## 1.

there are 4 functions relate to the goblins movement:

virtual void moveToPlayer(Player\* p);

virtual bool canSmellPlayer(Player\* p);

bool canSmell(char maze[MAXROW][MAXCOL],int sr, int sc, int er, int ec);

void nextMove();

first, the canSmellPlayer are called and set up a weighted map that number of steps a goblin need to take from it own position to player’s position, if the player is within not 15 steps then the canSmellPlayer function return false.

the real recursive function is canSmell. which is called by the canSmellPlayer function. it set the weighted map like this:

w w w w w w w w w w

w 8 7 6 5 6 7 8 9 w

w w 8 w 4 w w w w w

w 109 w 3 w 5 6 7 w

w 1110w 2 3 4 w 8 w

w w w w 1 w w w 9 w

w w w 1 **0** 1 2 w w w

w w w w w 2 w w 7 w

w w w w w 3 4 5 6 w

w w w w w w w w w w

say the red bold 0 is where goblin standing, the goblin can go up, down, left, right by just 1 step, so the 1 around it is the step that the goblin takes to get there. the ‘w’ means that there is a wall so the goblin cannot go through.

if the player is in 15 steps, then the goblin will call nextMove function.

the nextMove start from the position where the player is standing and look up, down, left, right, move one step back at the position where it has 1 less weight of it current position, and travel back to where the goblin is standing, then we have a path between goblin and player, so we know where goblin need to move next.

the moveToPlayer is called when moving the goblin, which calls nextMove, and if player is not within 15 steps then the function return false.

## 2.

the game start from game object, and draw the dungeon and create player, monster and game objects. the dungeon will collect all the player and monsters status like attack, sleep, or read a scroll etc. and draw them when refreshing dungeon drawing. so the dungeon will redraw itself every time user press a key.

## 3

the room in the dungeon is generated as follows:

first, divid the dungeon into 5 sections from left to right, then for each section, divid top and bottom two part. then randomly decide if i want to draw both top and bottom part or just draw one big room at that section. if there are top and bottom two part, then draw a line which connect them.

after all rooms in 5 section are drawn, then choose the middle one as destination, then make all room connect to that room.