**Name:** Bushra Ashraf Bhatti

**Reg\_no:** BSE-2023-015

**Submitted to:** Dr. Irum Matloob

**Graded Activity 3**

**Exercise 2.5 (PEAS-exercise)**

For each of the following activities, give a PEAS description of the task environment and characterize it in terms of the properties listed in Section   
- Playing soccer.  
- Exploring the subsurface oceans of Titan.  
- Shopping for used AI books on the Internet.  
- Playing a tennis match.  
- Practicing tennis against a wall.  
- Performing a high jump.  
- Knitting a sweater.  
- Bidding on an item at an auction.

1. **Playing soccer:**

An intelligent agent (player) plays soccer with the goal of scoring goals and preventing the opponent from scoring, while cooperating with teammates.

1. **Exploring the Subsurface Oceans of Titan:**

A robotic submarine explores Titan’s subsurface oceans to collect data about its composition, environment, and potential for life.

1. **Shopping for Used AI Books on the Internet**

An intelligent shopping agent searches online stores to find and purchase used AI books at the best price and condition.

1. **Playing a Tennis Match**

A player competes against an opponent in a tennis match, aiming to win by strategically scoring points.

1. **Practicing Tennis Against a Wall**

A tennis player practices alone by hitting the ball against a wall repeatedly to improve control and consistency.

1. **Performing a High Jump**

An athlete runs and jumps over a horizontal bar, trying to clear it without knocking it down.

1. **Knitting a Sweater**

A person uses knitting needles and yarn to create a sweater by following a specific pattern and sequence of stitches.

1. **Bidding on an Item at an Auction**

A bidder competes with others to win an item by offering the highest price within a limited time or until all others stop bidding.

**PEAS and Environment Properties:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Performance Measure (P)** | **Environment (E)** | **Actuators (A)** | **Sensors (S)** | **Environment Properties** |
| **1. Playing Soccer** | Number of goals, teamwork, defense, match win | Soccer field, weather, ball, players, referee | Legs, head, body | Eyes, ears, body balance | Partially observable, multi-agent, dynamic, continuous, stochastic |
| **2. Exploring the Subsurface Oceans of Titan** | Data collected, navigation accuracy, mission success | Subsurface ocean, temperature, pressure, terrain | Thrusters, robotic arms, drills | Cameras, sonar, pressure, temperature sensors | Partially observable, single-agent, dynamic, continuous, stochastic |
| **3. Shopping for Used AI Books Online** | Best deal (price, quality, delivery time) | Internet stores, listings, sellers | Mouse clicks, keyboard input | Display screen, reviews, prices | Fully observable, single-agent, static, discrete, deterministic |
| **4. Playing a Tennis Match** | Points scored, match won | Tennis court, opponent, weather, ball | Arms, legs, racket | Eyes, ears, touch | Partially observable, multi-agent, dynamic, continuous, stochastic |
| **5. Practicing Tennis Against a Wall** | Number of successful returns, consistency | Wall, ball, racket, surroundings | Arms, legs, racket | Eyes, touch, balance | Fully observable, single-agent, semi-dynamic, continuous, deterministic |
| **6. Performing a High Jump** | Height achieved, bar clearance | Track, bar, ground, weather | Legs, arms, body | Eyes, balance | Fully observable, single-agent, static, continuous, deterministic |
| **7. Knitting a Sweater** | Accuracy, neatness, completion time | Yarn, needles, pattern | Hands, fingers | Eyes, touch | Fully observable, single-agent, static, discrete, deterministic |
| **8. Bidding at an Auction** | Win item at lowest price | Auction platform, bidders, auctioneer | Mouse clicks, voice (bids) | Display screen, price updates | Partially observable, multi-agent, dynamic, discrete, stochastic |

**(Exercise 2.6)**

For each of the following activities, give a PEAS description of the task environment and characterize it in terms of the properties listed in Section   
- Performing a gymnastics floor routine.  
- Exploring the subsurface oceans of Titan.  
- Playing soccer.  
- Shopping for used AI books on the Internet.  
- Practicing tennis against a wall.  
- Performing a high jump.  
- Bidding on an item at an auction.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Performance Measure (P)** | **Environment (E)** | **Actuators (A)** | **Sensors (S)** | **Environment Properties** |
| **1. Performing a Gymnastics Floor Routine** | Score from judges based on accuracy, creativity, balance | Gymnastics floor, judges, lighting | Arms, legs, body | Eyes, balance, motion sensors | Partially observable, single-agent, dynamic, continuous, stochastic |
| **2. Exploring the Subsurface Oceans of Titan** | Amount and accuracy of data collected, successful navigation | Subsurface ocean, temperature, terrain, currents | Thrusters, robotic arms, drills | Cameras, sonar, temperature and pressure sensors | Partially observable, single-agent, dynamic, continuous, stochastic |
| **3. Playing Soccer** | Goals scored, teamwork, defense, match victory | Soccer field, ball, players, weather | Legs, head, body | Eyes, ears, touch, balance | Partially observable, multi-agent, dynamic, continuous, stochastic |
| **4. Shopping for Used AI Books Online** | Find best price, quality, delivery option | Online marketplace, sellers, listings | Mouse clicks, typing, selection | Screen display, reviews, prices | Fully observable, single-agent, static, discrete, deterministic |
| **5. Practicing Tennis Against a Wall** | Accuracy and consistency of hits | Wall, ball, racket, surroundings | Arms, legs, racket | Eyes, ears, balance | Fully observable, single-agent, semi-dynamic, continuous, deterministic |
| **6. Performing a High Jump** | Height achieved, successful clearance | Track, bar, ground, weather | Legs, arms, body | Eyes, balance, timing sense | Fully observable, single-agent, static, continuous, deterministic |
| **7. Bidding on an Item at an Auction** | Winning item at lowest possible price | Auction platform, bidders, auctioneer | Mouse clicks, bids, voice | Display screen, sound, price updates | Partially observable, multi-agent, dynamic, discrete, stochastic |

**(Exercise 2.7)**

Define in your own words the following terms: agent, agent function, agent program, rationality, autonomy, reflex agent, model-based agent, goal-based agent, utility-based agent, learning agent.

1. **Agent**

An agent is an entity that perceives its environment through sensors and acts upon that environment using actuators to achieve a specific goal.  
In other words, an agent observes what is happening around it and decides what actions to take to perform a task effectively.  
For example, a robot vacuum cleaner senses dirt and obstacles and moves accordingly to clean the floor.

1. **Agent Function**

An agent function is a conceptual or mathematical mapping from percepts (inputs received from the environment) to actions (outputs performed by the agent).  
It defines what the agent should do in every possible situation it can encounter.  
For instance, if a robot senses an obstacle, the agent function might map this percept to the action “turn left.”

1. **Agent Program**

An agent program is the practical implementation of the agent function in the form of code or instructions that run on the agent’s hardware.  
It takes percepts as inputs and produces actions as outputs.  
For example, the control software inside a self-driving car that processes sensor data and sends commands to the steering and brakes is an agent program.

1. **Rationality**

Rationality refers to the ability of an agent to make the best possible decision that maximizes its performance measure, given the knowledge and percepts it has at a particular time.  
A rational agent always tries to choose the most appropriate action that leads to the achievement of its goals.

1. **Autonomy**

Autonomy is the degree to which an agent’s behavior is determined by its own experience and internal decision-making rather than by external control or prior programming.  
An autonomous agent can operate independently, learn from its environment, and adjust its actions accordingly without continuous human supervision.

1. **Reflex Agent**

A reflex agent selects actions based solely on the current percept, ignoring the history of past percepts.  
It operates on simple condition–action rules, often known as “if–then” rules.  
For example, a thermostat that turns on the heater if the temperature falls below a certain value is a reflex agent.

1. **Model-Based Agent**

A model-based agent maintains an internal model or representation of the environment that helps it track unobserved aspects of the world.  
This model allows the agent to handle partially observable environments by remembering past information and predicting future states.  
For example, a robotic vacuum that remembers the layout of a room to clean more efficiently is model-based.

1. **Goal-Based Agent**

A goal-based agent chooses its actions by considering the goal or desired outcome it wants to achieve.  
It does not just react to the current situation but plans and evaluates different possible actions to find the best path to reach its goal.  
For example, a navigation system that calculates the shortest route to a destination works as a goal-based agent.

1. **Utility-Based Agent**

A utility-based agent evaluates possible actions by using a utility function, which measures how desirable a particular state or outcome is.  
It does not just aim to achieve a goal but also considers which action provides the highest satisfaction or benefit.  
For example, a self-driving car may choose a route that balances both travel time and passenger comfort.

1. **Learning Agent**

A learning agent is capable of improving its performance over time through learning from experience.  
It observes the results of its actions and modifies its behavior to perform better in the future.  
For example, an AI system that learns to recognize faces more accurately after being trained with more images is a learning agent.

**(Exercise 2.8)**

This exercise explores the differences between agent functions and agent programs.

**QUESTION NO 1:** **Can there be more than one agent program that implements a given agent function? Give an example, or show why one is not possible.**

**YES,** there can be more than one agent program that implements the same agent function.  
For example, in the given **Simple Reflex Agent**, the rule is:

If the square is “Dirty” → Clean it  
If the square is “Clean” → Move

This rule (agent function) could be implemented in many different ways in code — using if-else, a dictionary, or a switch statement.  
All these programs will still produce the same behavior (same actions for the same percepts).

**QUESTION NO 2:** **Are there agent functions that cannot be implemented by any agent program?**

**Yes**, some agent functions cannot be implemented by any agent program.  
This is because:

* There are infinitely many possible percept sequences (inputs),
* But only a finite number of programs that can be stored in a computer.

So, some functions would require infinite memory or computation, which no computer can handle.  
In our reflex agent example, the program can only clean a fixed number of squares — it cannot handle an infinite grid or remember infinite past percepts.

**QUESTION NO 3:** Given a fixed machine architecture, does each agent program implement exactly one agent function?

**Yes**, each agent program running on a fixed machine implements exactly one agent function.  
For example, the same reflex agent code will always do this:

“Dirty” → “Clean”  
“Clean” → “Move”

No matter how many times it runs, the output for the same percept will always be the same.

**QUESTION NO 4:** Given an architecture with n bits of storage, how many different possible agent programs are there?

The number of possible agent programs = **2ⁿ**.  
Each bit can be either 0 or 1, so if a machine has n bits, there are **2ⁿ** unique ways to arrange those bits.  
Example:

* If n = 4 → 2⁴ = 16 possible programs
* If n = 8 → 2⁸ = 256 possible programs

**QUESTION NO 5**: Suppose we keep the agent program fixed but speed up the machine by a factor of two. Does that change the agent function?

**No**, speeding up the machine does **not** change the agent function.  
The **behavior (input → output mapping)** stays the same — only the **speed** of execution changes.

In our example:

* Whether the reflex agent runs slowly or twice as fast, it will still clean when dirty and move when clean.
* Only the time it takes to do it will be different.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*