**Documento di presentazione:**

* Confronto matrice int – matrice bool e matrice stampata
* Creazione path
* Risoluzione bug del path che attraversa stanze
* Future robe:  
  - Salvataggi  
  - Intelligenza artificiale
* Bug conosciuti:  
  - Possono esserci più porte vicine, di cui una chiusa e una aperta

**Makefile**

* Aggiungere path

**Generazione stanze**

* Collegare numero stanze e grandezza stanze (più stanze ci sono, meno saranno grandi)
* Se tra due stanze non c’è un corridoio, metterci più nemici
* Controllare se ci sono già caratteri dove si sta inserendo una nuova stanza
* Decidere randomicamente se due stanze sono collegate o no
* TUTTE le stanze devono avere un collegamento
* Per non rischiare che le stanza avessero solo gli angoli collegati, si fa un controllo su quanti muri le stanze attaccate abbiano in comune (se >2 allora vanno bene, altrimenti una va rigenerata)
* XML E VARIABILI: le stanze hanno degli id, i muri di ogni stanza saranno composti da i loro id -> in questo modo posso avere una matrice che rappresenta la mappa con caratteri scrivibili normalmente e quando dovrò stampare la mappa, al posto dei caratteri potrò mettere il chWall di game.rooms[id] (altrimenti avrei dovuto non avere la matrice rappresentante la stanza, ma soltanto scorrere tutte le stanze ogni volta che veniva stampata a schermo la mappa) -> facilitato anche il fatto che il giocatore vede un’interfaccia 81x19, complicata da stampare con quest’ultimo metodo

PROPOSTA:

prima generare le stanze e poi indicare quanto è grande “game” (l’alternativa sarebbe stata generare la grandezza di game e poi le stanza, ma avrebbe richiesto troppi controlli e modifiche alle stanze per farle entrare nella matrice)

**Matrice mappa**

* Matrice di int:  
  NUMERI POSITIVI: 0 = vuoto, 1 = player, 2 = enemy, 3 = treasure chest, da 4 a 32.767 = muro di stanze (id)  
  NUMERI NEGATIVI: da -4 a – 32.767 = floor/path/door della stanza di id = abs(numero)

**Pathfinding**

<https://www.growingwiththeweb.com/2012/06/a-pathfinding-algorithm.html>

**Elements**

* Ordinarli per ridurre complessità

**Difficoltà**

* Variare numero elementi utili
* Variare numero elementi “nemici”

**Appeal grafico**

* <https://stackoverflow.com/questions/9158150/colored-output-in-c/9158263>

**STUDIARE PER ESAME**

* **Stream**