

Snake Game an Intelligent agent

Bharath G. Marulasiddappa

Bxg141330

Problem Description:

The goal of the project “Snake Game” is to allow snake to find the object using search mechanism. A* with heuristics would be used as a search algorithm. The destination of the object keeps changing (is implemented by a random function). Snake searches the object using A* with heuristics search approach. Also I have kept track of the nodes in search space and logged it on the console.

Knowledge Base (KB):

There's a start point and a goal/end-point that the snake must reach. There are two kinds of knowledge base

1. Knowledge on the objects that the snake encounters.
2. Knowledge on the path that the snake can take from start-point to goal.

All these are encoded in KB using OWL. Knowledge on the objects is considered as the unfavorable situation and knowledge on the path is considered as the favorable situation for the snake. So in order, for the snake to reach the destination it has to avoid unfavorable situation and must follow the favorable situation to reach the Goal.

Proposed Solution:

Each node is a point at which the snake might find a favorable/unfavorable object. Each directed edge is the path the snake can take from one node to another. The objective of the game would then be - What are all the paths the snake can take in the DAG, while using the knowledge from the Ontology that tells the snake what paths are favorable and what are not.

Implementation Details:

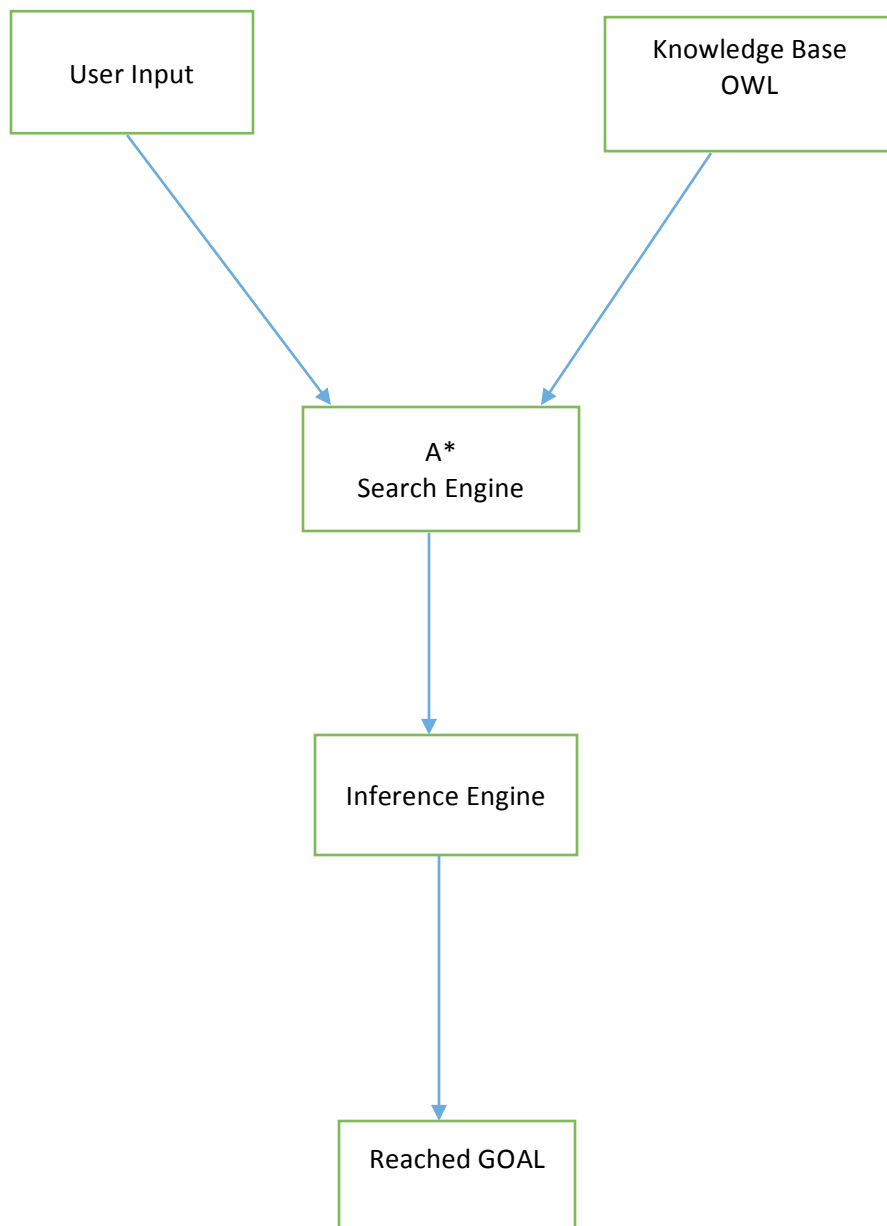
Designed a Web App.

Knowledge base OWL + XML, A* search algorithm is implemented in JavaScript, User Interface: JS, CSS and Html5 (web app). Gradle is a build automation tool built for Web Services (explained later in Challenges faced section). Implemented Web Services in Java. Jersey framework is used. Service hosted on Jetty Server.

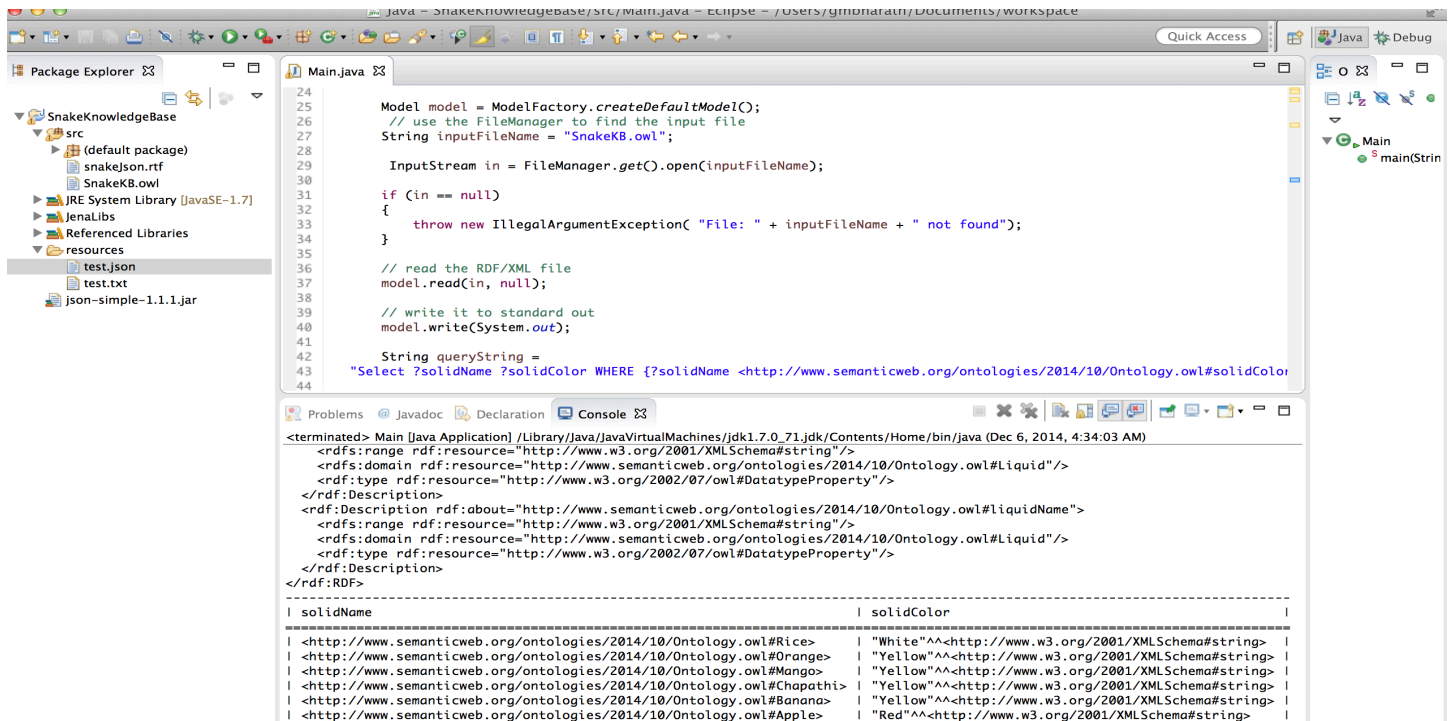
Challenges faced:

- 1) One of the major challenges faced was getting used to SPARQL. Reason because SPARQL was not like an SQLite. I had to dig deep to understand SPARQL.
- 2) Apache Jena is NOT compatible with Java 1.6 SDK. It's compatible with Java 1.7 and higher versions. Challenging because, error in the eclipse console was misleading. Realized only later after reading through README file.
- 3) Major challenge faced was reading the values from KB. Realized only later that there are no built in libraries to read the data through JavaScript. Built a web service to read the data from Knowledge Base. Please check the project about the implementation.
- 4) Web service (<http://localhost:8080/Bharath/rest/hello>) is hosted using Gradle (a build automation tool) on Jetty server. For Gradle to run, we need to set \$PATH the Gradle Path in ~/.bashrc. Terminal actually runs bash_profile whenever Terminal is launched. So its better to set \$PATH in bash_profile on Mac OS X.

Architectural Diagram:



Java Application Snapshot

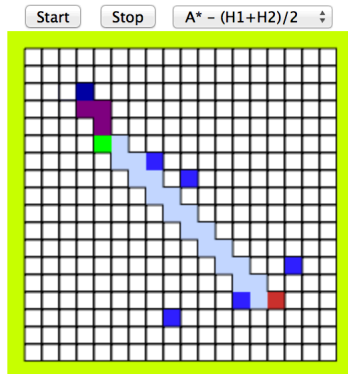


Future Enhancements:

Various search algorithms like DFS, BFS etc. can be implemented and see how snake searches over different KB values.

Examples:

User can give multiple inputs with different heuristics. Snake take different paths depending up the heuristics and KB values.



Programming Tools:

- 1) Eclipse IDE for Java EE.
- 2) Protégé for knowledge base development
- 3) Apache Jena API for importing knowledge base in Java.
- 4) User Interface: JavaScript, CSS, HTML 5, jQuery
- 5) Gradle a build analysis tool
- 6) Jetty server

Terminal Screenshot

```

    <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Solid"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Dew">
    <Ontology:liquidColor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">White</Ontology:liquidColor>
    <Ontology:liquidName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Dew</Ontology:liquidName>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Liquid"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Sprite">
    <Ontology:liquidColor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">White</Ontology:liquidColor>
    <Ontology:liquidName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Sprite</Ontology:liquidName>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Liquid"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Solid">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Food"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Liquid">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Food"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#ThumbsUp">
    <Ontology:liquidName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ThumbsUp</Ontology:liquidName>
    <Ontology:liquidColor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Brown</Ontology:liquidColor>
    <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Liquid"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Mango">
    <Ontology:solidColor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Yellow</Ontology:solidColor>
    <Ontology:solidName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Mango</Ontology:solidName>
    <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Solid"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Food">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#liquidColor">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Liquid"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#liquidName">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/10/Ontology.owl#Liquid"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
</rdf:RDF>
log4j:WARN No appenders could be found for logger (com.hp.hpl.jena.sparql.mgt.ARQMgt).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
> Building 75% > :jettyRun > Running at http://localhost:8080/Bharath

```