Class Diagram
Notes on Packages

package agent

The agent package implements the agent model of Russell & Norvig. It is problem domain independent: these interfaces will be implemented by problem domain classes and agents. It is provided and you do not need to alter this package.

package search

The search package contains all classes related to generic search algorithms. It is problem domain independent: searches of different problem domains can be carried out by passing appropriate implementations of State and NodeInfo to the search algorithms. It is partially provided, and you are required to complete the search algorithms.

package wordChess

The wordChess package is an implementation of the “world” for this particular problem domain. It defines the environment in which the agents are operating, what percepts they will receive, and what operations they can perform on the environment. It is thus problem domain dependent: different problem domains will replace this package with different classes. For the word chess problem domain, you are required to implement this package.

package wordPlayer

The wordPlayer package is where your agent(s) (or players) belong. The agents implement the Agent interface, and make use of the search package as necessary. They should setup appropriate searches by passing suitable implementations of State and NodeInfo, and run the searches as required when percepts are passed to their getActions() methods. The agents will thus use domain specific information. You are required to implement this package. (You could think of the search package along with State and NodeInfo classes as forming a working representation of the world in the agent’s “mind”. The wordChess package on the other hand represents the “physical” world.)
General Comments

The package names wordChess and wordPlayer are suggestions - you may use different names. What is important is the division of responsibility between the packages. This provides a high degree of modularity. Notice that a different package of players could be substituted without “breaking” the other packages. Similarly a different problem domain and agents for that domain could be developed by replacing the wordChess and wordPlayer packages. This will be done in the subsequent laboratories.

Aside from the programming benefits of modularity, the packages also match the conceptual structure. Whereas the wordChess package represents the “physical” world, the search package along with State and NodeInfo classes can be thought of as forming a working representation of the world in the agent’s “mind”. In a similar way humans picture or imagine the world and plan their actions before physically acting.