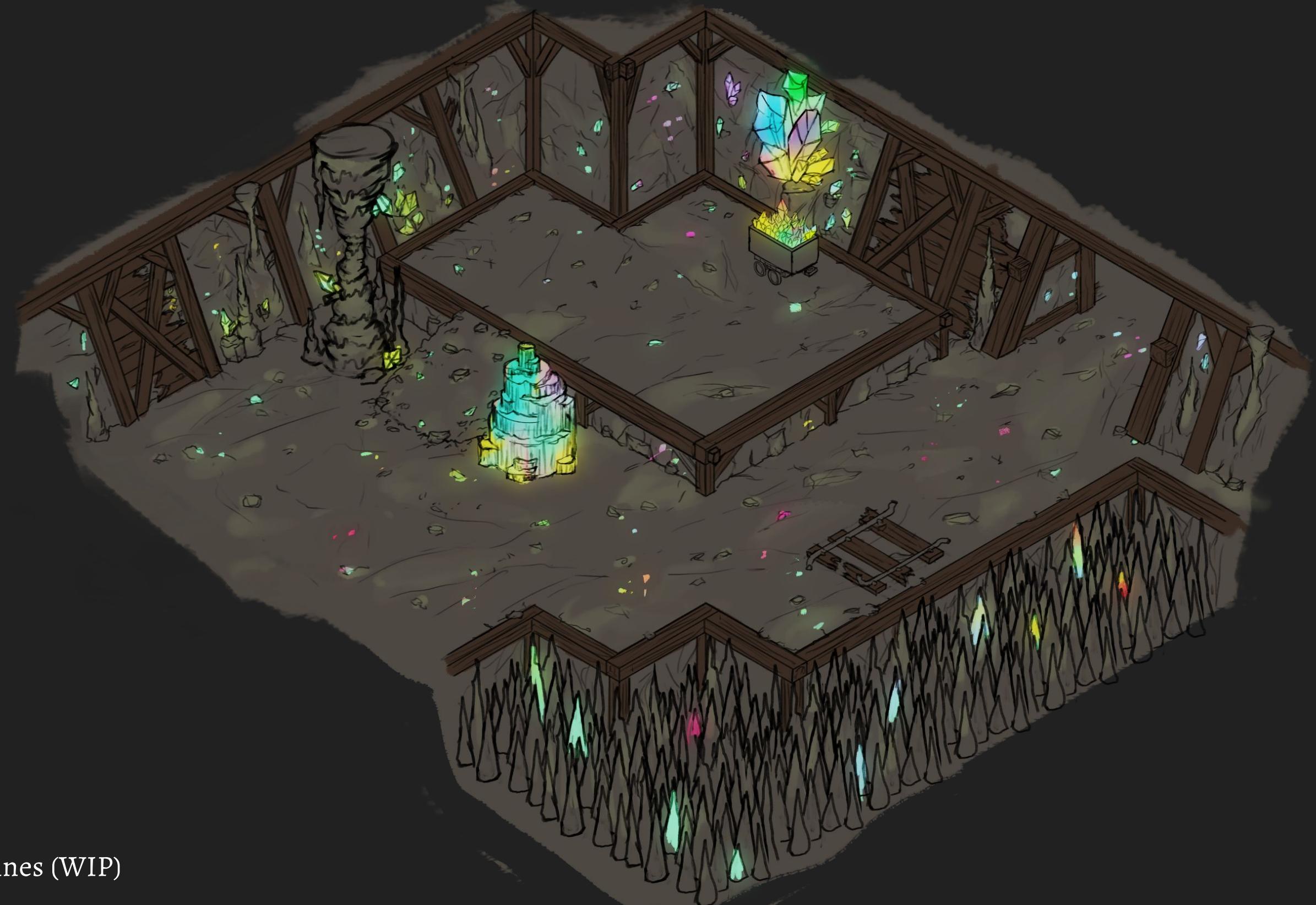




Forgotten Woods (WIP)



Vulcanic Temple (WIP)



Crystal Mines (WIP)

FUTURE

Coming weeks

- More general concept art
- Choosing two themes
- Tileset production:
 - Detail concepts
 - Modelling/sculpting
 - Texturing
- Implementation
 - Theme swap mechanic

**THANKS FOR
LISTENING**

Questions?

