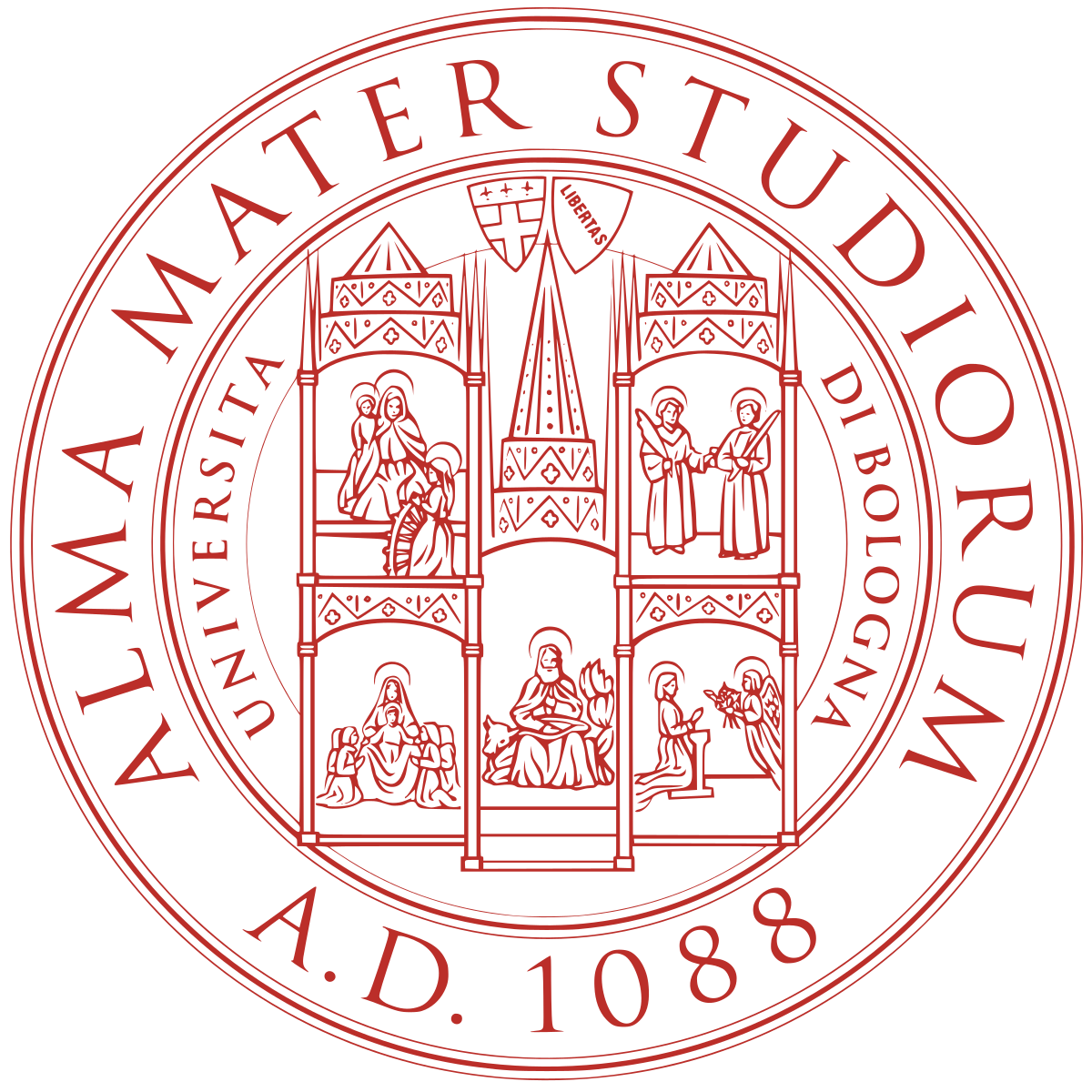
PROGETTO PROGRAMMAZIONE ANNO 2021/2022



Marco Coppola 0001020433

Gabriele Randi 001021542

(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 )

# Indice

# Inizializzazione e generazione della mappa

Il primo problema di cui ci siamo accorti durante lo sviluppo del gioco è stato che la dimensione del terminale non è sempre abbastanza grande da permettere alla mappa (e i vari elementi come mini-mappa o statistiche del gioco) di rientrare nei bordi.

Per questo, una volta fissato le dimensioni della mappa, viene invocata una funzione che permette all’utente di ridimensionare il terminale affinché non raggiunga una dimensione accettabile.

Immagine che contiene testo, verde

Descrizione generata automaticamente

Subito dopo parte la generazione della mappa e delle stanze, ognuna associata ad un file. json nel quale sono descritti gli artefatti e le entità da creare e in che posizione.

DESCRIZIONE FUNZIONE GENERAZIONE MAPPA

# Il motore di gioco

Dopo aver configurato tutto e dopo aver generato stanze ed entità, il gioco vero e proprio è pronto per partire. Una delle sfide più grandi a livello implementativo è stato decidere come permettere alle varie entità di muoversi ed effettuare azioni tutte allo stesso tempo, per evitare in ogni frame di avere entità che si muovono prima delle altre oppure ancora peggio, non permettere al giocatore di evitare un proiettile in arrivo in quanto la sua posizione non è ancora stata aggiornata.

Abbiamo deciso allora di prendere spunto dal funzionamento di molte librerie grafiche, che consistono di inserire una classe astratta (nel nostro caso GameObject) da cui tutti gli oggetti che hanno bisogno di essere aggiornati ereditano e che consiste di due metodi fondamentali:

* DoFrame:
* Draw:

Abbiamo infine arricchito la classe GameObject con delle proprietà condivise da tutti gli oggetti di gioco, ovvero la posizione all’interno della mappa, il carattere identificativo dell’oggetto che comparirà sulla mappa una volta chiamato Draw() e la lista di tutti gli oggetti attualmente nella stanza di gioco per un accesso immediato (ad esempio calcolare la distanza di un nemico dal giocatore, determinare le varie collisioni ecc.)

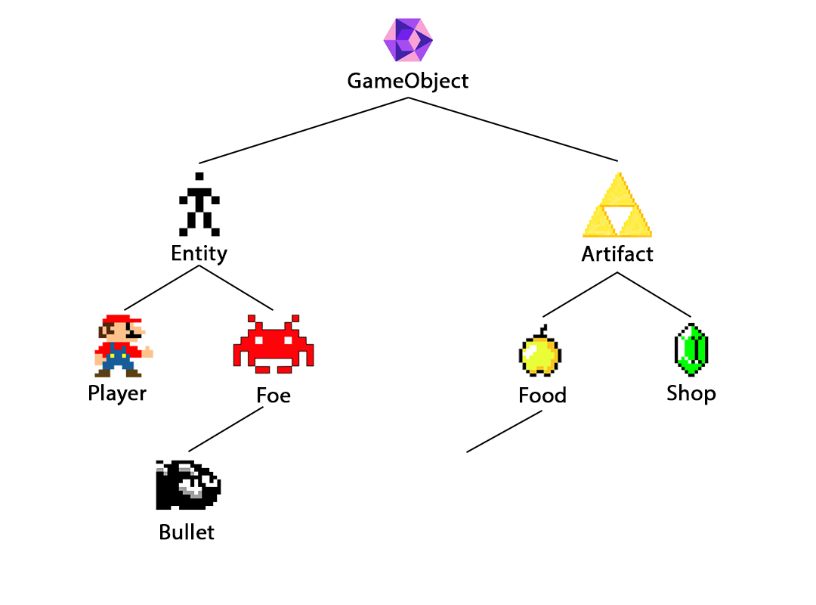
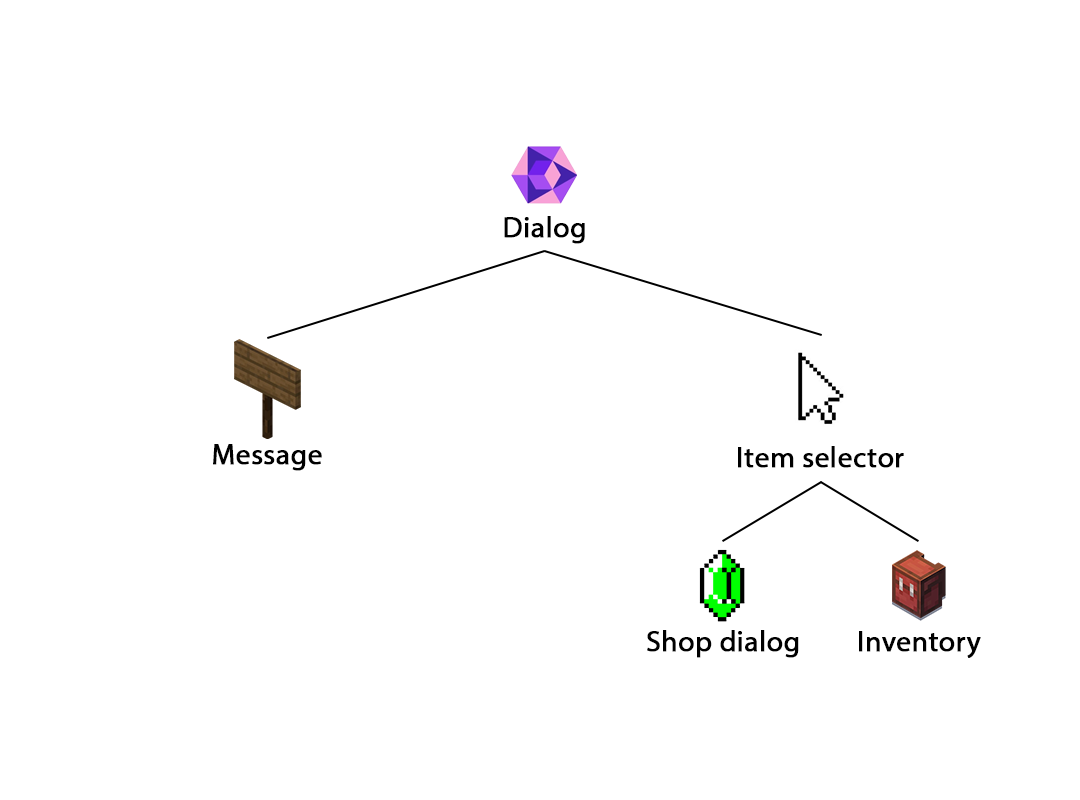
# Immagine che contiene testo Descrizione generata automaticamente

# Lo spostamento attraverso le varie stanze

# Ereditarietà

Il concetto di **ereditarietà** svolge un ruolo fondamentale all’interno del progetto, applicandosi alla perfezione per la **molteplicità** degli oggetti di gioco che, pur condividendo la maggior parte delle variabili e delle funzioni basilari, si distinguono ognuno per qualche particolarità che viene implementata in modo diverso.

I due ambiti principali in cui è stata applicata l’ereditarietà sono stati i vari **oggetti di gioco** (il giocatore stesso, i vari artefatti e i diversi tipi di nemici) e i **dialoghi** (che includono messaggi, inventario e il negozio di oggetti)



# Gli oggetti di gioco

Spiega qui le implementazioni con strutture dati dinamiche, inserimento rimozione ecc.

# Conclusione