

Il Domain Specific Language (DSL) di Minecraft, noto come mfunction, consente lo sviluppo di pack: moduli di contenuti progettati per alterare o aggiungere meccaniche di gioco. Nonostante il suo ampio utilizzo, questo linguaggio presenta notevoli limitazioni strutturali e sintattiche: ogni funzione deve essere definita in un file separato e non dispone di costrutti quali variabili, istruzioni condizionali e meccanismi di iterazione. Questi vincoli producono codice prolioso e ripetitivo, compromettendo la leggibilità e la manutenibilità dei progetti di ampia scala.

Per superare tali problemi, questa tesi propone una libreria Java sviluppata durante il tirocinio accademico che, a partire da un'analisi approfondita delle limitazioni e difetti di mfunction, giunge alla formulazione di un'astrazione che rappresenta la struttura di un pack come un albero di oggetti tipizzati. Sfruttando costrutti di Java e factory methods, la libreria consente la generazione programmatica dei pack, offrendo zucchero sintattico e metodi di utilità che semplificano l'accesso ai file di risorse principali. L'approccio proposto sfrutta la programmazione ad oggetti per fornire validazione statica, supporta la definizione di più risorse all'interno di un singolo file sorgente e automatizza la generazione di boilerplate, eliminando così la necessità di preprocessori o script esterni.

Un working example conferma l'approccio scelto: nel pack di esempio il codice scritto è ridotto del 40%, consolidando 31 file in 3 file sorgenti, con miglioramenti significativi in termini di densità del codice e manutenibilità del progetto.