Homework N.1 - Apache Jena, RDF, OWL e SPARQL

Scopo del progetto

L’homework si basa sull’ontologia che vede come dominio parte del videogioco FPS/Action RPG *Destiny 2* sviluppato da *Bungie* e pubblicato da *Activision*.

Scopo del progetto è quello di inferire proprietà di una cerchia di conoscenti, generata coerentemente a *Foaf*, legate al dominio dell’ontologia, utilizzando come fonte di conoscenza dataset remoti di Linked Open Data (*DBPedia* e *DestinyPedia*).

Legenda

Nei seguenti paragrafi i nomi in ***grassetto*** rappresentano i file contenenti informazione, .owl .rdf .xlsx ecc.

I nomi in ***violetto*** invece rappresentano i file legati alla parte di programmazione o di modifica dei file contenenti informazione.

Per distinguere i file di informazione, ogni volta che vengono eseguite operazioni di modifica su di essi, si è optato per salvarli con un nome diverso legato al tema della modifica effettuata.

Prerequisiti

L’ontologia da cui parte l’homework è “***destiny2v2\_5.owl”*** risultato di varie versioni le quali sono state generate tramite il software dedicato alla creazione di ontologie *Protegè*. In tale ontologia sono definite le classi, sottoclassi e principali proprietà dei personaggi giocabili e non all’interno del videogioco.

A priori invece è importante la presenza del dataset di conoscenze coerente a foaf generato con il framework di Java Jena in ***“friendDestiny.java”*** in cui sono elencati cinque conoscenti con i relativi nickname in vista del collegamento con l’ontologia. Una volta esportato in JSON-LD, il modello ***“friends\_jsonjena.json”*** verrà utilizzato in seguito.

Siccome *DestinyPedia* non dispone delle proprietà semantiche necessarie per l’estensione dell’ontologia, si è supposto che in caso di presenza di tali proprietà, fosse possibile trarre informazione equivalente a quella presente in “***Notable\_Guardians.xlsx”***dalle infobox delle varie risorse.

Questo file dovrebbe rappresentare in forma schematica dei personaggi non giocabili di rilievo divisi per classe (Hunter: H, Titan: T, Warlock: W) ognuno dei quali difende un pianeta da un tipo di nemico.

Grazie a questa schematizzazione e alla *Transformation Rule* ***“Notable\_Guardians\_TR.json”*** è stato possibile importare gli individui nell’ontologia mediante il plug-in di Protegè *Cellfie.*

La creazione degli individui che si riferiscono ai personaggi giocati è stata eseguita in modo analogo con i file ***“FriendCharacters.xlsx”*** e ***“Friend\_TR.json”***.

Viste e riscontrate delle difficoltà nel far funzionare un reasoner OWL 2 su Jena in modo da inferire le proprietà inverse e l’appartenenza alle superclassi direttamente in Eclipse e vista l’uguaglianza del risultato, si è risolto il problema partendo da un modello già ragionato esportato da Protegè mediante l’opzione di esportazione di assiomi inferiti come ontologia. Questa ontologia con assiomi inferiti corrisponde al file ***“destiny”*** senza estensione che è quello da cui parte la catena di operazioni su Jena.

Computazione e inferenza

Un primo passo di connessione verso un dataset di Linked Open Data viene effettuato in ***“reasonerDestiny.java”*** in cui mediante ARQ, query engine per Jena, viene interrogato DBPedia con query SPARQL per collegare gli individui dell’ontologia di tipo “Planet”, ” Moon” e “Asteroid” alle relative risorse online tramite proprietà *owl:sameAs*. Esempi sono gli individui Earth (Planet), Mars (Planet), Io (Moon) e 7066 Nessus (Asteroid). Il risultato è salvato in ***“destiny\_extended.rdf”***

Il passo successivo è quello di connessione tra il dataset di conoscenti e i relativi personaggi giocati, individui dell’ontologia. Il collegamento avviene tramite matching del nickname della persona in ***“friends\_jsonjena.json”*** e il nome dell’individuo in ***“destiny\_extended.rdf”***. A testimonianza del collegamento è possibile vedere che gli individui nella nuova ontologia estesa ***“destiny\_extended\_friends.rdf”*** hanno una proprietà isPlayedBy definita nel *namespace* dell’ontologia che come oggetto punta all’ IRI del giocatore. Viceversa, le persone nella nuova versione delle conoscenze chiamata ***“friends\_jena\_extended.rdf”*** hanno una proprietà plays, sempre definita nel namespace dell’ontologia, che punta all’ IRI del personaggio.

Come ultimo passo in ***“notable\_mission.java”*** agli individui corrispondenti ai personaggi giocati dal dataset di conoscenti viene aggiunto un pianeta da difendere e un nemico da combattere sulla base delle regole importate da “***Notable\_Guardians.xlsx”.*** Con delle query SPARQL vengono selezionati il luogo e il nemico assegnati al *notable\_guardian* che ha la stessa combinazione di razza e sottoclasse del giocatore.

**Nota:** la sottoclasse assegnata ad un individuo (*ArcstriderSubclass, GunslingerSubclass, VoidwalkerSubclass ecc*…), nel caso dei *notable­\_guardians* è assegnata direttamente dalla regola ***“Notable\_Guardians\_TR.json”*** mentre nel caso dei personaggi giocati dai conoscenti è inferita a seconda del potere specificato in ***“FriendCharacters.xlsx”*** al fine di poter sperimentare e testare il reasoner utilizzato (Hermit).