**Text

Description automatically generated with medium confidence**

**PRU211M PROJECT REPORT**

**C# Game Unity**

**–Hanoi, Jan 01, 2023 –**

**Table of Contents**

I. Project Report

1. Status Report

2. Team Involvements

3. Issues/Suggestions

II. Software Design Document

1. Flowchart game

2.  Detail

2.1 Map

2.2. Character

2.3. Player

2.6 Map

**I. Project Report**

**1. Team Involvements**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task** | **Member** | **Notes (Task Details, etc.)** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |

**2. Cốt truyện**

**3. Game Description**

**II. Game Design Document**

**1. Flowchart game**

**2. Detail**

**2.1 Map**

**-** Trước tiên ta có 4 loại phòng cơ bản có hình dạng giống hình dưới đây. Và thêm một phòng closed

****

**Text

Description automatically generated with medium confidence**

* Sau đó ta tìm đường đi từ điểm bắt đầu là các dãy ô hàng 1 và chọn ngẫu nhiên lấy bất kỳ 1 ô. Sau đó tìm 1 con đường ngẫu nhiên đi được từ trên xuống dưới sao cho các đường đi đều nằm trong không gian đã được định sẵn và điểm kết thúc phải nằm ở dãy cuối cùng.

**Chart

Description automatically generatedChart

Description automatically generated**

**Shape, square

Description automatically generatedA picture containing text, clock

Description automatically generated**

* Cách di chuyển đó là đầu tiên chọn ngẫu nhiên điểm bắt đầu ở hàng trên cùng và tạo ra một phòng có các lỗ hổng trái hoặc phải hoặc phía dưới ở vị trí đó. Sau đó di chuyển ngẫu nhiên sau trái phải sao cho đảm bảo phòng tiếp theo không vượt giới hạn đã định sẵn. Và tiếp tục tạo ngẫu nhiên các phòng bằng cách chọn ngẫu nhiên 1 phòng bất kỳ trong danh sách rooms(chứa các roomPrefab). Trong mỗi phòng thì có các điểm 1 điểm là ở vị trí các ô nhất định, và chứa danh sách tile hoặc chướng ngại vật hoặc quái hoặc vàng. Chúng ta sẽ lấy ngẫu nhiên các đối tượng đó và xây nên 1 căn phòng. Và cứ thế tiếp tục cho đến khi đến điểm kết thúc. Việc tạo căn phòng phải đảm bảo nó không quay trở lại hàng mà nó bắt đầu xuất phát. Nếu hướng đã chọn là xuống, đầu tiên kiểm tra xem phòng bên dưới vị trí hiện tại có lỗ hổng ở phía trên hay không. Nếu có, chúng ta sẽ di chuyển xuống dưới một bước và đặt một căn phòng ngẫu nhiên với một lỗ hổng ở phía trên ở vị trí đó. Nếu phòng bên dưới không có lỗ hổng ở phía trên, ta sẽ phá hủy phòng đó và đặt một căn phòng có cả lỗ hổng ở phía trên và dưới ở vị trí đó. Sau đó di chuyển xuống một bước và đặt một căn phòng có lỗ hổng ở phía trên ở vị trí đó. Sau đó ngẫu nhiên chọn một hướng mới để di chuyển. Sau đó chúng ta đặt các phòng kín lấp đầy các vị trí còn lại. (Các phòng kín có được không tùy thuộc vào ngẫu nhiên chọn các tile)

**A picture containing diagram

Description automatically generatedA picture containing diagram

Description automatically generated**

* Dưới đây là các phòng được thiết kế trong unity:
  + Mỗi điểm đều có các danh sách tile riêng
  + Một vài điểm đặc biết có thể sinh quái, chướng ngại vật, vàng và một vài đối tượng khác.

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

**Chart, scatter chart

Description automatically generated** **Chart, scatter chart

Description automatically generated**

- Giải quyết bài toán trên:

Giả sử mức độ được tạo ra từ 16 phòng trong lưới 4x4.

1. Khởi tạo lưới 4x4, ban đầu tất cả các ô đều có giá trị là 0
2. Chọn một ô bắt đầu ở hàng đầu tiên, tại vị trí (x, 1), với x là một số nguyên trong đoạn từ 1 đến 4
3. Thiết lập giá trị của ô (x, 1) là 1 hoặc 2
4. Với mỗi bước di chuyển, thực hiện các bước sau:

* Chọn một số ngẫu nhiên trong đoạn từ 1 đến 5, phân phối đồng nhất
* Nếu số được chọn là 1 hoặc 2, di chuyển sang phải, nếu số được chọn là 3 hoặc 4, di chuyển sang trái, nếu số được chọn là 5, di chuyển xuống
* Nếu ô đích đã nằm ngoài lưới, di chuyển xuống và đổi hướng trái / phải
* Nếu ô đích đã có giá trị khác 0, quay trở lại bước trước và chọn một số khác

1. Lặp lại bước 4 cho đến khi tìm được đường đi đến điểm kết thúc.
2. Fill tất cả các vị trí còn lại là room closed

Chú ý rằng trong trường hợp đặc biệt của phòng bắt đầu, loại phòng bắt đầu không ảnh hưởng đến phòng bắt đầu, vì vậy chúng ta có thể bỏ qua nó trong thuật toán.

**2.2 Character**

**2.3 Player**