
PROGETTO 2021/2022

IL PROGETTO SERVE PER

- ▶ Farvi sperimentare le tecniche viste a lezione su scala più grande di quanto possibile in una lezione di 3 ore, in direzione di quello che succede nel mondo reale
- ▶ I progetti reali sono spesso MOLTO più grandi
- ▶ Mettervi alla prova in uno scenario che lascia spazio alla vostra fantasia, ma pone anche alcuni vincoli

VALUTAZIONE DEL PROGETTO

- ▶ Il progetto è parte integrante dell'esame ed è **obbligatorio**
- ▶ Ha un voto massimo di 8 punti che si sommano al voto dello scritto (max 24)
- ▶ Il lavoro viene svolto in gruppi, ma la valutazione del progetto è **individuale**
 - ▶ Studenti dello stesso gruppo possono prendere voti diversi!
- ▶ Il progetto **si consegna una volta sola** ed il voto è valido per l'intero anno accademico (anche per quello successivo se le regole non cambiano)
- ▶ Chi non ottiene almeno 2 punti alla discussione orale dovrà presentare un nuovo progetto

GRUPPI

- ▶ Il progetto si svolge in gruppi di 3/4 persone, ma il voto non è di gruppo
- ▶ NON è possibile fare gruppi di 5 o più persone
- ▶ Sono ammessi **progetti individuali in casi eccezionali** (ma il carico di lavoro non diminuisce)
- ▶ Auto-organizzatevi per la creazione di gruppi
- ▶ Comunicherete i componenti del gruppo in fase di prenotazione e/o discussione del progetto

QUANDO CONSEGNARE IL PROGETTO?

- ▶ Il progetto potrà essere presentato entro il 28/02/2023
- ▶ Orientativamente ci sarà un appello ogni appello dello scritto (uno/due per sessione)
- ▶ L'esame, per il momento, si svolgerà tramite Teams
- ▶ Se ci sono molto richieste, possiamo fare ulteriori date, fuori dalle date prestabilite

CONSEGNA E DISCUSSIONE

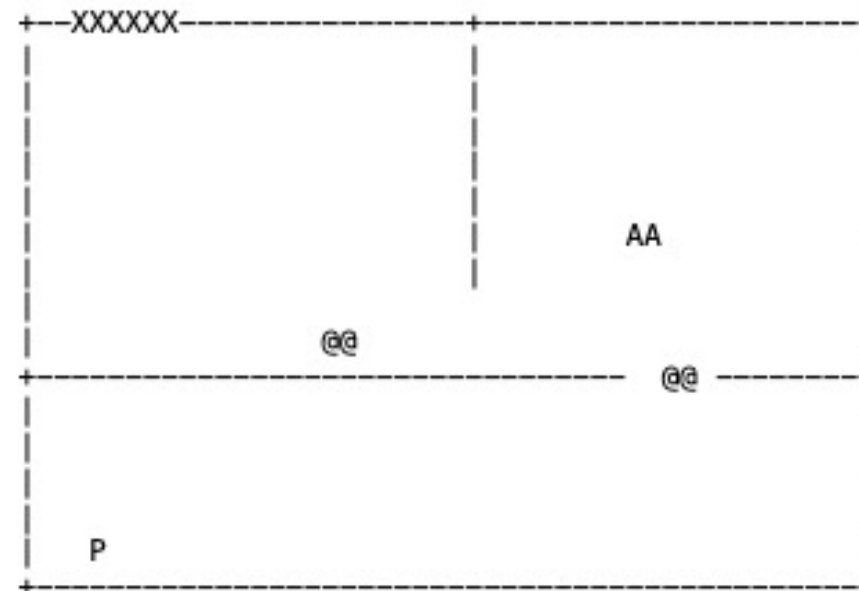
- ▶ Il progetto si consegna **una settimana** prima della discussione
- ▶ TUTTI i membri del gruppo devono partecipare alla discussione
- ▶ Il gruppo consegna:
 - ▶ Codice (sorgenti e binari)
 - ▶ File README
 - ▶ Screen recording che mostri l'esecuzione del gioco
 - ▶ Breve relazione (3/4 pagine) in cui si descrivono le principali scelte nell'implementazione del progetto e la **suddivisione del lavoro** tra i vari componenti del gruppo
- ▶ Consegna tramite mail a giuseppe.lisanti@unibo.it e adele.veschetti2@unibo.it

CONTENUTI DEL GIOCO

- ▶ Si richiede di implementare un gioco platform in grafica ASCII
- ▶ Sviluppato su più livelli
- ▶ A punteggio
- ▶ Il protagonista viene controllato da tastiera

Legenda:

- ▶ P: personaggio
- ▶ AA: artefatto
- ▶ @@: nemico
- ▶ XXXXXX: porta chiusa
- ▶ " " : porta aperta



POINTS: 30

LIFE: ♥♥♥

SVOLGIMENTO DEL GIOCO

Requisiti minimi e obbligatori:

- ▶ La mappa viene esplorata man mano dal giocatore
- ▶ Non esistono traguardi
- ▶ Il protagonista può tornare indietro nella mappa e trovare esattamente tutto come lo ha lasciato
- ▶ Devono essere presenti **diversi tipi di nemici e diversi tipi di artefatti**
- ▶ Per passare da una stanza all'altra è necessario prendere artefatti/sconfiggere nemici
- ▶ È possibile trovare "poteri" per sbloccare passaggi che portano ad artefatti

IL PUNTEGGIO

- ▶ Il giocatore umano controlla il protagonista
- ▶ Il protagonista ha un punteggio vita e cuori; a 10 punti vita corrisponde un cuore
- ▶ Nel percorso il protagonista incontra nemici, artefatti e poteri:
 - ▶ I nemici **tolgono** cuori; diversi tipi di nemici tolgono diverse quantità di cuori
 - ▶ Gli artefatti **aumentano** la vita
 - ▶ I poteri **sbloccano passaggi** che portano ad artefatti, ripristinano vita se in precedenza si è stati colpiti da un nemico e/o aprono porte delle stanze
- ▶ I nemici sono controllati dal computer: si possono muovere e sparare, possono venire uccisi dal protagonista (es. sparando)
- ▶ La mappa ha una visuale dall'alto ed è divisa in stanze; le stanze hanno più ingressi/uscite (porte)
- ▶ Quando la vita del protagonista scende a 0 il gioco termina

IMPOSTAZIONE DEL PROGETTO

Vincoli:

- ▶ La mappa deve avere una grafica ASCII, **è possibile utilizzare solo le librerie curses/ncurses.h** per gestire la grafica
- ▶ Gli elementi delle stanze devono essere gestiti tramite strutture dati dinamiche (prevedendo, quindi, inserimento e rimozione)
- ▶ Il progetto deve essere realizzato usando le classi
- ▶ Il progetto è organizzato in più file
- ▶ Ad ogni classe corrispondono due file: NomeClasse.cpp e NomeClasse.hpp

ESEMPIO DI FILE

NomeClasse.hpp

```
class NomeClasse{  
protected:  
    int field;  
    ...  
public:  
    ...  
    void method();  
    ...  
};
```

NomeClasse.cpp

```
#include "NomeClasse.hpp"  
  
void NomeClasse::method(){  
    // do something  
}
```

IN OGNI FILE IN CUI SI USA IL TIPO "NOMECLASSE" BISOGNA IMPORTARE NOMECLASSE.HPP