PizzaTime Project Report

Paradigmi di Programmazione e Sviluppo

Lorenzo Chiana
Meshua Galassi
Giada Gibertoni
Giacomo Pasini

Anni accademici 2018–2019 e 2020–2021

Indice

Capitolo 1

Processo di sviluppo

Nel seguente capitolo si discute delle metodologie di sviluppo adottate dal team e delle scelte tecnologiche fatte per il compimento del progetto.

1.1 Metodologia di sviluppo

1.2 Strumenti utilizzati

1.2.1 Versioning

Il sistema di *versioning* ha ricorperto un ruolo centrale nella realizzazione del progetto e per tale scopo abbiamo scelto di utilizzare **Git**. Per quanto riguarda l'hosting del repository abbiamo optato per la piattaforma **GitHub**.

Il flusso di lavoro è stato definito secondo le linee guida del metodo di branching **git-flow**:

- Il branch *master* contiene il codice relativo a ciascuna release.
- Il branch *development* ospita il codice realizzato durante uno sprint. Tale codice è testato e stabile, ma non ancora giudicato completamente utilizzabile.
- I vari branch *task-** corrispondono a ciascun task individuato durante i vari sprint. Tale codice è in fase di implementazione e può contenere codice non completamente funzionante, in quanto ancora in sviluppo.
- non sono stati necessari branch di *hotfix*.

1.2.2 Dependency management & buildscript

Per la gestione delle dipendenze, quali ad esempio librerie o plugin, e la compilazione del codice, come strumento di automazione dello sviluppo si è utilizzato **sbt**. In particolare, sono stati scritti buildscript per automatizzare:

- la gestione delle *dipendenze* provenienti da repository Maven;
- il processo di *testing* tramite ScalaTest e controllo di *qualità del codice* tramite Scalastyle;

- la gestione delle dipendenze dei vari moduli di *JavaFX* per le varie versione di Java.
- generazione dei *Jar* eseguibili.

1.2.3 Continuous Integration

Per la verifica del codice prodotto e dei reativi test si è optato per *Travis CI*. Questo servizio offre la possibilità di registrare un web-hook al repository GitHub su cui è istanziato il progetto, così da provocare l'esecuzione di Travis CI ad ogni commit effettuato sul repository. La configurazione di Travis è stata definita utilizzando il formato YAML e in particolare sono stati specificati:

- il linguaggio (Scala);
- la versione di Scala;
- la JDK da utilizzare.

Capitolo 2

Analisi del dominio

2.1 Descrizione del gioco

Capitolo 3

Requisiti

3.1 Requisiti di business

3.2 Requisiti utente

3.3 Requisiti funzionali

3.4 Requisiti non funzionali

3.5 Requisiti implementativi

Capitolo 4

Design architetturale

4.1 Architettura complessiva

4.1.1 Model

4.1.2 View

4.1.3 Controller

Capitolo 5

Design di dettaglio

5.1 Organizzazione del codice

5.2 Pattern utilizzati

5.3 Scelte rilevanti

Capitolo 6

Implementazione

6.1 Lorenzo Chiana

6.2 Meshua Galassi

6.3 Giada Gibertoni

6.4 Giacomo Pasini

Capitolo 7

Retrospettiva e commenti finali