Find information about the different agents mentioned above, also include agents, mentioned in the lecture.

Software agents

A software agent is a program that lives inside a computer environment and can take actions on for other computer programs or for the user itself. It is not considered intelligent since it follows strict task list that it tries to complete and then simply lays dormant.

Intelligent agents

In AI these types of agents are autonomous that observe and reacts to environment through their sensors in a constant activity of achieving their goals.

Russel & Norvig proposed five different classes to sort agents in based on their perceived intelligence and capability.

- *Simple reflex.* Act only on the current state that they are perceiving and do not consider their past history and actions. Only work in fully observable environments.
- *Model-based reflex.* Work in partially observable environments and works based on how it experiences and perceives the world around it.
- *Goal-based.* Upgrade to the model based ones, since they try to achieve a goal based on search and planning algorithms.
- *Utility-based.* A utility is introduced to the agent, so it will try to maximize a score for its goal. By doing so, it can choose different actions based on it.
- *Learning.* This one is the most dynamic and bases it knowledge about learning about its unknown environment. I think this one is the most interesting agent since it starts out stupid without knowing what it can do but surely and slowly learns what it should do based on the environment around it. Like a baby.

I once wrote a rather interesting agent in F# that could move around an unknown world without knowing what it should or could do. For every move it made, it gained some value based on the action taken and stored the value of the move in a brain. The value was a combination of the state it was in, the state it ended up in, the world around it and the world around it after taking the step. By memorizing these things, each state was valued by this function

```
Var newValue = (1.0 - alpha) * value + alpha * ( exp.Reward + gamma * vNext)
```

Where the **bold** part is a former similar state and the value it generated for that state. The *italic* part is the new value in a similar state that is added to the old memory to improve the step taken. The most interesting part about this method was so called epsilon decision learning, but it introduced the concept of before taking a move based on past experience, in 5% of all moves it would rather pick a random move rather than using its brain which improved its average score. Why is that?

This is just related to the human behavior. Learning new things requires us to step out of our comfort zone and trying out new things whether they are good or bad. If they are good, we gain value from those decisions. If they are indeed bad, we do burn ourselves and do not do them again. The project is included in the assignment.

Meta-agents

Just like Meta data is information about data, Meta agents are agents for agents. They monitor them in a way by delegating tasks, track performance and processes they are carrying out. Furthermore,

they can help them by changing their states and provide best routing path based on observed knowledge provided by other agents. For an example, meta agents are used in distributed active replication of servers and group communication. You can think of it as the HiveMind from Starcraft 1.

Other agents from lecture

- **Negotiation / bargain.** Used in auctions to bid on items for salve. They are often based on negotiation set, a protocol and a rule.
- **Rational.** Have some sense of judgement and try to plan their actions based on expected outcomes.
- **BDI.** Have certain beliefs of how the world works, desires to accomplish goals and intensions that the agent would like to act upon.
- **Mobile.** Move around the environment and can adjust to new environments. Web crawlers and more.
- **Autonomous.** Is an agent that can act on its own based on the world perceived. Robots, cars and other things that are really cool ☺

MAS and environments for the agents.

Multi-agent systems focus on decentralized knowledge and problem solving tasks. The system is a combination of agents that are loosely coupled in a network that interact together to try so solve a given task.

Question 1

Explain why a MAS (multi agent system) for an army of drones is better rather than having multiple drones all with their own single isolated agent. Write a short example where MAS comes in handy.

An agent has a certain task or role to fulfill, while a MAS is often bound to a certain objective. By having multiple individual agents trying to complete an objective together while stile fulfilling their role they need to communicate, negotiate and cooperate. MAS usually provides the framework for that architecture.

An example of this can be demonstrated as an army of drones attacking an enemy town. The town has anti air guns, tanks, soldiers and other defense, so some drones are specially designed to counter those obstacles. If you had 100 of drones flying, they would first have to communicate so they wouldn't crash into each other. Drone A has the task to eliminate tanks while drone B has the task to disarming soldiers. If drone A is positioned on the left side and sees a soldier, it could notify drone B about that he needs to fulfill his task over there.

Question 2

What are the most important aspects of MAS systems and how are they beneficial for the system?

Two of the most unique factors of MAS is that the control is decentralized and the knowledge is distributed. By relying on such factors, one might gain faster problem solving, more flexibility and increased reliability of its systems.

Question 3

Classify each of the following objects by being either Software agents or Intelligent agents.

- Help wizards **Software**
- Camera that centers a specific person in every shot **Intelligent**
- A quest giving NPC in a computer game **Software (if static)**
- Cleaning robot that cleans your house **Intelligent**
- The animated paper-clip agent in Microsoft Office **Software**

Question 4

Provide and discuss a clear difference between a human, a robot and an agent software.

A human has eyes, ears and other organs for sensors and hands legs, mouth, and other body parts to move around.
A robotic agent might have some cameras for vision or infrared range finders for sensors and various motors to move from place to place.
A software agent receives keyboard pushes, files and network data as sensory inputs and takes action in its environment by displaying relevant info on the screen, writing to the files or sending network acknowledgement.

Question 5

You are opening up a store that sells shoes. There are certain tasks you need to take care of to help your customers out, both in the store and on your website. Some of these tasks are suggesting a shoe based on customer preferences, finding the lowest price of desired shoe style, help the customer put the shoe on, letting the customer know if some shoe is out of stock and when it will arrive and finally make sure the reception where customers can come pick up their shoes always looks nice and tidy.

Which cases would you let an agent handle and which cases would you let the human handle? Justify your answer!

The agent should of course take care of suggesting shoes, finding the lowest price and letting the customer know if the shoes are out of stock if it could live inside the systems server. It could access databases like user preferences, use cookies if they exists or access their Facebook or Gmail info if they allow that. If the agent was a robot that could both help the user get in the shoe or clean the reception, then we would definitely make it an agent as well.