



Knowledge Representation and Reasoning Project Report

Genshin Impact Knowledge Graph

Course Instructor: **Dr. Amna Basharat**

Team Members

Sarim Aeyzaz (i21-0328)

Raffay Khan (i21-0335)

Zaraar Malik (i21-2705)

Table of Contents

- 1) Domain Description
- 2) Competency Questions
- 3) Dataset Description
 - a) Description of each Table
- 4) Conceptual Model
 - a) Identified Vocabularies
- 5) External Linked Datasets
- 6) Designing the Ontology
 - a) Ontology Class Hierarchy
 - b) Ontology 11 Requirements
- 7) Mapping
- 8) Linking
- 9) Reasoning
 - a) SWRL Rules
 - b) Normal Reasoning
 - c) SWRL Reasoning
 - d) Linked Data Reasoning
- 10) Publishing
- 11) Validation using SPARQL Endpoint
 - a) Competency Questions Answered
- 12) Visualization
- 13) Reflect

Domain Description

Our KRR Knowledge Base will be about Genshin Impact. It's a game where you have 7 elements (Pyro, Cryo, Hydro, Electro, Dendro, Anemo & Geo) and each character is aligned to one of those elements only. Using the elements, the character is able to use their abilities and defeat enemies/bosses found throughout the world.

Each character can equip a specific type of weapon and certain characters are strong against certain enemies depending on the element of the character and the element of the respective enemy. In terms of progression, each character needs specific materials to upgrade themselves and their weapon.

As you play the game, you are able to control 4 characters at the same time. These characters can be of any element and of any role (Damage Dealer, Shielder, Healer, Support). When you are using multiple characters to defeat enemies, different elements are able to react with one another causing a reaction (Electro-charged, Vaporize, Melt, Crystalize, Freeze, Swirl, Spread, Aggravate). These reactions amplify the damage you deal to the enemy so you are incentivized to run specific team combinations to maximize a reaction type.

In terms of explorations, there are tons of chests and enemies you encounter along the way. These enemies can be of different tiers (Normal, Elite, Boss and Weekly Boss) and as such, the player has to defeat them in order to progress with the story.

Competency Questions

Here are a few competency questions we expected our knowledge graph to answer:

- What materials do I need to level up X character? (e.g. Kaeya)
- What materials do I need to upgrade X weapon? (e.g. Aquila Favonia)
- List me all characters of X element. (e.g. Hydro)
- Which element is involved in the most reactions?
- Which element has the most healers?
- List me all the weapons X character can use? (e.g. Kazuha)
- What characters are strong against X boss? (e.g. Pyro Regisvine)
- Create a team of 2 people which can do a vaporize reaction?
- List me an electro-charged only team with all 4 characters having a different role
- (Bonus Federated query) Get me the voice actor names for the characters from Wikidata

Dataset Description

Our data is unstructured as it is represented in the form of a game. Using this information, We extracted majority of the game's information and then gave it a formal representation via CSV files. Here is the final iteration of the CSV files and the attributes of each one:

- Characters
- Weapons
- Enemies
- Reactions

Note: A lot of this data was extracted from Project Amber website (<https://ambr.top/en>)

Description of each Table

- **Characters:**
 - character_name (Name of the character)
 - rarity (Rarity of the character, either 4-star or 5-star)
 - region (Location where the character resides at)
 - vision (One of the 7 elements possessed by the character)
 - weapon_type (One of the 5 weapon types used by the character)
 - constellation (Unique per character)
 - birthday (Birthday of the character)
 - ascension_boss (Material used by the character to level up)
 - limited (Boolean value representing if the character is limited or not)
 - ascension (Ascension passive granted when you level the character up)
 - role (Role of the character. Can be DPS, Healer, Shielder or Support)
 - gender (Male or Female)
 - archon (Boolean value representing if a character is an archon)
 - release_date (Release date of the character in the game)

	character_name	rarity	region	vision	weapon_type	constellation	birthday	ascension_boss	limited	ascension	role	gender	archon
0	Albedo	5	Mondstadt	Geo	Sword	Princeps_Cretaceus	2023-09-13	Basalt_Pillar	True	Geo_DMG_Bonus	Support	Male	False
1	Alhaitham	5	Sumeru	Dendro	Sword	Vultur_Volans	2023-02-11	Pseudo-Stamens	True	Dendro_DMG_Bonus	DPS	Male	False
2	Aloy	5	Mondstadt	Cryo	Bow	Nora_Fortis	2023-04-04	Crystalline_Bloom	True	Cryo_DMG_Bonus	Support	Female	False
3	Amber	4	Mondstadt	Pyro	Bow	Lepus	2023-08-10	Everflame_Seed	False	ATK	Support	Female	False
4	Arataki_Itto	5	Inazuma	Geo	Claymore	Taurus_Iracundus	2023-06-01	Riftborn_Regalia	True	CRIT_Rate	DPS	Male	False

- **Weapons:**
- Name (Name of the weapon)
- Type (One of the 5 types of the weapon)
- Rarity (Can be either 1-Star till 5-Star)
- UpgradeMaterial (Material used to upgrade the weapon)

	Name	Type	Rarity	UpgradeMaterial
0	Wolf's_Gravestone	Claymore	5	Tusk_of_the_Wolf
1	Skyward_Harp	Bow	5	Luminous_Sands_from_Guyun
2	Primordial_Jade_Winged-Spear	Polearm	5	Jade_Wing
3	Aquila_Favonia	Sword	5	Tusk_of_the_Wolf
4	Lost_Prayer_to_the_Sacred_Winds	Catalyst	5	Tile_of_Decarabian's_Tower

- **Reactions:**
- Element (Element involved in a reaction)
- causesReaction (Reaction caused by that element)

	Element	causesReaction
0	Pyro	Overloaded
1	Electro	Overloaded
2	Pyro	Melt
3	Cryo	Melt
4	Pyro	Vaporize

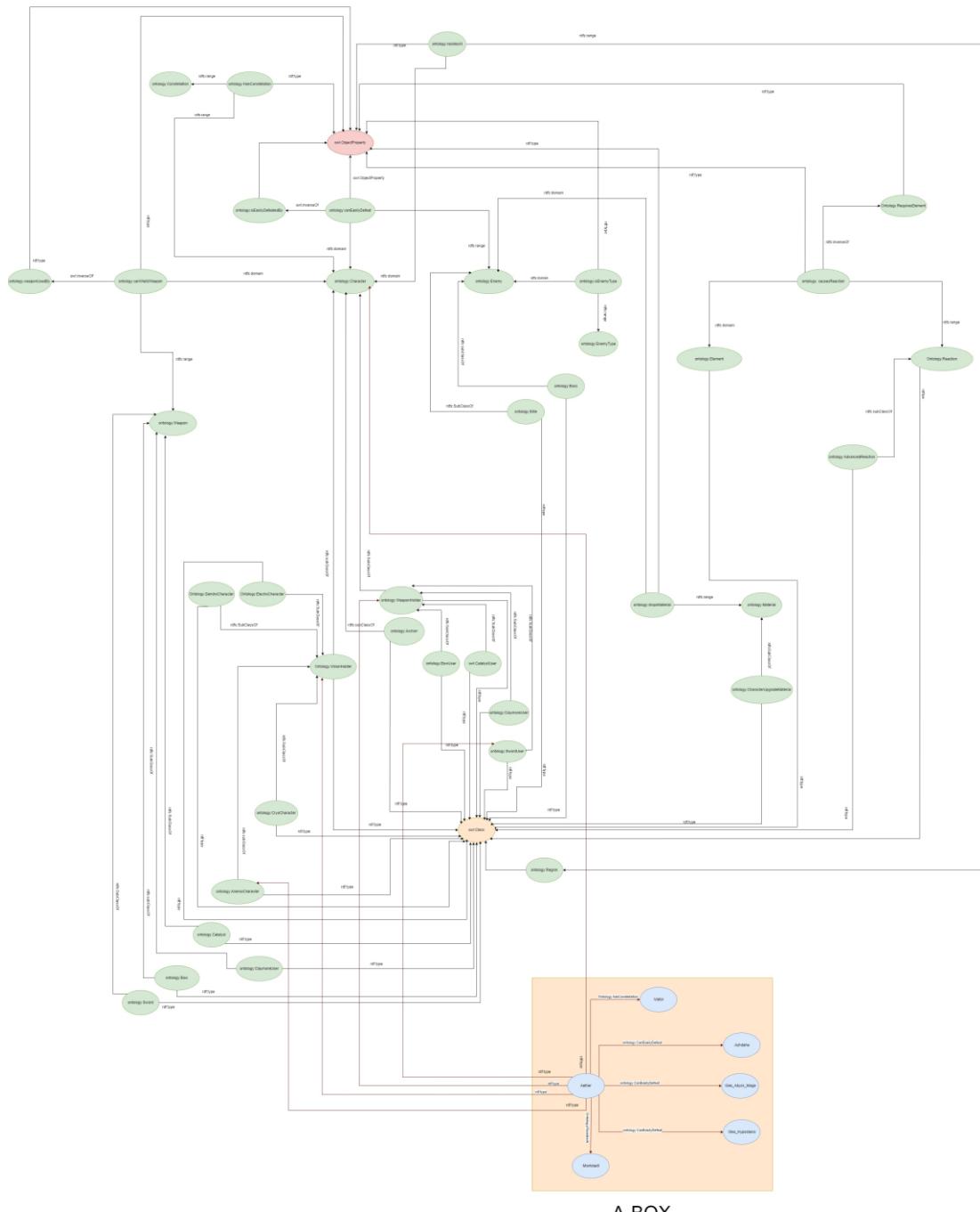
- **Enemies:**
- Name (Name of the enemy)
- Element (Element possessed by the enemy)
- WeakAgainst (Element the enemy is weak against)
- EnemyType (One of the 4 different types an enemy can be)
- DropsMaterial (Material dropped by the enemy)

	Name	Element	WeakAgainst	EnemyType	DropsMaterial
0	Hilichurl	Anemo	Electro	Normal	Firm_Arrowhead
1	Anemo_Samachurl	Anemo	Electro	Elite	Divining_Scroll
2	Hydro_Samachurl	Hydro	Pyro	Elite	Divining_Scroll
3	Cryo_Samachurl	Cryo	Pyro	Elite	Divining_Scroll
4	Mitachurl	Anemo	Electro	Elite	Firm_Arrowhead

Conceptual Model

There are 2 forms of conceptual models we have made, the first one is the true conceptual model but it will be extremely hard to interpret because it's too large. Here it is:

Conceptual Model of Genshin Impact Knowledge Graph (KRR Project)
Sarim Aeyzaz (i210328)
Zaraar Malik (i212705)
Raffay Khan(i210335)



A-BOX

Link to conceptual model:

<https://drive.google.com/file/d/11EpZZ3f0h0P1mvO4ud1TpwQZb5wZZgsV/view?usp=sharing>

Here is the second one which is taken from Protégé (it's missing the object properties but at-least it is readable):



Identified Vocabularies

- canEasilyDefeat
- canWeildWeapon
- causesReaction
- dropsMaterial
- elementAlignment
- enemyWeakAgainst
- hasConstellation
- hasResident
- hasRole
- isEasilyDefeatedBy
- isEnemyType
- isWeaponType
- requiresElement
- residesAt
- usesCharacterUpgradeMaterials
- usesWeaponType
- usesWeaponUpgradeMaterial
- weaponUsedBy
- birthday
- hasAscensionPassive
- hasGender
- isArchon
- isLimitedCharacter
- rarity
- releaseDate
- foaf:name

External Linked Dataset

I tried to search a lot for an external dataset related to Genshin but sadly, I couldn't find a single thing. The only source I could find is partial information on character only in Wikidata and thus, just decided to link all character of Genshin listed on Wikidata to my characters.

However, do note that Wikidata doesn't have every character of Genshin impact on its website so not every single character could be linked :(.

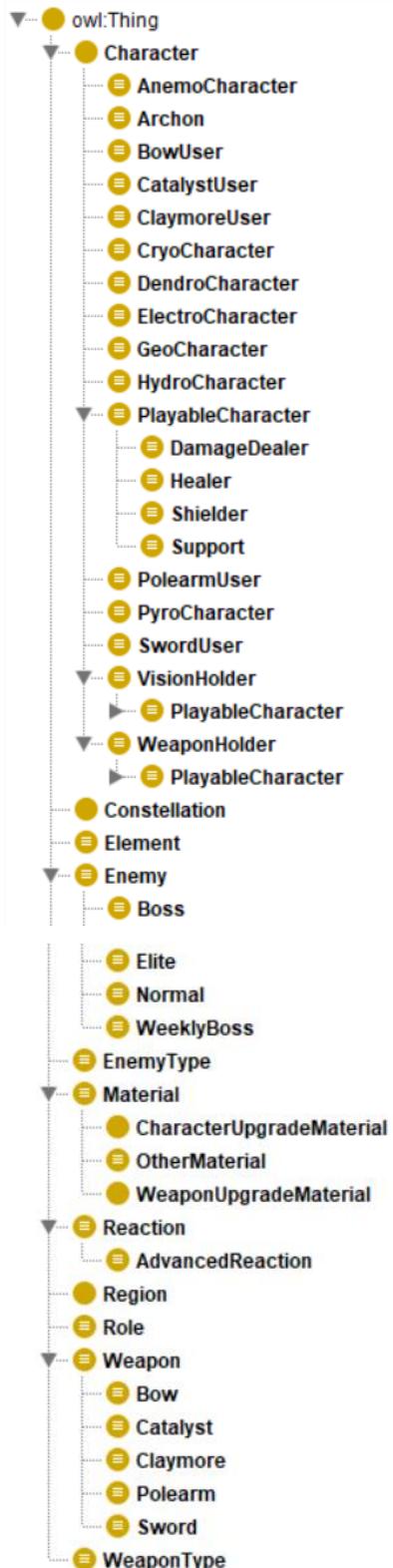
Here it is: <https://www.wikidata.org/wiki/Q65059474>

The screenshot shows the Wikidata item page for the character Candace. At the top, there's a banner for 'Data Modelling Days' from November 30th to December 2nd. The main content includes:

- Label:** Candace
- Description:** playable character in Genshin Impact
- Statements:**
 - instance of:** video game character
 - sex or gender:** female
 - country of citizenship:** Sumeru
- Also known as:** Wikipedia, Wikibooks, Wikinews, Wikiquote, Wikisource, Wikiversity, Wikivoyage, Wiktionary, Multilingual sites

Designing the Ontology

Ontology Class Hierarchy



Description:

In this class diagram, most of these classes are supposed to be inferred. This was done to increase the semantic expressivity of the data (And allow SWRL rules usage later on).

Below were the requirements for the Ontology file, I'll list each one of them and highlight where they are fulfilled with a justified reason:

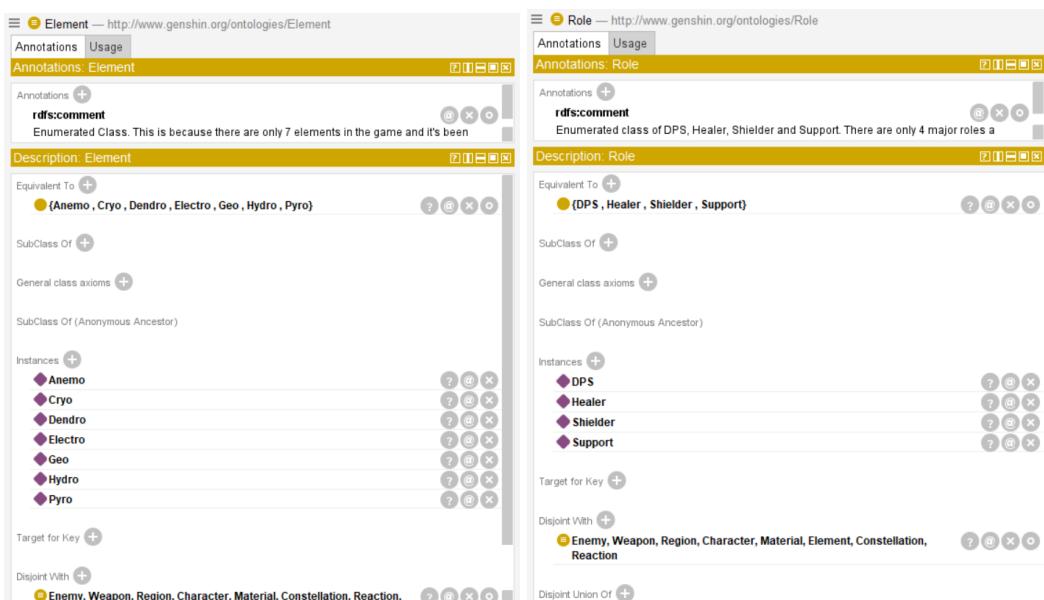
Ontology 11 Requirements

1. At least one class defined as enumeration of its individuals:

In my ontology, four classes were defined as enumerations, Element, Role, EnemyType and WeaponType.

- Elements contain 7 individuals (Pyro, Cryo, Anemo, Geo, Dendro, Electro, Hydro)
- Role contains 4 individuals (DPS, Healer, Shielder, Support).
- EnemyType contains 4 individuals (Normal, Elite, Boss, Weekly Boss)
- WeaponType contains 5 individuals (Spear, Bow, Claymore, Polearm, Catalyst)

It's heavily implied that there will be 7 elements only throughout the entire story and there are 4 major roles in the game currently. This sits well with the game's design since you can have a maximum of 4 characters in your party. EnemyType and WeaponType may increase in the future but it's been 3 years since the game's launch and these have stayed the same.



2. At least one class defined using property cardinality restrictions:

All reactions in Genshin happen between at-least 2 elements (and advanced reactions occurring between more than 2 elements. Therefore, cardinality restrictions are placed where each individual needs to be related to ‘requiresElement’ to 2 or more than 2 Elements respectively.

The screenshot displays two ontology classes in a semantic web editor:

Reaction (<http://www.genshin.org/ontologies/Reaction>)

- Annotations:** rdfs:comment: Reactions are well....reactions caused when two elements combine, creating a new effect. You
- Description:** Reaction
- Equivalent To:** requiresElement min 2 Element
- SubClass Of:** Reaction
- General class axioms:** None
- SubClass Of (Anonymous Ancestor):** None
- Instances:**
 - Burning
 - Electro-Charged
 - Frozen
 - Melt
 - Overloaded
 - Quicken
 - Superconduct
 - Vaporize

AdvancedReaction (<http://www.genshin.org/ontologies/AdvancedReaction>)

- Annotations:** rdfs:comment: Those specific reactions where more than 2 elements can participate in
- Description:** AdvancedReaction
- Equivalent To:** requiresElement min 3 Element
- SubClass Of:** Reaction
- General class axioms:** None
- SubClass Of (Anonymous Ancestor):** Reaction
- Instances:**
 - Burgeon
 - Crystallize
 - Hyperbloom
 - Swirl

3. At least one class defined using property range restriction

Almost all of my character classes are defined using this method. GeoCharacter, AnemoCharacter, HydroCharacter, CryoCharacter, PyroCharacter, DendroCharacter, GeoCharacter needs to have an individual which is related to “elementAlignment” property on a specific element (e.g. CryoCharacter individual needs to have elementAlignment to cryo), BowUser, ClaymoreUser, PolearmUser, CatalystUser and SwordUser are related in a similar manner.

The image shows two side-by-side ontology editor interfaces. The left interface is for 'CryoCharacter' and the right is for 'PolearmUser'. Both interfaces have tabs for 'Annotations' and 'Usage'. The 'Annotations' tab is active for both. The 'Description' section for CryoCharacter contains the comment: 'A character which has a Cryo Vision'. The 'Equivalent To' section lists 'Character' and '(elementAlignment value Cryo)'. The 'SubClass Of' section lists 'VisionHolder'. The 'General class axioms' section lists 'AnemoCharacter or CryoCharacter or DendroCharacter or ElectroCharacter or GeoCharacter or HydroCharacter or PyroCharacter'. The 'Instances' section lists characters: Aloy, Charlotte, Chongyun, Diona, Eula, Fremenit, Ganyu, Kaeya, Candace, Cyuno, Hu_Tao, Mika, Raiden_Shogun, Rosaria, Shenhe, Thoma, and Xiangling. The right interface for 'PolearmUser' has a similar structure. It lists 'Character' and '(usesWeaponType value "Polearm")' in the 'Equivalent To' section, and 'WeaponHolder' in the 'SubClass Of' section. The 'General class axioms' section lists 'BowUser or CatalystUser or ClaymoreUser or PolearmUser or SwordUser'. The 'Instances' section lists the same characters as the left interface.

4. At least one class defined as a union of classes

Vision holder class is a union of property restricted class of characters of each element.

Similarly, Weapon holder is a union of property restricted class of character of each weapon type:

The image shows two side-by-side ontology editor interfaces. The left interface is for 'VisionHolder' and the right is for 'WeaponHolder'. Both interfaces have tabs for 'Annotations' and 'Usage'. The 'Annotations' tab is active for both. The 'Description' section for VisionHolder contains the comment: 'Any character which has a vision (visions are basically elemental affinities given to characters)'. The 'Equivalent To' section lists 'Character' and '(elementAlignment value AnemoCharacter or CryoCharacter or DendroCharacter or ElectroCharacter or GeoCharacter or HydroCharacter or PyroCharacter)'. The right interface for 'WeaponHolder' has a similar structure. It lists 'Character' and '(usesWeaponType value "BowUser or CatalystUser or ClaymoreUser or PolearmUser or SwordUser")' in the 'Equivalent To' section, and 'WeaponHolder' in the 'SubClass Of' section. The 'General class axioms' section lists 'BowUser or CatalystUser or ClaymoreUser or PolearmUser or SwordUser'. The 'Instances' section lists the same characters as the left interface.

Materials class is also a union of CharacterUpgradeMaterials and WeaponUpgradeMaterials.

Similarly, Weapon class is a union of all people weapon type classes:

The screenshot shows two ontology cards side-by-side. The left card is for the 'Material' class, which is defined as a union of 'CharacterUpgradeMaterial' and 'WeaponUpgradeMaterial'. It includes annotations like 'rdfs:comment' stating 'Materials are resources dropped by enemies. Used to level characters or weapons' and an equivalent class statement. The right card is for the 'Weapon' class, which is defined as a union of 'Bow', 'Catalyst', 'Claymore', 'Polearm', and 'Sword'. It includes annotations like 'rdfs:comment' stating 'Weapons wielded by characters' and an equivalent class statement.

5. At least one class defined as an intersection of classes

PlayableCharacters is one such class where if a character is both a vision holder and wields a weapon, they are deemed as playable characters. (Since all playable characters in Genshin hold a vision and are able to wield a weapon)

The screenshot shows the 'PlayableCharacter' class card. It defines the class as an intersection of 'VisionHolder' and 'WeaponHolder'. It includes annotations like 'rdfs:comment' stating 'Every playable character in genshin has a vision and has some preferred weapon they like' and an equivalent class statement. The class is also a subclass of 'Character', 'VisionHolder', and 'WeaponHolder', and is a general class axiom. It is also a subclass of several specific character types like AnemoCharacter, CryoCharacter, DendroCharacter, ElectroCharacter, GeoCharacter, HydroCharacter, PyroCharacter, BowUser, CatalystUser, ClaymoreUser, PolearmUser, and SwordUser.

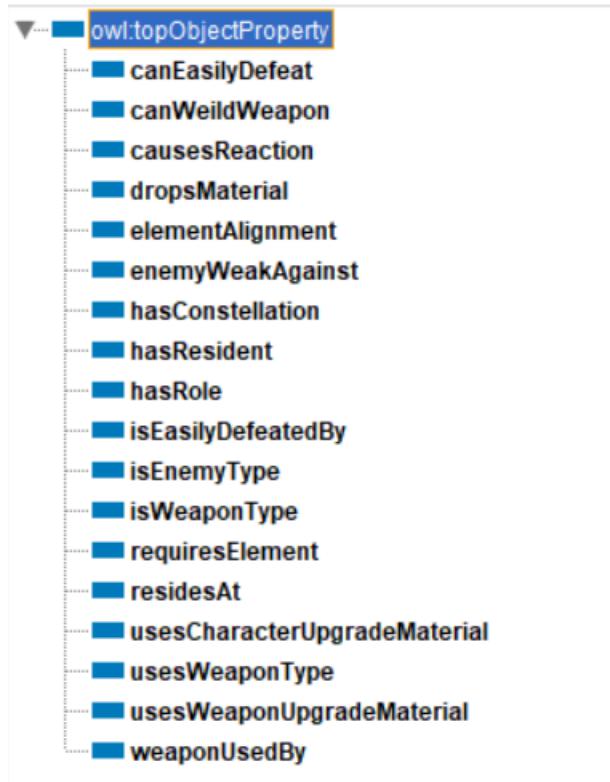
6. At least one class defined as a complement of classes

Here, we have other materials defined as those materials which are not character upgrade materials nor weapon upgrade materials:

The screenshot shows the 'OtherMaterial' class card. It defines the class as a complement of 'CharacterUpgradeMaterial' and 'WeaponUpgradeMaterial'. It includes annotations like 'rdfs:comment' stating 'Any material that isn't a weapon upgrade material nor a character upgrade material' and an equivalent class statement.

7. At least 7 object properties in total

Here you go :D. There's annotations for each object property explaining it too in the RDF/XML file.



8. At least one object property should be functional

ElementAlignment is a functional property. It's made functional because a character cannot have more than 1 element. If they do, it should mean that the two element instances are the same individuals. This is a fact imposed by the story of Genshin for all of its characters. Even the archons have only 1 element aligned to them.

Annotations: elementAlignment

rdfs:comment

Any character having an element. It's made functional because a character cannot have more than 1 element. If they do, it should mean that the two element instances are the same individuals.

Char Description: elementAlignment

Functional

9. At least one object property should be inverse functional

hasConstellation is an inverse functional property. Every character has their own unique constellation in the game (To be more in-depth without spoiling, it represents ‘them’ in the ‘sky’). Therefore, a constellation cannot belong to more than 1 character. If that case ever happens, then both character individuals are to be inferred as the same individual.

The screenshot shows the 'Annotations' tab for the 'hasConstellation' property. The 'rdfs:comment' field contains: 'Any character that has a constellation. This is mentioned in the database and has been made inverse functional because a constellation cannot belong to more than 1 character. If that case ever happens, then both character individuals are the same individual.' The 'Description' tab shows the following restrictions:

- Equivalent To: None
- SubProperty Of: None
- Inverse Of: None
- Domains (intersection): Character
- Ranges (intersection): Constellation

10. At least 3 object properties should have some range restrictions

Here you go! I'll show 4 of them but a lot of my object properties have range and domains restrictions:

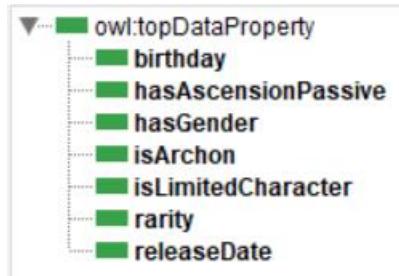
The screenshots show four properties with their respective range restrictions:

- canEasilyDefeat**: Range is Enemy.
- canWieldWeapon**: Range is Weapon.
- causesReaction**: Range is Reaction.
- dropsMaterial**: Range is Material.

Each screenshot displays the 'Annotations' and 'Description' tabs with their respective details and restrictions.

11. At least 7 datatype properties in total

Here you go!



Mapping

Here are the 4 databases I used to map into my RDF/XML ontology:

Character Table (Displaying only 5 rows)

	character_name	rarity	region	vision	weapon_type	constellation	birthday	ascension_boss	limited	ascension	role	gender
0	Albedo	5	Mondstadt	Geo	Sword	Princeps_Cretaceus	2023-09-13	Basalt_Pillar	True	Geo_DMG_Bonus	Support	Male
1	Alhaitham	5	Sumeru	Dendro	Sword	Vultur_Volans	2023-02-11	Pseudo-Stamens	True	Dendro_DMG_Bonus	DPS	Male
2	Aloy	5	Mondstadt	Cryo	Bow	Nora_Fortis	2023-04-04	Crystalline_Bloom	True	Cryo_DMG_Bonus	Support	Female
3	Amber	4	Mondstadt	Pyro	Bow	Lepus	2023-08-10	Everflame_Seed	False	ATK	Support	Female
4	Arataki_Itto	5	Inazuma	Geo	Claymore	Taurus_Iracundus	2023-06-01	Riftborn_Regalia	True	CRIT_Rate	DPS	Male

Reaction Table (Displaying only 5 rows)

	Element	causesReaction
0	Pyro	Overloaded
1	Electro	Overloaded
2	Pyro	Melt
3	Cryo	Melt
4	Pyro	Vaporize

Weapons Table (Displaying only 5 rows)

	Name	Type	Rarity	UpgradeMaterial
0	Wolf's_Gravestone	Claymore	5	Tusk_of_the_Wolf
1	Skyward_Harp	Bow	5	Luminous_Sands_from_Guyun
2	Primordial_Jade_Winged-Spear	Polearm	5	Jade_Wing
3	Aquila_Favonia	Sword	5	Tusk_of_the_Wolf
4	Lost_Prayer_to_the_Sacred_Winds	Catalyst	5	Tile_of_Decarabian's_Tower

Enemies Table (Displaying only 5 rows)

	Name	Element	WeakAgainst	EnemyType	DropsMaterial
0	Hilichurl	Anemo	Electro	Normal	Firm_Arrowhead
1	Anemo_Samachurl	Anemo	Electro	Elite	Divining_Scroll
2	Hydro_Samachurl	Hydro	Pyro	Elite	Divining_Scroll
3	Cryo_Samachurl	Cryo	Pyro	Elite	Divining_Scroll
4	Mitachurl	Anemo	Electro	Elite	Firm_Arrowhead

First I parsed my ontology using RDFLib library, and then, it was just a matter of adding the right triples into the graph. Here's an example of what it looks like for Characters:

```
for index, row in df.iterrows():
    name = URIRef(gRes+row['character_name'])
    constellation = URIRef(gRes+row['constellation'])
    characterMaterial = URIRef(gRes+row['ascension_boss'])
    role = URIRef(gRes+row['role'])

    g.add((name, RDF.type, URIRef(gOnto + "Character")))
    g.add((name, RDF.type, OWL.NamedIndividual))
    g.add((name, FOAF.name, Literal(row['character_name'])))
    g.add((name, RDFS.label, Literal(row['character_name']), datatype = XSD.string))

    g.add((constellation, RDF.type, URIRef(gOnto + "Constellation")))
    g.add((constellation, RDF.type, OWL.NamedIndividual))
    g.add((constellation, RDFS.label, Literal(row['constellation']), datatype = XSD.string))

    g.add((characterMaterial, RDF.type, URIRef(gOnto + "Material")))
    g.add((characterMaterial, RDF.type, OWL.NamedIndividual))
    g.add((characterMaterial, RDFS.label, Literal(row['ascension_boss']), datatype = XSD.string))

    g.add((role, RDF.type, URIRef(gOnto + "Role")))
    g.add((role, RDF.type, OWL.NamedIndividual))
    g.add((role, RDFS.label, Literal(row['role']), datatype = XSD.string))

    g.add((name, URIRef(gOnto + "rarity"), Literal(row['rarity']), datatype = XSD.integer))
    g.add((name, URIRef(gOnto + "residesAt"), URIRef(gRes + row['region'])))
    g.add((name, URIRef(gOnto + "elementAlignment"), URIRef(gRes + row['vision'])))
    g.add((name, URIRef(gOnto + "usesWeaponType"), Literal(row['weapon_type']), datatype= XSD.string))
    g.add((name, URIRef(gOnto + "hasConstellation"), constellation))
    g.add((name, URIRef(gOnto + "birthday"), Literal(row['birthday']), datatype= XSD.dateTime))
    g.add((name, URIRef(gOnto + "usesCharacterUpgradeMaterial"), characterMaterial))
    g.add((name, URIRef(gOnto + "isLimitedCharacter"), Literal(row['limited']), datatype = XSD.boolean))
    g.add((name, URIRef(gOnto + "hasAscensionPassive"), Literal(row['ascension']), datatype = XSD.string))
    g.add((name, URIRef(gOnto + "hasRole"), role))
    g.add((name, URIRef(gOnto + "hasGender"), Literal(row['gender']), datatype = XSD.string))
```

After every time I added a table, I saved the updated ontology into a new file:

```
1 g.serialize(destination="afterAddingCharacters.rdf", format="application/rdf+xml")
```

Linking

Genshin Impact characters from Wikidata were linked with my characters in my ontology. Unfortunately, Wikidata didn't contain all of the characters so only some characters could be linked to there.

Linking with Wikidata

```
: 1 # Keqing
 2 character = URIRef(gRes + 'Keqing')
 3 wikidataLink = URIRef(wd + 'Q107287676')
 4 g.add((character, OWL.sameAs, wikidataLink))
 5
 6 # Yun Jin
 7 character = URIRef(gRes + 'Yun_Jin')
 8 wikidataLink = URIRef(wd + 'Q110653384')
 9 g.add((character, OWL.sameAs, wikidataLink))
10
11 # Shenhe
12 character = URIRef(gRes + 'Shenhe')
13 wikidataLink = URIRef(wd + 'Q110677319')
14 g.add((character, OWL.sameAs, wikidataLink))
15
16 # Ganyu
17 character = URIRef(gRes + 'Ganyu')
18 wikidataLink = URIRef(wd + 'Q107618618')
19 g.add((character, OWL.sameAs, wikidataLink))
20
21 # Diluc
22 character = URIRef(gRes + 'Diluc')
23 wikidataLink = URIRef(wd + 'Q107263978')
24 g.add((character, OWL.sameAs, wikidataLink))
25
26 # Hu Tao
27 character = URIRef(gRes + 'Barbara')
28 wikidataLink = URIRef(wd + 'Q110911467')
29 g.add((character, OWL.sameAs, wikidataLink))
30
31 # Yanfei
32 character = URIRef(gRes + 'Yanfei')
33 wikidataLink = URIRef(wd + 'Q110965539')
34 g.add((character, OWL.sameAs, wikidataLink))
35
36 # Amber
37 character = URIRef(gRes + 'Amber')
38 wikidataLink = URIRef(wd + 'Q111008775')
39 g.add((character, OWL.sameAs, wikidataLink))
40
```

As such, 30 characters were successfully linked with Wikidata

Reasoning

SWRL Rules

Rule 1-5:

I wanted to make some inferences that if there is a character belonging to BowUser defined class, then it should be able to wield every single instance of Bow class. Unfortunately, OWL didn't have the expressivity to allow this so I had to use some SWRL rules to make these inferences possible. Here's how the above situation looks like as a SWRL rule:

```
genshinOntology:Bow(?b) ^ genshinOntology:BowUser(?a)  
-> genshinOntology:canWeildWeapon(?a, ?b)
```

As such, 5 rules were made like this, each one trying to make inferences between specific weapon types and their respective holders.

Rule 6:

Another place SWRL rule was implemented was to answer one of my competency question: "What characters can easily defeat X enemy?"

I had the weakness element of the enemy and each character's element alignment. All I had to do in SWRL was to make such question's relation a possibility. Here's how it looks like:

```
genshinOntology:elementAlignment(?character, ?element) ^ genshinOntology:enemyWeakAgainst(?enemy, ?element) ^  
genshinOntology:Character(?character) ^ genshinOntology:Enemy(?enemy) ->  
genshinOntology:canEasilyDefeat(?character, ?enemy)
```

It's a bit long, I know. Class type checks had to be in place so I wouldn't get any information where "Enemy" is weak against "Enemy". Since enemy shouldn't be the domain of this property.

Reasoning

After running the reasoner, we notice a lot of stuff happening. Here are all the places where the reasoner inferred something (SWRL rule reasonings are at the bottom):

The screenshot shows the 'Annotations' tab for the 'AnemoCharacter' ontology. It contains the following annotations:

- rdfs:comment**: A character which has an Anemo Vision

The 'Description' tab for 'AnemoCharacter' is also visible. The 'Instances' section lists the following characters, each with a status indicator (grey circle with question mark, '@', or 'X'):

- Aether
- Faruzan
- Jean
- Kaedehara_Kazuha
- Lumine
- Lynette
- Sayu
- Shikanoin_Heizou
- Sucrose
- Venti
- Wanderer
- Xiao

For element specific character classes, it perfectly reasoned which characters should belong into which elemental class

The screenshot shows the 'Annotations' tab for the 'BowUser' ontology. It contains the following annotations:

- rdfs:comment**: A character which uses a Bow

The 'Description' tab for 'BowUser' is also visible. The 'Instances' section lists the following characters, each with a status indicator:

- Aloy
- Amber
- Collei
- Diona
- Faruzan
- Fischl
- Ganyu
- Gorou
- Kujou_Sara
- Lyney
- Tartaglia
- Tighnari
- Venti
- Yelan
- Yoimiya

For weapon specific character classes, it perfectly reasoned on which characters wield a bow and thus, are now individuals of BowUser

The screenshot shows the Protege ontology editor interface for the `DamageDealer` class. The main pane displays the following details:

- Annotations:** `rdfs:comment` - Prioritizes in dealing damage to the enemy.
- Description:** DamageDealer
- Equivalent To:** PlayableCharacter and (hasRole value DPS)
- SubClass Of:** PlayableCharacter
- General class axioms:** None listed.
- SubClass Of (Anonymous Ancestor):**
 - AnemoCharacter or CryoCharacter or DendroCharacter or ElectroCharacter or GeoCharacter or HydroCharacter or PyroCharacter
 - BowUser or CatalystUser or ClaymoreUser or PolarmUser or SwordUser
 - VisionHolder and WeaponHolder
- Instances:**
 - Aether
 - Alhautham
 - Arataki_Ito
 - Cyno
 - Diluc

According to the dataset, it also reasoned perfectly on which characters are classified as damage dealers

The screenshot shows the Protege ontology editor interface for the `Boss` class. The main pane displays the following details:

- Annotations:** `rdfs:comment` - Very tough enemies (They even have a health bar shown on the screen).
- Description:** Boss
- Equivalent To:** Enemy and (enemyType value "Boss")
- SubClass Of:** Enemy
- General class axioms:** None listed.
- SubClass Of (Anonymous Ancestor):**
 - enemyType some xsd:string
- Instances:**
 - Anemo_Hypostasis
 - Cryo_Regisvine
 - Electro_Hypostasis
 - Geo_Hypostasis
 - Pyro_Regisvine

It also reasoned perfectly on which enemies should be considered as individuals of boss class

Annotations: Weekly_Boss

rdfs:comment
They are bosses but are related to the story. You can claim rewards from them once per week,

Description: Weekly_Boss

Equivalent To

- Enemy
- and (enemyType value "Weekly_Boss")

SubClass Of

- Boss

General class axioms

SubClass Of (Anonymous Ancestor)

- enemyType some xsd:string
- Enemy
- and (enemyType value "Boss")

Instances

- Andrius
- Azhdaha
- Childe
- Dvalin

Similarly, it reasoned perfectly on which enemies are of weekly boss types too :D

Annotations: CharacterUpgradeMaterial

rdfs:comment
A material which is used to upgrade characters

Description: CharacterUpgradeMaterial

Equivalent To

SubClass Of

- Material

General class axioms

SubClass Of (Anonymous Ancestor)

- CharacterUpgradeMaterial or WeaponUpgradeMaterial

Instances

- Artificed_Spare_Clockwork_Component_-Coppelia
- Artificed_Spare_Clockwork_Component_-Coppelius
- Basalt_Pillar
- Cleansing_Heart
- Crystalline_Bloom
- Dew_of_Reputation
- Dragonheir's_False_Fin
- Emperor's_Resolution
- Everflame_Seed
- Evergloom_Ring
- Fontemer_Unhorn

From the materials and characters using those materials, it inferred which materials belong under the CharacterUpgradeMaterials class

Annotations: `rdfs:comment`
A material which is used to upgrade weapons

Description: WeaponUpgradeMaterial

Equivalent To: Material

General class axioms:

SubClass Of (Anonymous Ancestor): CharacterUpgradeMaterial or WeaponUpgradeMaterial

Instances:

- Agnidus_Agate_Sliver
- Basalt_Pillar
- Broken_Fang
- Crystalline_Bloom
- Fragment_of_Decarabian's_Epic
- Guyun_Stone_Forest
- Heavy_Horn
- Jade_Wing
- Luminous_Sands_from_Guyun
- Lustrous_Stone_from_Guyun
- Mist_Veiled_Lead_Elixir

From the materials and weapons using those materials, it inferred which materials belong under the `WeaponUpgradeMaterials` class

Annotations: `rdfs:comment`
Reactions are well...reactions caused when two elements combine, creating a new effect You

Description: Reaction

Equivalent To: requiresElement min 2 Element

SubClass Of:

General class axioms:

SubClass Of (Anonymous Ancestor):

Instances:

- Burning
- Electro-Charged
- Frozen
- Melt
- Overloaded
- Quicken
- Superconduct
- Vaporize

It also inferred correctly on Reactions

☰ ● AdvancedReaction — http://www.genshin.org/ontologies/AdvancedReaction

Annotations Usage

Annotations: AdvancedReaction

Annotations +

rdfs:comment
Those specific reactions where more than 2 elements can participate in

Description: AdvancedReaction

Equivalent To +

● **requiresElement min 3 Element**

SubClass Of +

● Reaction

General class axioms +

SubClass Of (Anonymous Ancestor)

● **requiresElement min 2 Element**

Instances +

- ◆ Burgeon
- ◆ Crystallize
- ◆ Hyperbloom
- ◆ Swirl

As well as advanced reactions :)

☰ ● Region — http://www.genshin.org/ontologies/Region

Annotations Usage

Annotations: Region

Annotations +

rdfs:comment
Countries where characters usually reside in

Description: Region

Equivalent To +

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

- ◆ Fontaine
- ◆ Inazuma
- ◆ Liyue
- ◆ Mondstadt
- ◆ Snezhnaya
- ◆ Sumeru

It also inferred the regions automatically

Bow — http://www.genshin.org/ontologies/Bow

Annotations: Bow

rdfs:comment
Represents any bow in the game

Description: Bow

Equivalent To
Weapon
and (isWeaponType value "Bow")

SubClass Of
Weapon

General class axioms

SubClass Of (Anonymous Ancestor)
Bow or Catalyst or Claymore or Polearm or Sword
Bow or Catalyst or Claymore or Polearm or Sword

Instances

- Compound_Bow
- Favonius_Warbow
- Rust
- Skyward_Harp
- The_Stringless
- The_Viridescent_Hunt

Sword — http://www.genshin.org/ontologies/Sword

Annotations: Sword

rdfs:comment
Represents any sword in the game

Description: Sword

Equivalent To
Weapon
and (isWeaponType value "Sword")

SubClass Of
Weapon

General class axioms

SubClass Of (Anonymous Ancestor)
Bow or Catalyst or Claymore or Polearm or Sword
Bow or Catalyst or Claymore or Polearm or Sword

Instances

- Aquila_Favonia
- Lion's_Roar
- Prototype_Rancour
- Sacrificial_Sword
- Sword_of_Descension
- The_Black_Sword

It also correctly inferred on weapons being classified into their respective weapon categories!

SWRL Reasoning

Description: Anemo_Hypostasis	Types +	Property assertions: Anemo_Hypostasis
<p>Types +</p> <ul style="list-style-type: none"> Enemy Boss <p>Same Individual As +</p> <p>Different Individuals +</p>	<p>?</p> <p>?</p> <p>?</p> <p>?</p>	<p>enemyWeakAgainst Electro</p> <p>dropsMaterial Vayuda_Turquoise_Chunk</p> <p>elementAlignment Anemo</p> <p>isEasilyDefeatedBy Razor</p> <p>isEasilyDefeatedBy Yae_Miko</p> <p>isEasilyDefeatedBy Beidou</p> <p>isEasilyDefeatedBy Dori</p> <p>isEasilyDefeatedBy Fischl</p> <p>isEasilyDefeatedBy Lisa</p> <p>isEasilyDefeatedBy Cyno</p> <p>isEasilyDefeatedBy Kuki_Shinobu</p> <p>isEasilyDefeatedBy Raiden_Shogun</p> <p>isEasilyDefeatedBy Kujou_Sara</p> <p>isEasilyDefeatedBy Keqing</p>
Description: Barbara	Types +	Property assertions: Barbara
<p>Types +</p> <ul style="list-style-type: none"> Character CatalystUser Healer HydroCharacter <p>Same Individual As +</p> <p>Different Individuals +</p>	<p>?</p> <p>?</p> <p>?</p> <p>?</p>	<p>hasRole Healer</p> <p>hasConstellation Crater</p> <p>usesCharacterUpgradeMaterial Cleansing_Heart</p> <p>residesAt Mondstadt</p> <p>elementAlignment Hydro</p> <p>canEasilyDefeat Large_Electro_Slime</p> <p>canEasilyDefeat Electro_Hypostasis</p> <p>canEasilyDefeat Electro_Slime</p> <p>canEasilyDefeat Dendro_Slime</p> <p>canEasilyDefeat Electro_Abyss_Mage</p> <p>canWeildWeapon Solar_Pearl</p> <p>canWeildWeapon Mappa_Mare</p> <p>canWeildWeapon Wine_and_Song</p> <p>canWeildWeapon Lost_Prayer_to_the_Sacred_Winds</p> <p>canWeildWeapon Widsith</p>
Description: Aquila_Favonia	Types +	Property assertions: Aquila_Favonia
<p>Types +</p> <ul style="list-style-type: none"> Weapon Sword <p>Same Individual As +</p> <p>Different Individuals +</p>	<p>?</p> <p>?</p>	<p>usesWeaponUpgradeMaterial Tusk_of_the_Wolf</p> <p>weaponUsedBy Furina</p> <p>weaponUsedBy Kamisato_Ayato</p> <p>weaponUsedBy Layla</p> <p>weaponUsedBy Kirara</p> <p>weaponUsedBy Xingqiu</p> <p>weaponUsedBy Nilou</p> <p>weaponUsedBy Lumine</p> <p>weaponUsedBy Kaeya</p> <p>weaponUsedBy Alhaitham</p> <p>weaponUsedBy Kuki_Shinobu</p> <p>weaponUsedBy Jean</p> <p>weaponUsedBy Keqing</p> <p>weaponUsedBy Aether</p>

As we can see, SWRL rules allowed such automatic inferences to happen between individuals. This wasn't possible in OWL because automating relations between instances of 2 different classes (like a dense neural network connection between 2 layers) wasn't possible with traditional owl expressivity.

Note: After mapping and linking, we saved everything into an RDF+XML file called the ‘afterAddingEverything.rdf’. Then we loaded our RDF+XML file into Protege and then we started the reasoner. After triples were reasoned, we saved the file into another RDF+XML file called ‘inferredKnowledgeAdded.rdf’.

Linked Data Reasoning

The screenshot shows the Protege reasoning interface. On the left, under 'Types', three classes are listed: PolearmUser, PyroCharacter, and Shielder. Below that, under 'Same Individual As', an individual named Thoma is shown. Under 'Different Individuals', there is a '+' button. On the right, under 'Object property assertions', a list of triples is shown:

Object property assertions	Value	Subject
canEasilyDefeat	Cryo_Abyss_Mage	?
canEasilyDefeat	Cryo_Regisvine	?
canEasilyDefeat	Cryo_Slime	?
canEasilyDefeat	Large_Cryo_Slime	?
canEasilyDefeat	Cryo_Samachurl	?
canEasilyDefeat	Andrius	?
canEasilyDefeat	Hydro_Samachurl	?
canWeildWeapon	Crescent_Pike	?
canWeildWeapon	Dragonl's_Bane	?
canWeildWeapon	Prototype_Starglitter	?
canWeildWeapon	Primordial_Jade_Winged-Spear	?
canWeildWeapon	Royal_Spear	?
canWeildWeapon	Deathmatch	?
elementAlignment	Pyro	?
hasConstellation	Polarum_Scutum	?

Here you can see the Wikidata entry for Thoma was successfully linked with our Thoma character in the ontology file.

Publishing

We published the data into GraphDB and Fuseki to create a SPARQL endpoint for our ontology. Here is how it looks like:

Active repository

Local

 GenshinImpact · Genshin Impact Knowledge Graph (after [Import RDF data](#) [Export RDF data](#))

total statements **10,995** 7,910 explicit
3,085 inferred
1.39 expansion ratio

GraphDB

SPARQL Query & Update

```
PREFIX ontology: <http://www.genshin.org/ontology/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?characterName ?voiceActor ?voiceActorName
WHERE {
?character a ontology:Character;
rdfs:label ?characterName.
?voiceActor rdfs:label ?voiceActorName.
SERVICE <https://query.wikidata.org/sparql> {
?character wdt:P725 ?voiceActor.
?voiceActor rdfs:label ?voiceActorName.
FILTER (?voiceActorName = "un")
}
```

Table Raw Response Pivot Table Google Chart Run

Filter query results ▲ Showing results from 1 to 60 of 60. Query took 34s, yesterday at 22:36.

	characterName	voiceActor	voiceActorName
1	"Shenhe"	http://www.wikidata.org/entity/Q49566	"Ayako Kawasumi"@en
2	"Fischl"	http://www.wikidata.org/entity/Q44552	"Maaya Uchida"@en
3	"Raiden_Shogun"	http://www.wikidata.org/entity/Q49552	"Miyuki Sawashiro"@en
4	"Lisa"	http://www.wikidata.org/entity/Q237371	"Rie Tanaka"@en
5	"Thoma"	http://www.wikidata.org/entity/Q361574	"Masakazu Morita"@en

Validation Using SPARQL Endpoint

Here are the 9 (+ 1 bonus) competency questions we expected our graph to answer. I added the bonus in during deliverable 4 because I was able to do Federated Queries using my graph

Q1. What materials do I need to level up X character? E.g Kaeya

The screenshot shows a SPARQL query editor and its results. The query is:

```
PREFIX ontology: <http://www.genshin.org/ontology/>
PREFIX resource: <http://www.genshin.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

# Q1. What material do I need to level up X character? E.g Kaeya
SELECT ?character ?characterName ?materialName {
    ?material rdfs:label ?materialName;
    a ontology:Material.
    ?character ontology:usesCharacterUpgradeMaterial ?material;
    a ontology:Character;
    foaf:name ?characterName.
    FILTER(?character = resource:Kaeya).
}
```

The results table shows one row:

character	characterName	materialName
1 http://www.genshin.org/resource/Kaeya	"Kaeya"	"Hoarfrost_Core"

Q2. What materials do I need to level up X weapon? E.g Favonius Warbow

The screenshot shows a SPARQL query editor and its results. The query is:

```
PREFIX ontology: <http://www.genshin.org/ontology/>
PREFIX resource: <http://www.genshin.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

# Q2. What materials do I need to level up X weapon? E.g Favonius Warbow
SELECT ?weapon ?weaponName ?materialName {
    ?material rdfs:label ?materialName;
    a ontology:Material.
    ?weapon ontology:usesWeaponUpgradeMaterial ?material;
    a ontology:Weapon;
    foaf:name ?weaponName.
    FILTER(?weapon = resource:Favonius_Warbow).
}
```

The results table shows one row:

weapon	weaponName	label
1 http://www.genshin.org/resource/Favonius_Warbow	"Favonius_Warbow"	"Basalt_Pillar"

Q3. List all characters of X element. E.g Hydro

The screenshot shows a SPARQL query interface with the following details:

- Query Editor:**
 - PREFIX ontology: <http://www.genshin.org/ontology/>
 - PREFIX resource: <http://www.genshin.org/resource/>
 - PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 - PREFIX foaf: <http://xmlns.com/foaf/0.1/>
 - SELECT ?character ?characterName ?element {
 ?character a ontology:Character;
 ontology:elementAlignment ?element;
 foaf:name ?characterName.
 }
 FILTER (?element = resource:Hydro)
}
- Run Button:** A red "Run" button is located at the bottom right of the query editor.
- Results Table:**
 - Header: character, characterName, element
 - Rows (12 total):

character	characterName	element
http://www.genshin.org/resource/Furina	"Furina"	http://www.genshin.org/resource/Hydro
http://www.genshin.org/resource/Kamisato_Ayato	"Kamisato_Ayato"	http://www.genshin.org/resource/Hydro
http://www.genshin.org/resource/Nilou	"Nilou"	http://www.genshin.org/resource/Hydro
http://www.genshin.org/resource/Xingqiu	"Xingqiu"	http://www.genshin.org/resource/Hydro
http://www.genshin.org/resource/Barbara	"Barbara"	http://www.genshin.org/resource/Hydro
https://www.wikidata.org/wiki/Q110911467	"Barbara"	http://www.genshin.org/resource/Hydro
http://www.genshin.org/resource/Candace	"Candace"	http://www.genshin.org/resource/Hydro
https://www.wikidata.org/wiki/Q121919991	"Candace"	http://www.genshin.org/resource/Hydro
http://www.genshin.org/resource/Tartaglia	"Tartaglia"	http://www.genshin.org/resource/Hydro
http://www.genshin.org/resource/Yelan	"Yelan"	http://www.genshin.org/resource/Hydro
https://www.wikidata.org/wiki/Q117195714	"Yelan"	http://www.genshin.org/resource/Hydro
http://www.genshin.org/resource/Mona	"Mona"	http://www.genshin.org/resource/Hydro

Q4. Which element is involved in the most reactions?

The screenshot shows a SPARQL query interface with the following details:

- Query Editor:**
 - PREFIX ontology: <http://www.genshin.org/ontology/>
 - PREFIX resource: <http://www.genshin.org/resource/>
 - PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 - PREFIX foaf: <http://xmlns.com/foaf/0.1/>
 - # Q4. Which element is involved in the most reactions?
 SELECT ?element (COUNT(?reaction) AS ?reactionCount)
 WHERE {
 ?reaction a ontology:Reaction;
 ontology:requiresElement ?element .
 }
 GROUP BY ?element
 ORDER BY DESC(?reactionCount)
 LIMIT 1
}
- Run Button:** A red "Run" button is located at the bottom right of the query editor.
- Results Table:**
 - Header: element, reactionCount
 - Row (1 total):

element	reactionCount
http://www.genshin.org/resource/Electro	7

Q5. Which element has the most healers?

The screenshot shows a SPARQL query interface with the following details:

- Query Editor:**
 - PREFIX ontology: <http://www.genshin.org/ontology/>
 - PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 - # Q5. Which element has the most healers?
 SELECT ?element ?elementName (COUNT(?healer) AS ?healerCount)
 WHERE {
 ?healer a ontology:Healer;
 ontology:elementAlignment ?element .
 ?element rdfs:label ?elementName .
 }
 GROUP BY ?element ?elementName
 ORDER BY DESC(?healerCount)
 LIMIT 1
}
- Run Button:** A red "Run" button is located at the bottom right of the query editor.
- Results Table:**
 - Header: element, elementName, healerCount
 - Row (1 total):

element	elementName	healerCount
http://www.genshin.org/resource/Cryo	"Cryo"	4

Q6. List me all the weapons a specific character can use? E.g Kaeya

The screenshot shows a SPARQL query interface with the following details:

- Query Editor:** Displays the SPARQL query:


```

PREFIX ontology: <http://www.genshin.org/ontology/>
PREFIX resource: <http://www.genshin.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# Q6. List me all the weapons a specific character can use? E.g Kaeya
SELECT ?character ?characterName ?weapon ?weaponName
WHERE {
  ?character a ontology:Character;
              rdfs:label ?characterName;
              ontology:canUseWeapon ?weapon .
  ?weapon a ontology:Weapon;
          rdfs:label ?weaponName .
  FILTER(?character = resource:Kaeya).
}
      
```
- Results Table:** Shows the results in a tabular format with columns: character, characterName, weapon, and weaponName. There are 6 rows of data.

character	characterName	weapon	weaponName
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Aquila_Favonia	"Aquila_Favonia"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Prototype_Rancour	"Prototype_Rancour"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Sacrificial_Sword	"Sacrificial_Sword"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Sword_of_Descension	"Sword_of_Descension"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/The_Black_Sword	"The_Black_Sword"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Lions_Roar	"Lions_Roar"

Q7. What characters are strong against X boss? E.g Pyro Regisvine

The screenshot shows a SPARQL query interface with the following details:

- Query Editor:** Displays the SPARQL query:


```

PREFIX ontology: <http://www.genshin.org/ontology/>
PREFIX resource: <http://www.genshin.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# Q7. What characters are strong against X boss? E.g Pyro Regisvine
SELECT DISTINCT ?characterName ?boss
WHERE {
  ?character a ontology:Character;
              rdfs:label ?characterName;
              ontology:canEasilyDefeat ?boss .
  ?boss a ontology:Boss .
  FILTER(?boss = resource:Pyro_Regisvine).
}
      
```
- Results Table:** Shows the results in a tabular format with columns: characterName and boss. There are 15 rows of data.

characterName	boss
"Aloy"	http://www.genshin.org/resource/Pyro_Regisvine
"Kaeya"	http://www.genshin.org/resource/Pyro_Regisvine
"Kamisato_Ayaka"	http://www.genshin.org/resource/Pyro_Regisvine
"Layla"	http://www.genshin.org/resource/Pyro_Regisvine
"Qiqi"	http://www.genshin.org/resource/Pyro_Regisvine
"Charlotte"	http://www.genshin.org/resource/Pyro_Regisvine
"Chongyun"	http://www.genshin.org/resource/Pyro_Regisvine
"Diona"	http://www.genshin.org/resource/Pyro_Regisvine
"Eula"	http://www.genshin.org/resource/Pyro_Regisvine
"Freminet"	http://www.genshin.org/resource/Pyro_Regisvine
"Ganyu"	http://www.genshin.org/resource/Pyro_Regisvine
"Mika"	http://www.genshin.org/resource/Pyro_Regisvine
"Rosaria"	http://www.genshin.org/resource/Pyro_Regisvine
"Shenhe"	http://www.genshin.org/resource/Pyro_Regisvine
"Wriothesley"	http://www.genshin.org/resource/Pyro_Regisvine

Q8: What two characters can be used to cause a Vaporize reaction?

```

Competency 1 × Competency 2 × Competency 3 × Competency 4 × Competency 5 ×
Competency 6 × Competency 7 × Competency 8 × Competency 9 × Competency 10 × ⓘ
1 PREFIX ontology: <http://www.genshin.org/ontology/>
2 PREFIX resource: <http://www.genshin.org/resource/>
3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 #Q8: What two characters can be used to cause a Vaporize reaction?
7
8 SELECT ?name1 ?name2 ?reaction
9 WHERE {
10   ?char1 rdf:type ontology:Character;
11   rdfs:label ?name1;
12   ontology:elementAlignment ?element1 .
13   ?char2 rdf:type ontology:Character;
14   rdfs:label ?name2;
15   ontology:elementAlignment ?element2 .
16   ?reaction ontology:requiresElement ?element1;
17   ontology:requiresElement ?element2 .
18   FILTER (?element1 != ?element2) .
19   FILTER (?reaction = resource:Vaporize) .
20 }
21

```

Table Raw Response Pivot Table Google Chart Download as

Filter query results ⚙ Showing results from 1 to 540 of 540. Query took 0.1s, yesterday at 21:04.

	name1	name2	reaction
1	'Furina'	'Amber'	http://www.genshin.org/resource/Vaporize
2	'Furina'	'Amber'	http://www.genshin.org/resource/Vaporize
3	'Furina'	'Bennett'	http://www.genshin.org/resource/Vaporize
4	'Furina'	'Dehya'	http://www.genshin.org/resource/Vaporize
5	'Furina'	'Diluc'	http://www.genshin.org/resource/Vaporize
6	'Furina'	'Diluc'	http://www.genshin.org/resource/Vaporize
7	'Furina'	'Hu_Tao'	http://www.genshin.org/resource/Vaporize
8	'Furina'	'Klee'	http://www.genshin.org/resource/Vaporize
9	'Furina'	'Lyney'	http://www.genshin.org/resource/Vaporize
10	'Furina'	'Thoma'	http://www.genshin.org/resource/Vaporize
11	'Furina'	'Thoma'	http://www.genshin.org/resource/Vaporize
12	'Furina'	'Xiangling'	http://www.genshin.org/resource/Vaporize

Q9: List me a team of 4 characters, each with a different role, which can cause a hyperbloom reaction together

```

Competency 1 × Competency 2 × Competency 3 × Competency 4 × Competency 5 ×
Competency 6 × Competency 7 × Competency 8 × Competency 9 × Competency 10 × ⓘ
1 PREFIX ontology: <http://www.genshin.org/ontology/>
2 PREFIX resource: <http://www.genshin.org/resource/>
3 PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
4 #Q9: List me a team of 4 characters, each with a different role, which
5   can cause a hyperbloom reaction together
6 SELECT DISTINCT ?nameLabel1 ?nameLabel2 ?nameLabel3 ?nameLabel4
7 WHERE {
8   ?name1 a ontology:PlayableCharacter;
9   ontology:hasRole ?role1;
10  rdfs:label ?nameLabel1;
11  ontology:elementAlignment ?elem1.
12  ?name2 a ontology:PlayableCharacter;
13  ontology:hasRole ?role2;
14  rdfs:label ?nameLabel2;
15  ontology:elementAlignment ?elem2.
16  ?name3 a ontology:PlayableCharacter;
17  ontology:hasRole ?role3;
18  rdfs:label ?nameLabel3;
19  ontology:elementAlignment ?elem3.
20  ?name4 a ontology:PlayableCharacter;
21  ontology:hasRole ?role4;
22  rdfs:label ?nameLabel4;
23  ontology:elementAlignment ?elem4.
24  ?reaction ontology:requiresElement ?elem1, ?elem2, ?elem3, ?elem4.
25  FILTER (?reaction = resource:Hyperbloom)
26  FILTER (?name1 != ?name2 && ?name1 != ?name3 && ?name1 != ?name4 &&
27  ?name2 != ?name3 && ?name2 != ?name4 && ?name3 != ?name4)
28  FILTER (?role1 != ?role2 && ?role1 != ?role3 && ?role1 != ?role4 &&
29  ?role2 != ?role3 && ?role2 != ?role4 && ?role3 != ?role4)
30  FILTER ((?elem1 != ?elem2 && ?elem1 != ?elem3 && ?elem1 != ?elem4) ||
31  (?elem1 != ?elem2 && ?elem1 != ?elem4 && ?elem2 != ?elem4) ||
32  (?elem1 != ?elem3 && ?elem1 != ?elem4 && ?elem3 != ?elem4) ||
33  (?elem2 != ?elem3 && ?elem2 != ?elem4 && ?elem3 != ?elem4))

```

Table Raw Response Pivot Table Google Chart Download as 1 2 3 4 5 ↻

Filter query results ⚙ Showing results from 1 to 1,000 of 8,712. Query took 0.4s, moments ago.

	nameLabel1	nameLabel2	nameLabel3	nameLabel4
1	'Alhaitham'	'Baizhu'	'Beidou'	'Barbara'
2	'Alhaitham'	'Baizhu'	'Beidou'	'Sangonomiya_Kokomi'
3	'Alhaitham'	'Baizhu'	'Dori'	'Barbara'
4	'Alhaitham'	'Baizhu'	'Dori'	'Sangonomiya_Kokomi'
5	'Alhaitham'	'Baizhu'	'Kujou_Sara'	'Barbara'
6	'Alhaitham'	'Baizhu'	'Kujou_Sara'	'Sangonomiya_Kokomi'
7	'Alhaitham'	'Baizhu'	'Kukl_Shinobu'	'Furina'
8	'Alhaitham'	'Baizhu'	'Kukl_Shinobu'	'Xingqiu'
9	'Alhaitham'	'Baizhu'	'Kukl_Shinobu'	'Candace'
10	'Alhaitham'	'Baizhu'	'Kukl_Shinobu'	'Mona'
11	'Alhaitham'	'Baizhu'	'Lisa'	'Barbara'
12	'Alhaitham'	'Baizhu'	'Lisa'	'Sangonomiya_Kokomi'
13	'Kirara'	'Baizhu'	'Beidou'	'Kamisato_Ayato'
14	'Kirara'	'Baizhu'	'Beidou'	'Nilou'
15	'Kirara'	'Baizhu'	'Beidou'	'Tartaglia'
16	'Kirara'	'Baizhu'	'Beidou'	'Yelan'
17	'Kirara'	'Baizhu'	'Beidou'	'Neuvillette'
18	'Kirara'	'Baizhu'	'Cyno'	'Furina'
19	'Kirara'	'Baizhu'	'Cyno'	'Xingqiu'

Q10 (Bonus Federated Query): Get the voice actor names from Wikidata for characters.

Competency 1 × Competency 2 × Competency 3 × Competency 4 × Competency 5 ×

Competency 6 × Competency 7 × Competency 8 × Competency 9 × Competency 10 × ⊕

```

1 PREFIX ontology: <http://www.genshin.org/ontology/>
2 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4
5 # Q10: Get the voice actor names from Wikidata for Characters.
6 SELECT ?characterName ?voiceActor ?voiceActorName
7 WHERE {
8   ?character a ontology:Character;
9     rdfs:label ?characterName.
10  SERVICE <https://query.wikidata.org/sparql> {
11    ?character wdt:P725 ?voiceActor.
12    ?voiceActor rdfs:label ?voiceActorName.
13    FILTER(LANG(?voiceActorName) = "en") .
14  }
15
16
17
18 }
```

Table Raw Response Pivot Table Google Chart Download as

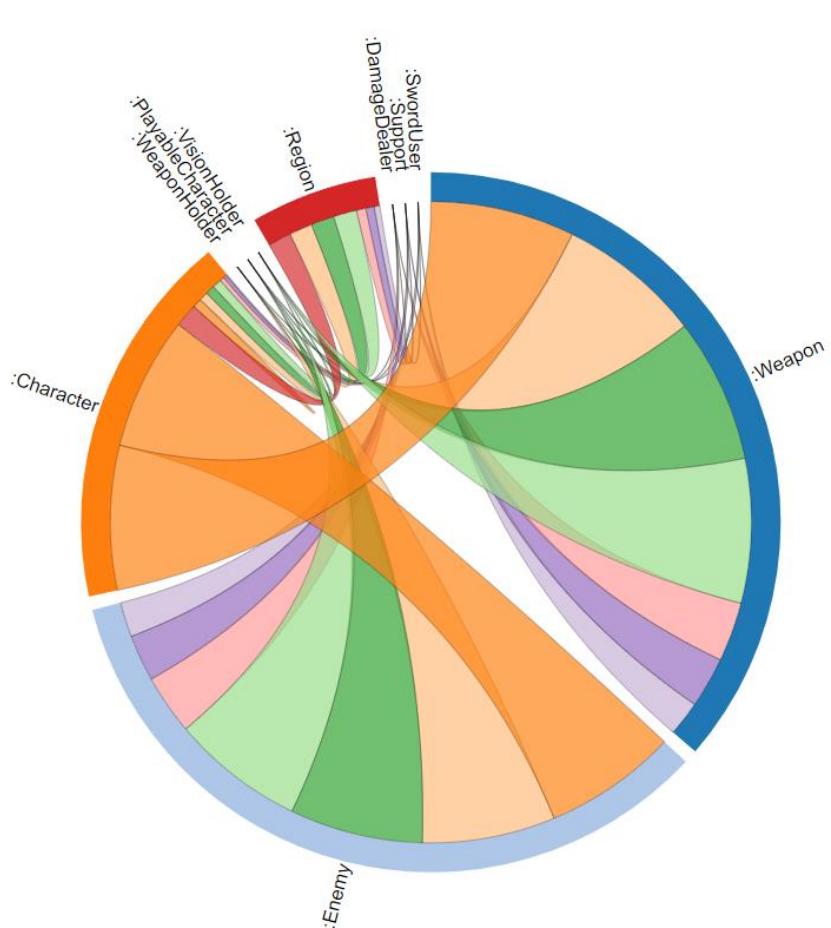
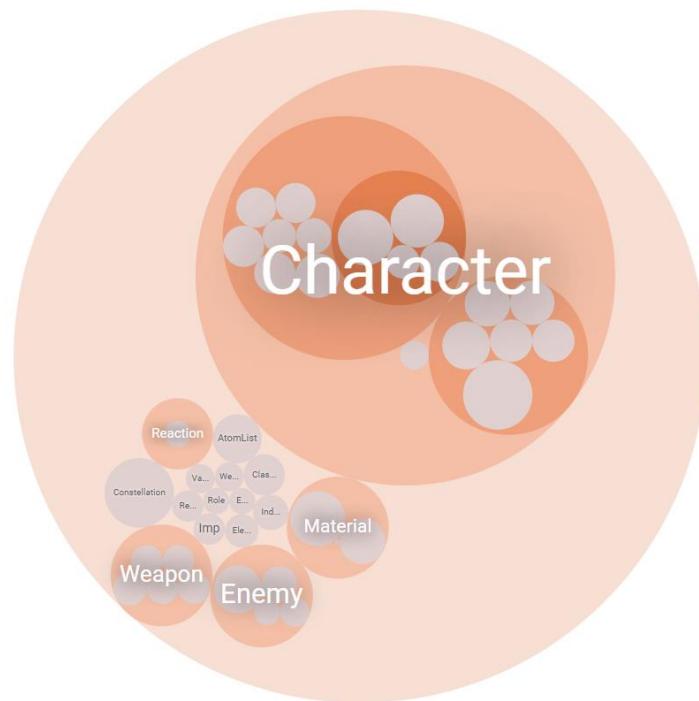
Filter query results Showing results from 1 to 60 of 60. Query took 36s, moments ago.

characterName	voiceActor	voiceActorName
"Fischl"	http://www.wikidata.org/entity/Q44552	"Maaya Uchida"@en
"Raiden_Shogun"	http://www.wikidata.org/entity/Q49552	"Miyuki Sawashiro"@en
"Shenhe"	http://www.wikidata.org/entity/Q49566	"Ayako Kawasumi"@en
"Lisa"	http://www.wikidata.org/entity/Q237371	"Rie Tanaka"@en
"Thoma"	http://www.wikidata.org/entity/Q961574	"Masakazu Morita"@en
"Keqing"	http://www.wikidata.org/entity/Q665398	"Eri Kitamura"@en
"Yae_Miko"	http://www.wikidata.org/entity/Q871452	"Ayane Sakura"@en
"Xiao"	http://www.wikidata.org/entity/Q957795	"Yoshitsugu Matsukawa"@en
"Qiqi"	http://www.wikidata.org/entity/Q1022023	"Yukari Tamura"@en
"Nahida"	http://www.wikidata.org/entity/Q1022023	"Yukari Tamura"@en
"Yoimiya"	http://www.wikidata.org/entity/Q1153353	"Kana Ueda"@en
"Diluc"	http://www.wikidata.org/entity/Q1139912	"Kensho Ono"@en

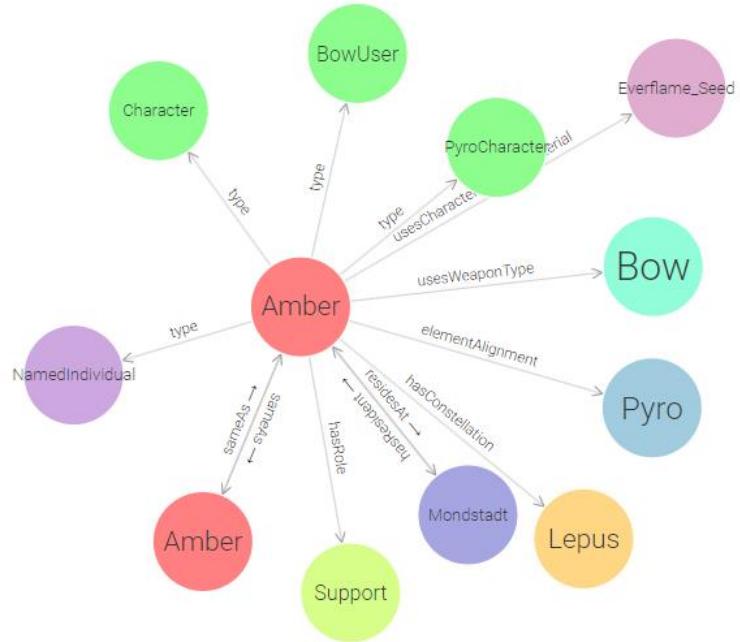
Run

Press Alt+Enter to autocomplete

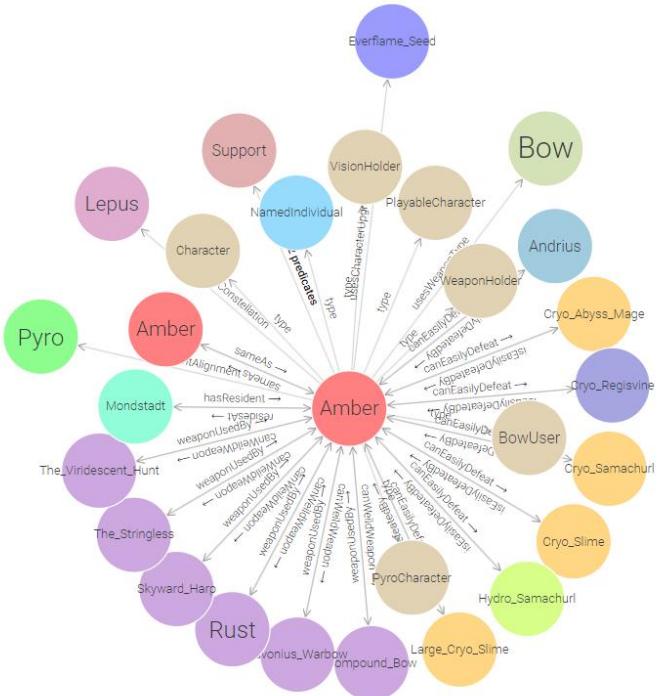
Visualizations

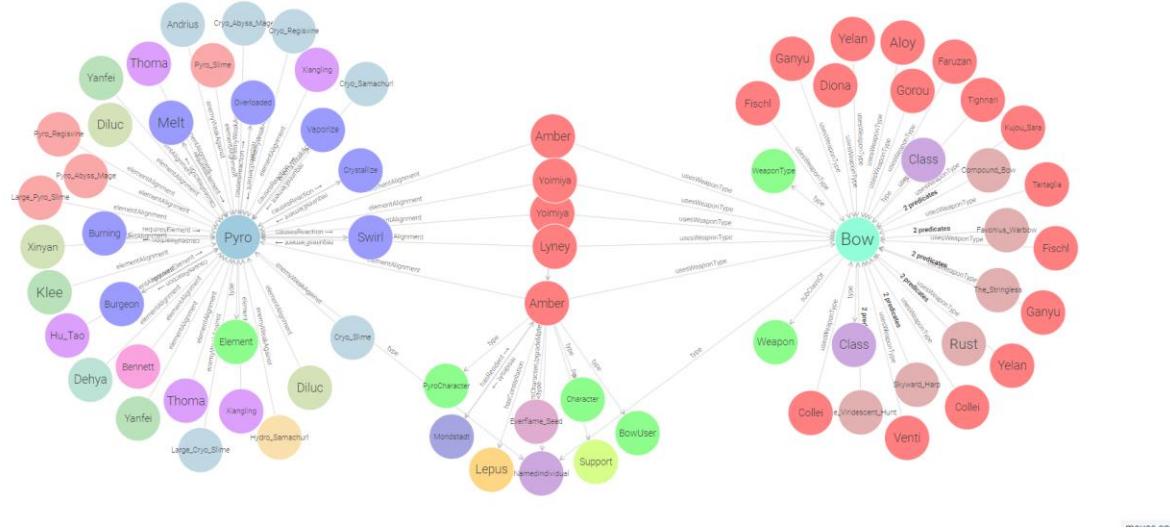


Before adding Inference

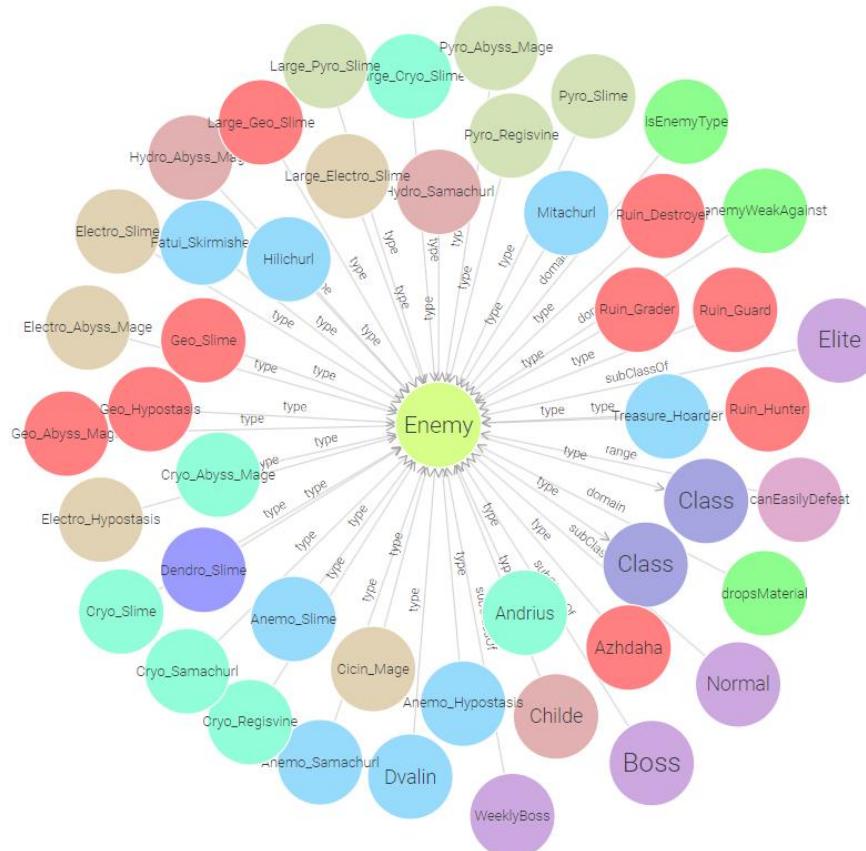


After adding Inference (The second Amber is its Wikidata counterpart, I used owl:sameAs for it)

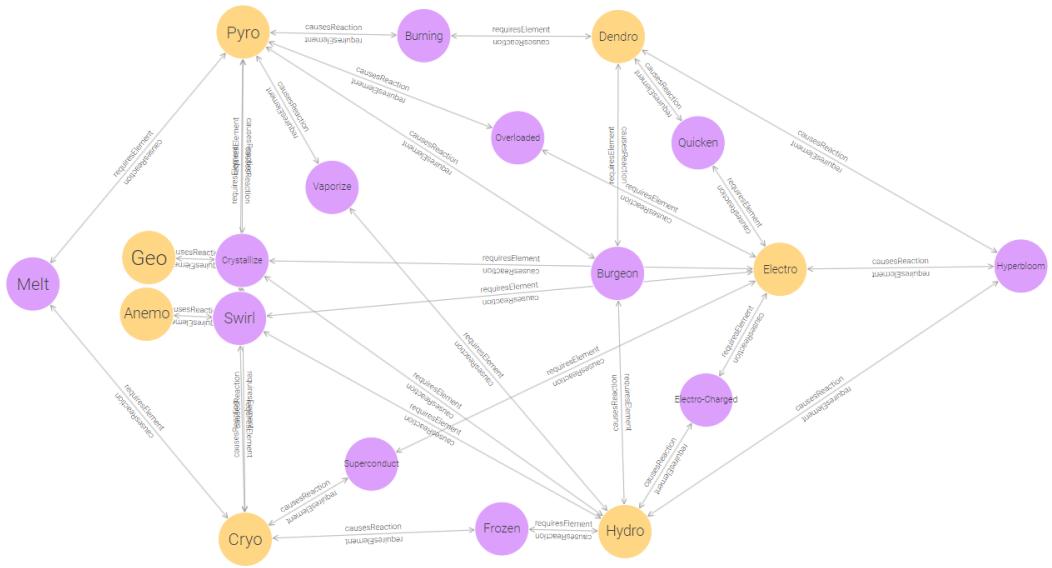




Over here, we can see Amber, Yoimiya and Lyney being Bow character who have a Pyro vision/



Here we can see the individuals related to `Enemy` class, the subclasses and domain/ranges for object properties where `Enemy` class is used in



These are the relations between Elements and Reactions

Reflect

This was honestly a fun project with a lot of learning involved. Perhaps the most important thing we learned was how to properly represent data in a meaningful way. There were a lot of times and a lot of wasted hours where we had to redesign our ontology because the relations we tried to achieve between individuals just weren't making any sense. But at the end, we managed to create a pretty accurate representation of how the game works in the form of a graph.

Converting unstructured data like this to linked data (especially when SWRL rules were used) did have a lot of benefits. Whenever we played the game and a new boss/weekly boss would come out, we would try to figure out which characters would be good against it and whenever a new character was added, we would try to figure out new teams we could make using that character. Thanks to the inferencing power of SWRL rules and OWL itself -we were able to automate this questions and stored all of the direct information necessary to answer it in our graph :).

Overall there is still a lot left in this game's graph development. If we can create a 1 to 1 model of the game using these technologies then theorycrafting team combinations and finding the most optimal teams to beat spiral abyss would be just automated within these graphs. In the future, I'll probably revisit this a few more times, (I'll need to convert this to an LPG first though) because this was more of a passion project than a project being done for a course.

Thank you so much for allowing our team to choose this topic, I know you must have been a bit worried because we even mentioned in the beginning that we have 0 tabular data to work on this project idea but when there's creative freedom given, there will always be a way.

Farewell