'The Elder Scrolls V: Skyrim' Skill Tree ontology

Coman Nicolae, Zavaczki Peter

Department of Computer Science
Technical University of Cluj-Napoca

ncoman32@yahoo.com
peter.zavaczki@gmail.com

May 22, 2019

Introduction

For our Knowledge Based Systems laboratory activity we created an ontology that models the skill tree from *The Elder Scrolls V: Skyrim* game.

In our ontology we tried to capture the data found in the skill tree as clearly and as completely as possible. Additionally, we introduced some builds a player of the game might want to go with, in which case we created a connection between the build and the skill classes that are most suitable for the build the player chose to play the role of.

Competency questions

The ontology should answer to the following competency questions:

- What are the classes of characters I can play?
- What are the skills suitable for build X?
- Should I invest in skill tree X if my character is build Y?
- What skills can I unlock at level X?
- What skill is required for unlocking skill X?
- What level is required for unlocking skill X?
- What are the perks provided by skill X?

Related ontologies

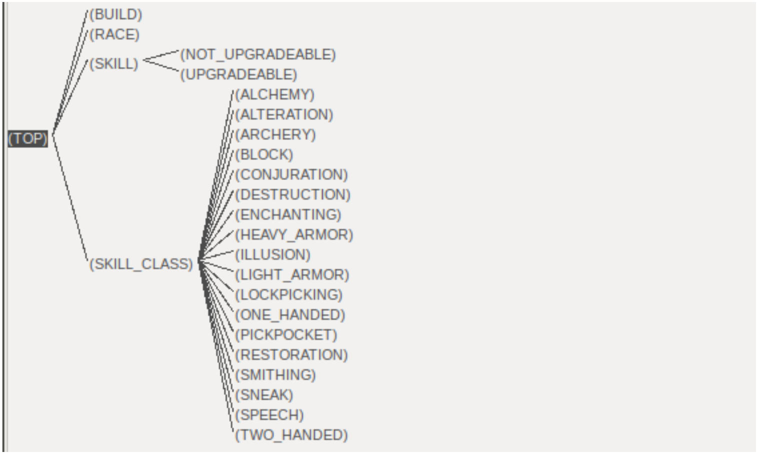
The ontologies we found were related to ours based on the fact that they all tackle the topic of video games.

- Dota 2 ontology - An ontology describing a scenario from the game - `https://ontohub.org/repositories/dota-2-ontology`
- Core Game Ontology - An ontology classifying games by their properties - `http://autosemanticgame.institutedigitalgames.com/ontologies/core-game-ontology/`
- Dota 2 item ontology - An ontology about the items and builds in Dota 2 - `https://ontohub.org/boc2018/Dota%202%20Item%20ontology`

Unfortunately none of these ontologies were useful to us, as we tackle a very specific topic. None of them are used.

Tbox

Our main concepts are Skill_class, Skill, Build and Race. The figure below shows the concepts in a more detailed way.



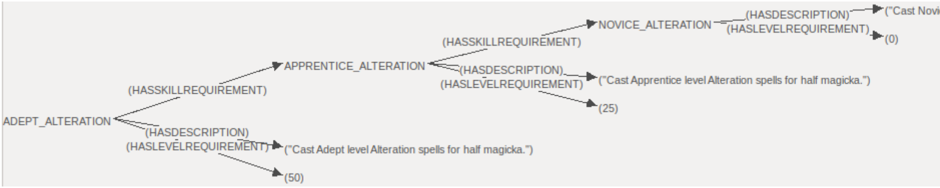
Tbox

The Skill_class is split into the existing 12 disjoint classes: *Archery, Block, Heavy Armor, One-handed, Smithing, Two-handed, Alteration, Conjuration, Destruction, Enchanting, Illusion, Restoration, Alchemy, Light Armor, Lockpicking, Pickpocket, Sneak* and *Speech*. These skill classes contain skills. Builds are paired with skill classes to tell what skill class is suitable for what build.

Skills can be either *Upgradeable* or *Not_upgradeable*, these two traits are obviously disjoint between eachother. Above this, we need to model the pre-required skill, for which we use a role called *hasSkillRequirement*. For the level requirement and perk description we use attributes.

Abox

Our Abox is mainly composed of instances of skills, then some races and some builds. Instances of races and builds are simple, since they are then used to determine which skill class to invest in.
This can be seen in the tree structure in the images below. The skill is only the root, but since the skill requirement takes another skill as parameter, it extends until it reaches a "leaf skill".



Queries

The last thing we do as part of loading the ontology and before we run our queries is running all the rules with the *run-all-rules* command. After this step, we can run our evaluation queries. We check the consistency, the size, the expressivity of our ontology, and finally, we answer the competency question with some custom queries.
For more details and for the full code, please check the full documentation document.

The End