Relazione
“SMOL”

Ettore Farinelli
Marco Galeri
Giovanni Paradisi
Mounir Samite

2 aprile 2023

Indice

Capitolo 1

Analisi

1.1 Requisiti

Il gruppo si pone come obiettivo quello di realizzare un videogioco 2D arcade con visuale dall'alto di nome "SMOL". Per arcade si intende una struttura di gioco ripetitiva in cui l'obbiettivo è accumulare più punti possibile

Requisiti funzionali

Requisiti non funzionali

1.2 Analisi e modello del dominio

Capitolo 2

Design

2.1 Architettura

L'architettura di GLaDOS segue il pattern architetturale MVC. Più nello specifico, a livello architetturale, si è scelto di utilizzare MVC in forma "ECB", ossia "entity-control-boundary"[1]. GLaDOS implementa l'interfaccia AI, ed è il controller del sistema. Essendo una intelligenza artificiale, è una classe attiva. GLaDOS accetta la registrazione di Input ed Output, che fanno parte della "view" di MVC, e sono il "boundary" di ECB. Gli Input rappresentano delle nuove informazioni che vengono fornite all'IA, ad esempio delle modifiche nel valore di un sensore, oppure un comando da parte dell'operatore. Questi input infatti forniscono eventi. Ottenere un evento è un'operazione bloccante: chi la esegue resta in attesa di un effettivo evento. Di fatto, quindi, GLaDOS si configura come entità *reattiva*. Ogni volta che c'è un cambio alla situazione del soggetto, GLaDOS notifica i suoi Output, informandoli su quale sia la situazione corrente. Conseguentemente, GLaDOS è un "observable" per Output.

Figura 2.1: Schema UML architetturale di GLaDOS. L'interfaccia `GLaDOS` è il controller del sistema, mentre `Input` ed `Output` sono le interfacce che mappano la view (o, più correttamente in questo specifico esempio, il boundary). Un'eventuale interfaccia grafica interattiva dovrà implementarle entrambe.

Con questa architettura, possono essere aggiunti un numero arbitrario di input ed output all'intelligenza artificiale. Ovviamente, mentre l'aggiunta di output è semplice e non richiede alcuna modifica all'IA, la presenza di nuovi

[1]Si fa presente che il pattern ECB effettivamente esiste in letteratura come "istanza" di MVC, e chi volesse può utilizzarlo come reificazione di MVC.

tipi di evento richiede invece in potenza aggiunte o rifiniture a GLaDOS. Questo è dovuto al fatto che nuovi Input rappresentano di fatto nuovi elementi della business logic, la cui alterazione od espansione inevitabilmente impatta il controller del progetto.

In Figura 2.1 è esemplificato il diagramma UML architetturale.

2.2 Design dettagliato

Ettore Farinelli

roba di Ettore

Marco Galeri

roba di Marco

Giovanni Paradisi

roba di Giovanni

Mounir Samite

roba di Mounir

Capitolo 3

Sviluppo

3.1 Testing automatizzato

3.2 Metodologia di lavoro

un po di roba sul DVCS

Ettore Farinelli

roba di Ettore

Marco Galeri

roba di Marco

Giovanni Paradisi

roba di Giovanni

Mounir Samite

Mi sono occupato di:

- Progettazione dell'utilizzo delle finestre con Window e WindowState con cui verranno renderizzate tutti i tipi di finestre come Menu, Gioco e Game Over (**package it.unibo.smol.view**)
- Implementazione del Menu di gioco nella classe MenuState (**package it.unibo.smol.view**)

- Three

3.3 Note di sviluppo

Ettore Farinelli

roba di Ettore

Marco Galeri

roba di Marco

Giovanni Paradisi

roba di Giovanni

Mounir Samite

roba di Mounir

Capitolo 4

Commenti finali

un po de roba

4.1 Autovalutazione e lavori futuri

Ettore Farinelli

roba di Ettore

Marco Galeri

roba di Marco

Giovanni Paradisi

roba di Giovanni

Mounir Samite

roba di Mounir

4.2 Difficoltà incontrate e commenti per i docenti

opzionale

Appendice A

Guida utente

Bibliografia