Week 7 Diary

Plan:

1. Randomly generated coins on maps. ✅
2. Updated sprites ✅
3. Fixed a serious fault on random generator ✅
4. Fixed other display and interactive errors ✅
5. Achievements still not completed. ⚠

Problem:

In LPC1768, due to the lack of real-time clock, rand() function shouldn't be used to generate random numbers, even adding "srand((unsigned)time(NULL));" makes no difference. This problem was found when noticing whenever I restart the processor, those coins which should be scattered randomly on maps only appeared on fixed points. This issue can be solved by introducing "analog.read()" to simulate random events.

```
13 unsigned int Game::random_generator(void)
14 {
15
16     unsigned int x = 0;
17     unsigned int iRandom = 0;
18     for (x = 0; x <= 32; x += 2)
19     {
20         iRandom += ((analog.read_u16() % 3) << x);
21         wait_us(10);
22     }
23     return iRandom;
24 }
```

Figure.1 Implementation of random generator↵

Outcome:

1. Random coins (Fig.2)

Coins scattered on random positions when you try entering a map again. In future, there'll be random respawn points, success points, even random maps.

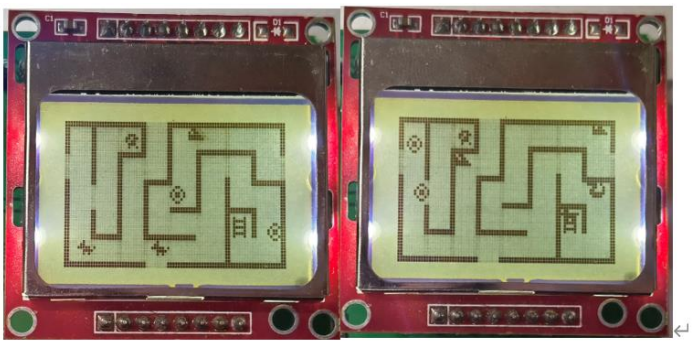

Figure.2 Randomly distributed coins↵

Codes are shown in Fig.2-x.

```
103             x = random_generator() % 84 + 1;
104             y = random_generator() % 48 + 1;
105             i = -1;
106             j = -1;
107             lines = 6;
108             columns = 6;
109             flag = true;
110             // Start detect points
111             if (x + columns > 84 || y + lines > 48)
112                 continue;
113             for (; i < columns + 1 && flag; i++)
114             {
115                 for (; j < lines + 1 && flag; j++)
116                 {
117                     switch (gameProperty.currentLayer)
118                     {
119                     case 1:
120                         if (map1[y + j][x + i])
121                             flag = false;
122                         break;
123                     case 2:
124                         if (map2[y + j][x + i])
125                             flag = false;
126                         break;
127                     case 3:
128                         break;
129                     }
130                 }
131             }
132             if (flag)
133             {
134                 count++;
135                 mapProperty.goldCoinPosition[count - 1][0] = x;
136                 mapProperty.goldCoinPosition[count - 1][1] = y;
137                 if (count == mapProperty.goldCoinNum)
```

Figure.2-x Codes on the implementation of random generated coins. Basically, the random generator returns random integer values (x and y), by checking the point (x, y) surroundings whether contain walls (not fully completed, since here should also check if the coin overlaps a ladder or others), if contains, then fails and go back to the loop; if success, and meets the number defined in mapProperty, break the loop.

2. Updated sprites (Fig.3)

Figure.3 Sprites

3. Other structural modifications.

Only in the Game.cpp, there are 1617 lines.

```
1605        }
1606    }
1607}
1608
1609void Game::respawnPlayer()
1610{
1611    gameProperty.player.x = mapProperty.respawnPosition[0];
1612    gameProperty.player.y = mapProperty.respawnPosition[1];
1613
1614    gameProperty.player.vx = gameProperty.player.vy = 0;
1615    gameProperty.player.facingLeft = true;
1616    gameProperty.player.dead = false;
1617}
```

Figure.4 Work have been done

4. Others.

‖ Now ladders and chest will be displayed in a proper position.

‖ Now player will not unlock a chest or get a coin simply go close to it, button B is introduced and should be pressed as the interactive button.
