

Reflection

Nicklas van der Aa Kühle - 316453

"During this semester's GMD course, I've had fun with Antonio in navigating the highs and lows of developing a game project from scratch. While not every moment was smooth, with bugs, unexpected behavior, and creative blocks making regular appearances, the process was incredibly rewarding. I've gained a deeper appreciation for the complexity of game systems and especially 2D games. I've tried working with 3D game development before, and I also had the XR course last semester, which I felt was a whole different kind of learning.

While other courses took up a lot of our time alongside GMD, Antonio and I made it through and created a satisfying product from our perspective. From the get go, as stated in the blog posts, we had some bigger ambitions, but went along with a bit smaller project. The reason for this was purely maintaining time and juggling all the other courses while working on GMD.

One of the snippets I wanted to include in my personal reflection was getting the objects we wanted to spawn in, to be in reachable and spawnable areas.

```
private void GatherValidPositions()
{
    validSpawnPositions.Clear();
    BoundsInt boundsInt = tilemap.cellBounds;
    TileBase[] allTiles = tilemap.GetTilesBlock(boundsInt);
    Vector3 start = tilemap.CellToWorld(new Vector3Int(boundsInt.xMin, boundsInt.yMin, 0));

    for (int x = 0; x < boundsInt.size.x; x++)
    {
        for (int y = 0; y < boundsInt.size.y; y++)
        {
            TileBase tile = allTiles[x + y * boundsInt.size.x];
            if (tile != null)
            {
                Vector3 place = start + new Vector3(x + 0.5f, y + 1.5f, 0);
                validSpawnPositions.Add(place);
            }
        }
    }
}
```

For our "Blocks" used to create our playable area, we created a tilemap from an asset we borrowed online. We define the area that we want the method to scan to get all tiles in the area. They are then converted to coordinates in our scene. It then checks the tiles to see if it's a valid tile and then stores the positions that are valid. In our ObjectSpawner, it's then a 50 / 50 chance that either a Mage or a normal melee enemy spawns. With this setup, we have the possibility to choose which enemies or items should spawn randomly in a battle arena.

One thing we really wanted to improve was the animations. The overall movement of our player works well, but visually getting the feedback through animations adds a lot of value to the end product. The idea was to make the game feel more alive and polished, but we ran out of time to fully get it working. We did look into Unity's Animator and started testing some transitions, but between fixing bugs and getting the core gameplay right, animations had to

take a back seat. Alongside the animations, more content in the game would have been nice to have, as there is no real ending to the game right now.

Overall, I think this course has been a great learning experience.

It gave me a better understanding of how to take a game from idea to implementation, not just from a technical perspective, but also in terms of scope, planning, and working with limited time. Even though we didn't manage to add everything we initially envisioned, I'm proud of the result we ended up with, and also tried something completely new, to create a 2D platformer game.