Progetto di Programmazione

Davide Foresti : 0001020882
Elia Friberg: 0001030735
Matteo Raggi : 0001021057
Mattia Lodi : 0001020617

febbraio 2023

1 Suddivisione dei compiti

Davide:

- Menu principale
- Menu di pausa
- Statistiche del giocatore

File: Menu, Menu_playing

Elia:

- infrastruttura
- Nemici
- Sistema di combattimento
- Debugging

File: Boom, Character, Chaser, Coward, Drunk, Enemy, Flyer, Projectile, Shooter, Stalker, Time, Turret

Matteo:

- Artefatti
- Hero
- Sistema di combattimento

File: Artifact, Hero

Mattia:

- Creazione e gestione delle stanze
- Strutture dati per la loro memorizzazione
- Grafica

File: Door, GeneralTemplate, Room, Templates, Wall

2 Scelte implementative

2.1 Il Menu

La classe Menu crea l'interfaccia con cui va a interagire l'utente all'avvio. Menu utilizza dei setter per salvare le impostazioni scelte dall'utente in variabili protette. Quando il menu si chiude si utilizzano dei getter per passare le impostazioni selezionate alla classe Game.

2.2 Il Gioco

La classe Game si occupa di gestire tutto ciò che riguarda il gioco. Game comprende 3 finestre (per la grafica delle stanze, delle statistiche del personaggio e per il punteggio), il personaggio (Hero), l'indice delle stanze e un puntatore alla stanza corrente.

2.3 Le Stanze

Ogni stanza contiene un puntatore alle stanze con cui confina che sono già state esplorate, in modo da potersi muovere in una di queste senza visitare l'indice; delle coordinate x e y per identificarla univocamente all'interno della mappa, dei flag per gestire lo stato di ogni porta (aperta/chiusa presente/non presente), e un template. Il template contiene i muri, le porte, i nemici e gli artefatti.
Le stanze sono memorizzate in una struttura dati dinamica di tipo vettore, chiamata indice, in modo da potere essere aggiunte gradualmente esplorando la mappa; l'indice viene usato solo quando si crea una nuova stanza per avere delle porte coerenti con il resto della mappa. In questo modo il cambio di stanza quando si torna in una stanza già visitata è molto rapido, ma è lento quando se ne visita una nuova.

2.4 I Template

Per il contenuto delle stanze si usa una classe che memorizza il contenuto della stanza, detta template, ce ne sono 40, di diverse rarità e ogni volta che si crea una nuova stanza, gliene ne viene asseganto uno.
Per potere essere interscambiabili i template sono tutti sottoclassi di GeneralTemplate e sono memorizzati all'interno della stanza come puntatori.

I muri, le porte, i nemici e gli artefatti sono tutti memorizzati in array dinamici in modo da potere variare in numero e contenuto per ogni stanza, anche tra quelle con lo stesso template.
Il file GeneralTemplate.cpp contiene delle funzioni per aggiungere alle stanze, insiemi di muri o porte in modo che abbiano una certa forma; per renderne più pulita l'inizializzazione.

2.5 Il Personaggio

La classe Hero controlla tutte le caratteristiche dell'Eroe: vita, danno, chiavi (per aprire le porte), tempi di ricarica, portata e abilità. Le caratteristiche dell'eroe prendono valori diversi a seconda della classe che viene scelta prima di iniziare a giocare. Diverse classi possono avere quindi più o meno vita, portata, danno e così via. La funzione "useAbility" attiva abilità differenti a seconda della classe selezionata. Le abilità possono modificare temporaneamente le caratteristiche dell'eroe o ripristinargli la vita. La funziona "attack" crea un nuovo proiettile nella direzione voluta. "CenterHero" serve a posizionare il personaggio al centro della stanza quando inizia il gioco. Tutte le restanti funzioni modificano le variabili del personaggio. Per esempio "useKey" consuma la chiave del personaggio e apre una porta, "increaseHealth" aumenta la vita.

2.6 I Nemici

I nemici sono sottoclassi di Character come l'eroe, ma anche di Enemy, che include due funzioni virtuali, una per la decisione della direzione da prendere, e una per il controllo dei proiettili che ogni nemico ha. Sono stati implementati diversi nemici:
Drunk che si muove a random e non spara,
Una turret che sta ferma e spara di continuo,
Coward che scappa e lancia bombe dietro di se, anche se limitate, coward include anche un sistema avanzato per trovare aperture e scappare senza fermarsi in un angolo, e può lasciare artefatti quando muore,
Chaser che ti insegue per colpirti da vicino
Flyer che si comporta come Chaser ma può volare sopra i muri e le porte chiuse
Shooter che ti spara da lontano,
Stalker che diventa invisibile mentre non ti spara,
Boom che viene vicino a te ed esplode a kamikaze.

Ogni nemico si attiva solo se è ad una certa distanza ed ha line of sight con il player, ma ha una corta memoria anche se viene persa per periodi brevi, dopo di che torna idle, stando fermo o si muove randomicamente. Dopo aver sparato ogni nemico ha bisogno di un tempo di reload, in cui è vulnerabile, mentre Boom ha un fuse timer prima di esplodere. Le statistiche di ogni nemico(hp, range dei nemici ranged, e dmg) dipendono dalla difficoltà, che scala direttamente con lo score ottenuto eliminando nemici. Il proiettile dipende dal nemico che l'ha sparato,con caratteristiche diverse per nemico. Ogni nemico può essere promosso a Boss, con statistiche maggiorate e che lasciano artefatti quando muoiono. I nemici possono essere spawnati con tipo scelto o random, con coordinate scelte o random, o qualsiasi combinazione di questi, Drunk e Turret sono gli unici due che a random non possono essere scelti.

2.7 Gli artefatti

Gli Artefatti sono creati come semplici Drawable. Sono quindi dei caratteri presenti nelle stanze di gioco, con cui possiamo interagire. Essi vengono creati insieme al Template della stanza. A differenza dei nemici che vengono creati casualmente, questi vengono posizionati in punti precisi per non permettere la generazione in luoghi troppo facili da raggiungere. Esistono quattro tipi di artefatti: Range, Danno, Vita e Chiave. Ogni artefatto aumenta la statistica associata e cura il giocatore. Possono rendere il gioco più facile con l'aumentare dello score, ma è bilanciato dall'aumento progressivo della difficoltà.

2.8 Il sistema di combattimento

Il sistema di combattimento regola danno inflitto e danno subito da nemici ed eroe diminuendo gli Hp del Character colpito, che arrivati a 0 lo uccidono. L'intero sistema è costruito su Projectile, che ogni tick che rimane sullo schermo aumenta il proprio Uptime. Una volta che l'Uptime supera il range associato al proprio nemico o dopo aver colpito qualcosa il proiettile scompare.