**Week 3 Diary**

**📝 Plan:**

~~1. Randomly generated mazes.~~  ( fixed maps ✅ )

~~2. Move of cameras.~~

3. Control of characters. ✅

4. Collision detection tests. ✅

5. Rules design. ✅

**🚫 Problem:**

LPC1768 has a 16K RAM, which only supports an array smaller than 130\*130. There’s no way but to load maps dynamically. But due to the high writing-time-delay and the requirement of high refreshing rates, this cannot be achieved.

I must change my idea from multiple rooms in one layer into one per layer.

**🎁 Outcome:**

1. Fixed mazes.

Store as const-variables, only occupy ROM space, Fig.1.

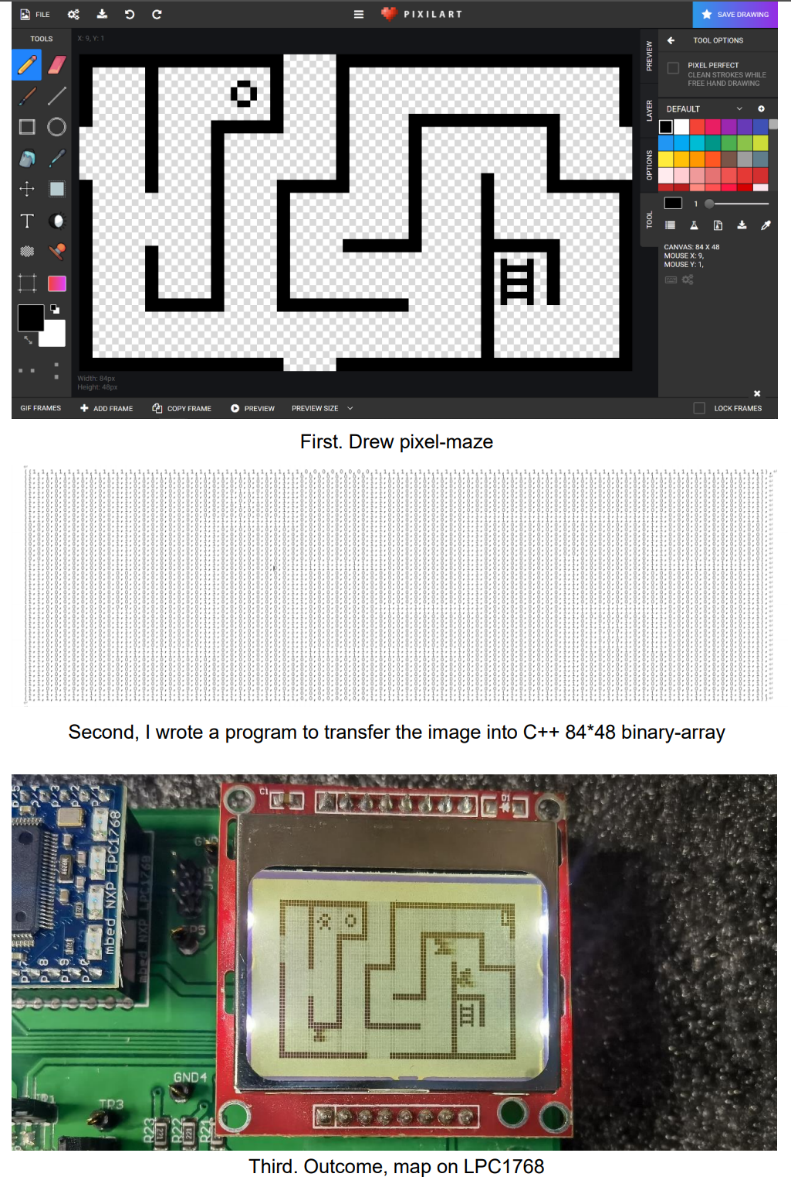


Fig.1 Production of one map

2. Control of characters.

By setting the velocity of x- and y-direction, each frame updates the new position, Fig.2.

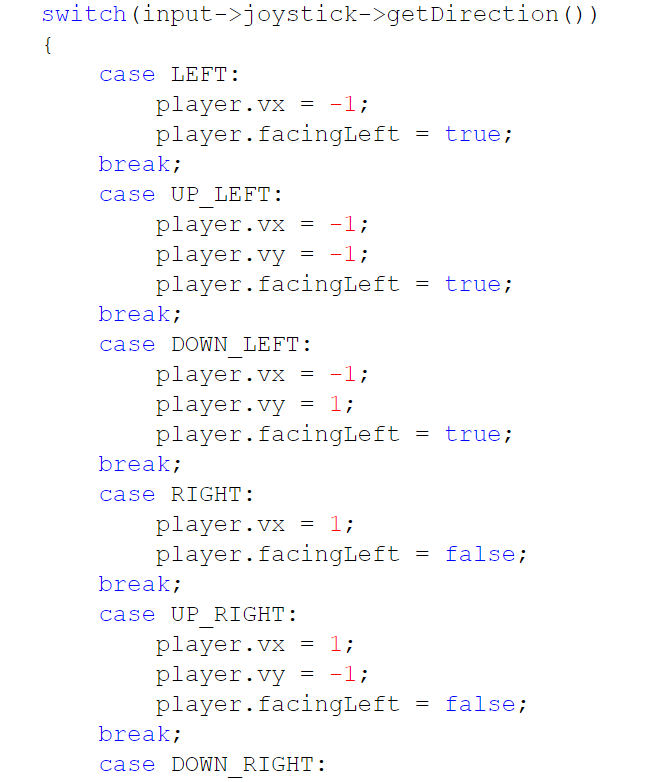


Fig.2 Control (part)

3. Collision detection\*.

Each entity is regarded as a rectangular shape, judging whether they are overlapped, Fig.3.

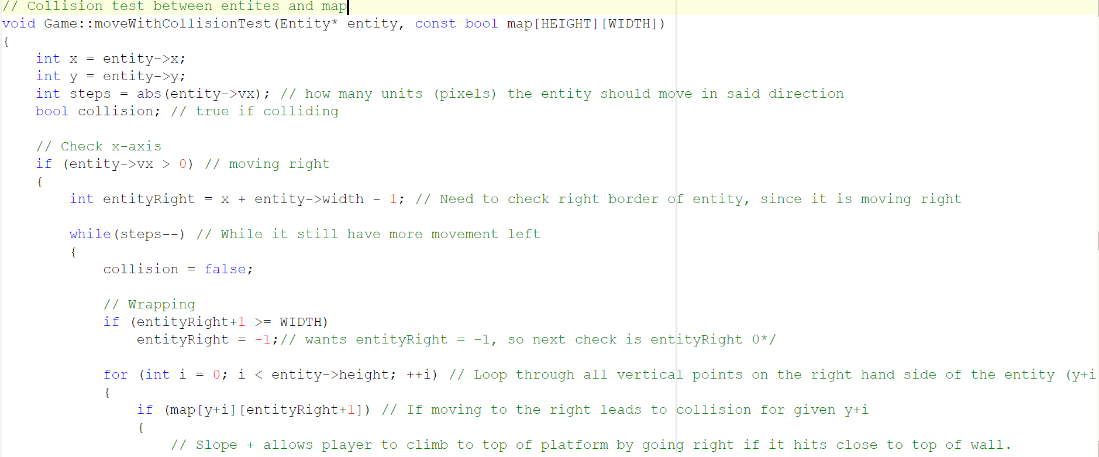


Fig.3 Collision detection (part)

4. Others.

‖ Properties of characters, enemies, gun ballets.

‖ Scoring system.

‖ Games start, refresh, and end conditions.

‖ Sprites of characters and enemies.

**📌 Important:**

I adapted an existing game engine, I referred to the general structure of codes, but I modified a lot. In the final version, I will show the differences in code submission and note the reference.