FPT UNIVERSITY

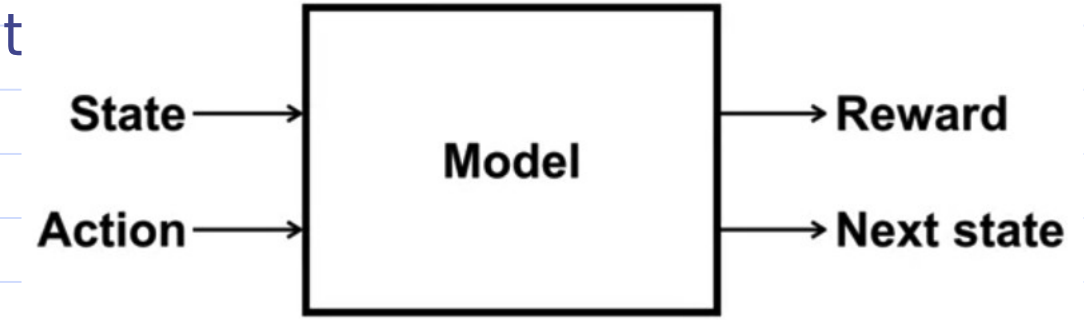
Define Model in Reinforcement Learning

Objectives

- Describe what a model is and how they can be used
- Classify models as distribution models or sample models
- Identify when to use a distribution model or sample model
- Describe the advantages and disadvantages of sample models and distribution models

What is a Model?

- ❑ Models are used to store knowledge about the dynamics.
- ❑ From a particular state in action, the model should produce a possible next state and reward.
- ❑ This allows us to see an outcome of an action without

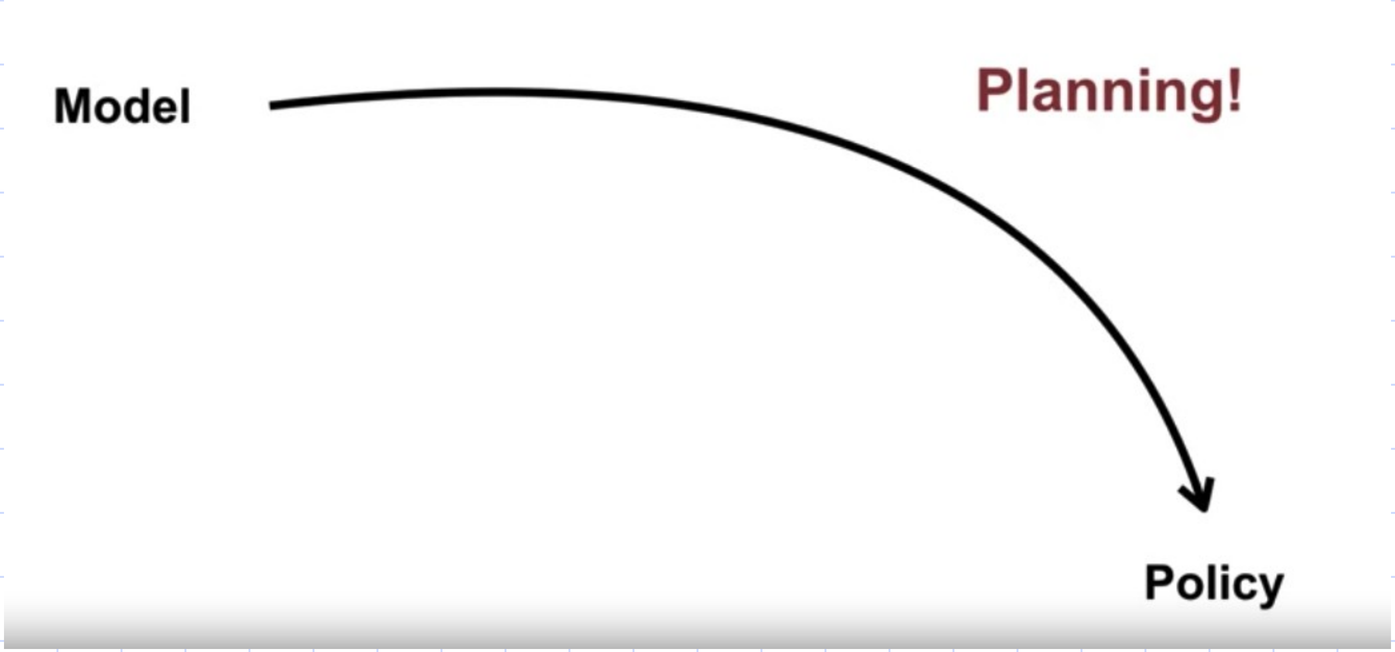


What is a Model?

- ❑ In the context of reinforcement learning, a model refers to a simplified representation of the environment that allows the agent to simulate and predict the outcomes of its actions without interacting directly with the real environment.
- ❑ In other words, a model provides a way for the agent to estimate the consequences of its actions before actually executing them.

What is a Model?

- □ A model allows for planning. Planning refers to the process of using a model to improve a policy

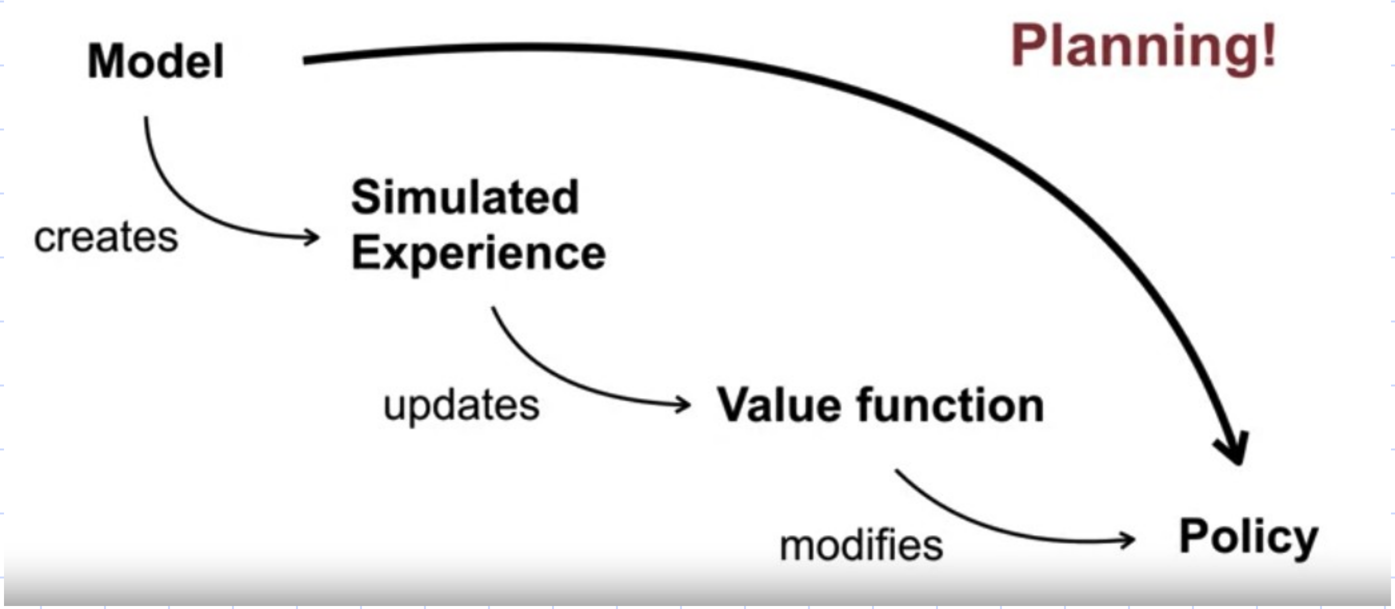


What is a Model?

- ❑ One way to plan with a model is to use simulated experience and perform value function updates as if those experiences happened.
- ❑ By improving the value estimates, we can make more informed decisions.
- ❑ Simulating experience improves the sample efficiency.
- ❑ The addition of simulated experience means the agent needs fewer interactions with the world to come up with the same policy

What is a Model?

- ❑ Models are used for planning



Model Types

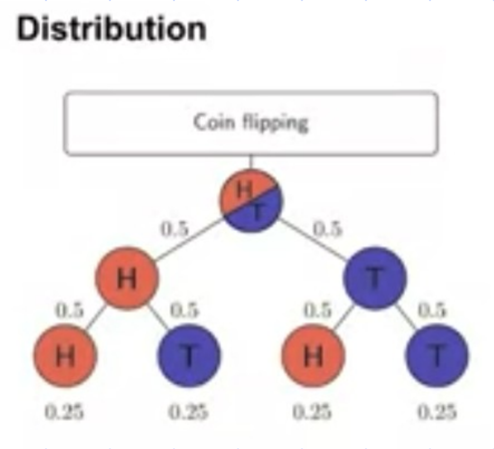
- ❑ Sample model, which produces an actual outcome drawn from some underlying probabilities.
  - ❑ For example, a sample model for flipping a coin can generate a random sequence of heads and tails.

Model Types

- ❑ Sample Model:
  - ❑ **Description**: A sample model provides specific samples or instances of the next state and reward for a given state-action pair.
  - ❑ **Implementation**: Instead of providing a probability distribution over all possible next states and rewards, a sample model generates actual samples based on the transition dynamics of the environment.
  - ❑ **Example**: In a sample model for a gridworld environment, when the agent takes an action in a particular state, the sample model directly provides a specific next state and reward based on the environment's transition probabilities.

Model Types

- ❑ Distribution model which completely specifies the likelihood or probability of every outcome.
  - ❑ In the coin flipping example, it would say that heads could occur 50 percent of the time and that tails could occur 50 percent of the time.
  - ❑ It could also produce th ence of heads and tails using this informa



Model Types

- Distribution Model:
  - **Description**: A distribution model provides a probability distribution over all possible next states and rewards for a given state-action pair.
  - **Implementation**: Instead of providing specific samples, a distribution model represents the uncertainty of the environment's dynamics by assigning probabilities to each possible outcome.
  - **Example**: In a distribution model for a gridworld environment, when the agent takes an action in a particular state, the distribution model specifies the probabilities of transitioning to each neighboring state and receiving each

Model Types

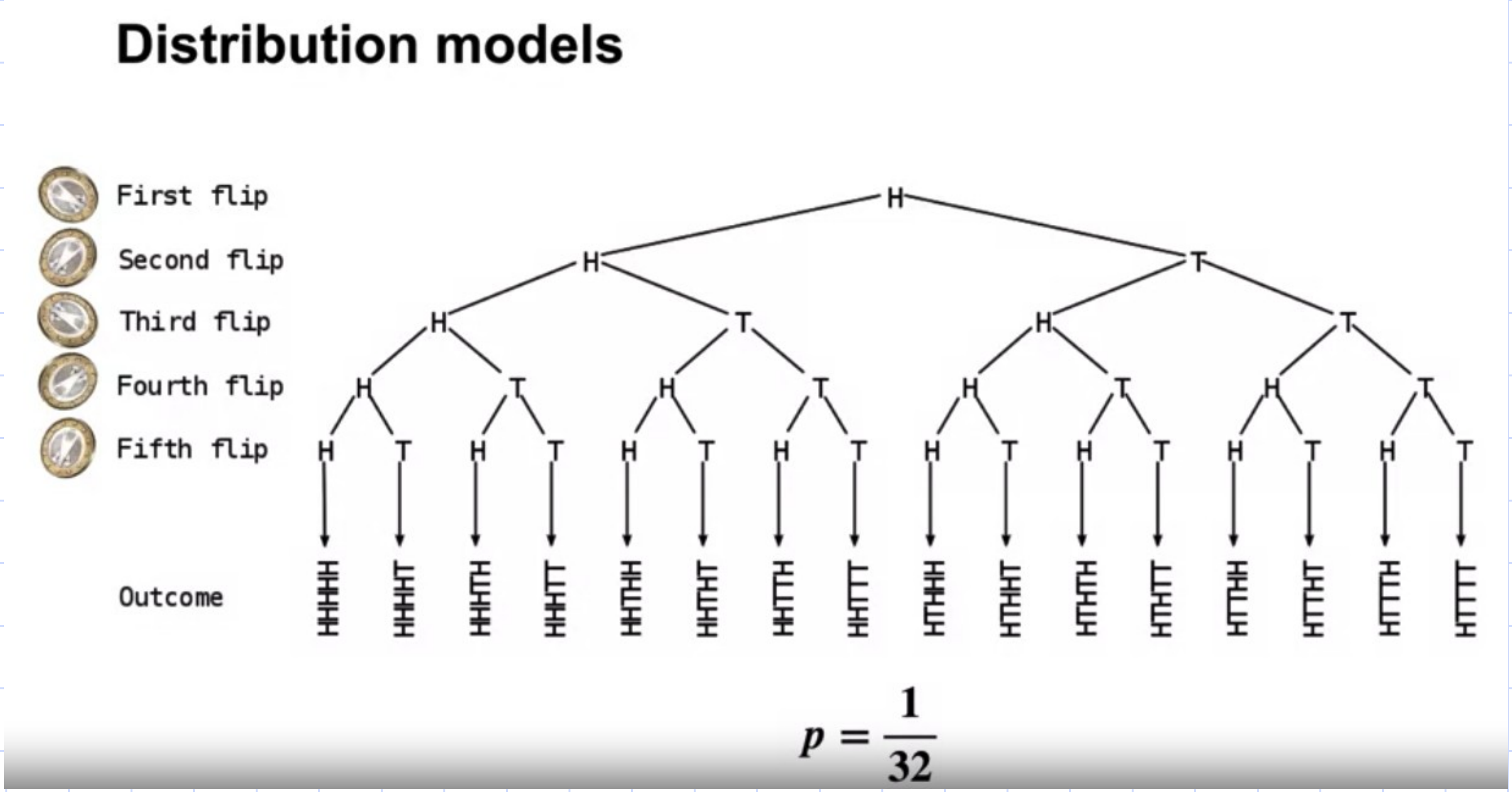
- Sample models can be computationally inexpensive because random outcomes can be produced according to a set of rules.
  - For example, to flip five coins, a sample model can randomly pick zero or one independently five times.
  - It only needs to produce a single outcome for each flip

Model Types

- Distribution models contain more information, but can be difficult to specify and can become very large.
  - The example for flipping five coin: the distribution model would enumerate every possible sequence of heads and tails you could get across the five coins and assign a probability to each sequence.
  - For this simple problem, that would consist of fully describing 32 possible outcomes.

Model Types

- ❑ Distribution models



Distribution models

$$p = \frac{1}{32}$$

Sample vs Distribution Models

- ❑ Sample models and distribution models are both types of models used in reinforcement learning to represent the environment's dynamics.
- ❑ Sample models provide specific instances of next states and rewards, while distribution models represent probability distributions over all possible outcomes.
- ❑ The choice between them depends on the characteristics of the environment and the specific requirements of the reinforcement learning problem.

Sample vs Distribution Models

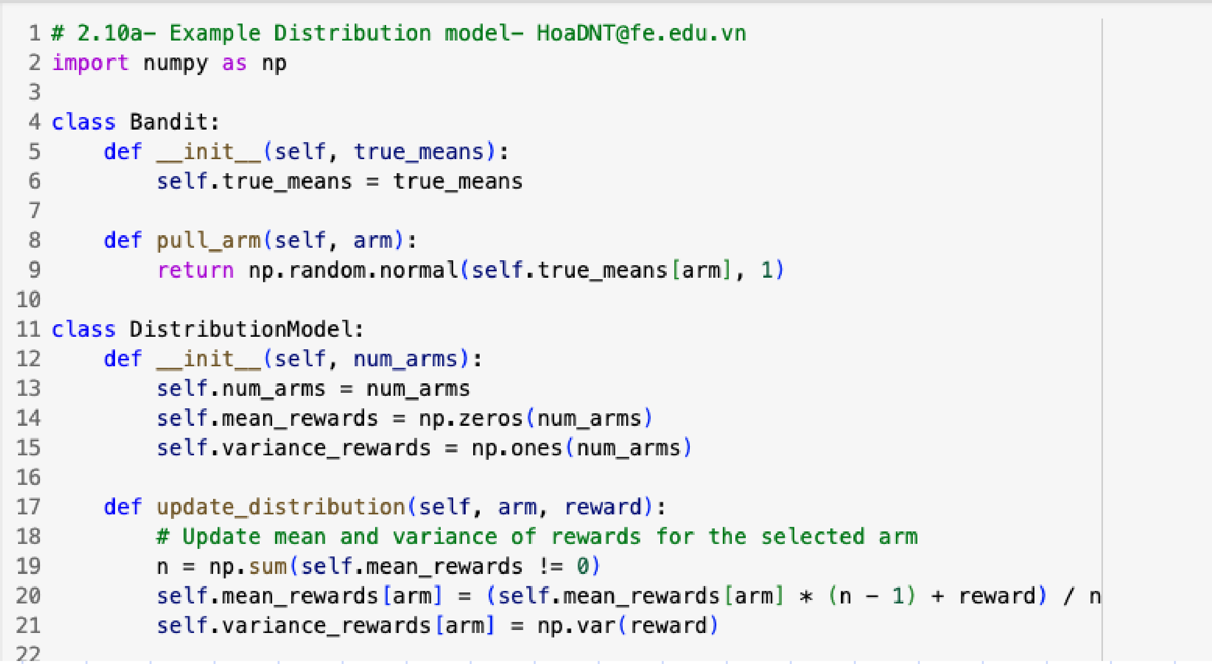
- **Representation**: Sample models represent specific instances of next states and rewards, while distribution models represent probability distributions over all possible outcomes.
- **Uncertainty**: Sample models do not explicitly capture uncertainty in the environment's dynamics, whereas distribution models provide a measure of uncertainty through probability distributions.

Sample vs Distribution Models

- **Efficiency**: Sample models can be more memory-intensive and less efficient to compute compared to distribution models, especially in environments with large state and action spaces.
- **Applicability**: Sample models are often used in deterministic environments or when exact samples are readily available, while distribution models are more suitable for stochastic environments or when uncertainty needs to be explicitly accounted for.

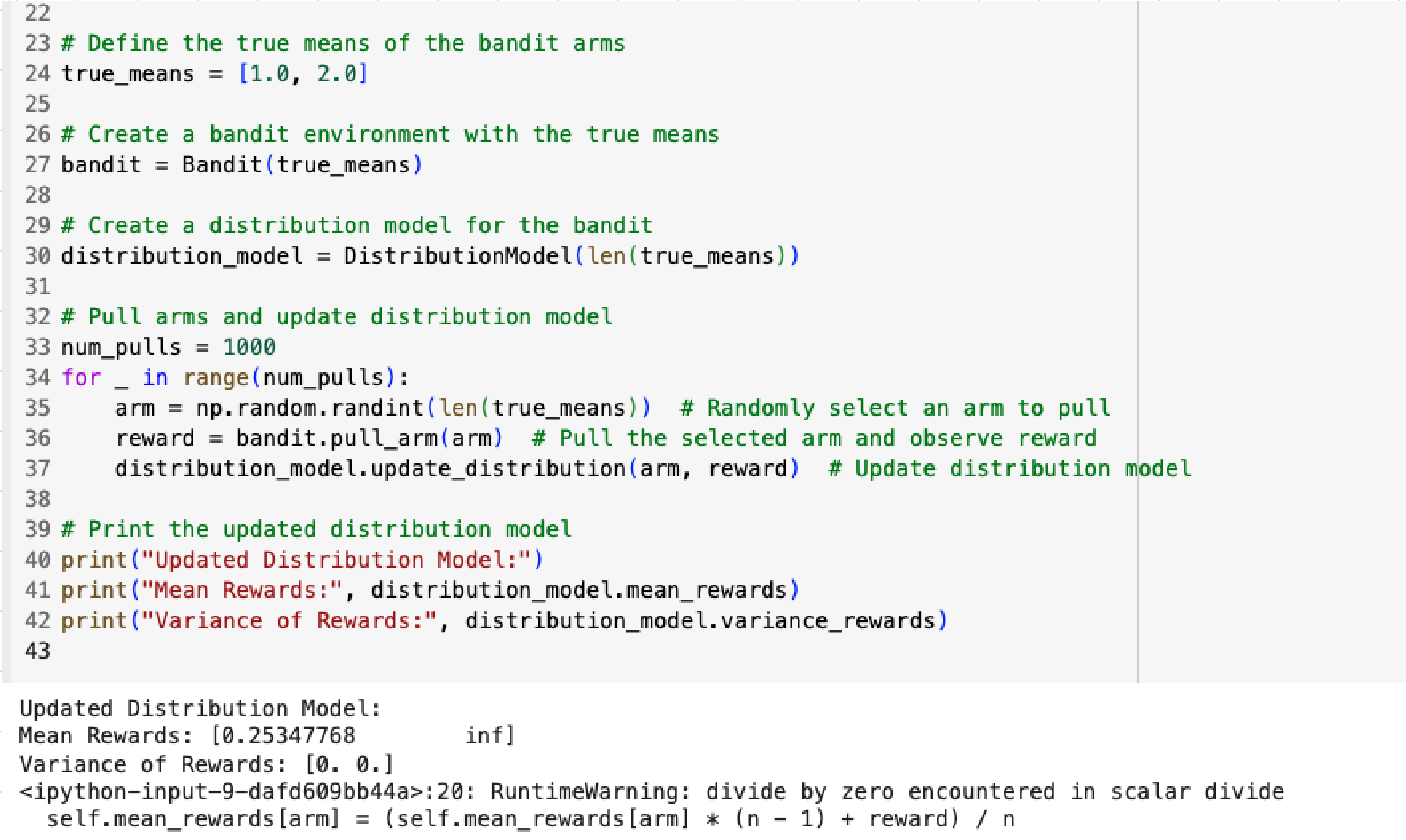
Distribution Models Example

- ❑ Example of a distribution model for a two-armed bandit problem to represent and update probability distributions of rewards



```
# 2.10a- Example Distribution model- HoaDNT@fe.edu.vn
import numpy as np

class Bandit:
    def __init__(self, true_means):
        self.true_means = true_means

    def pull_arm(self, arm):
        return np.random.normal(self.true_means[arm], 1)

class DistributionModel:
    def __init__(self, num_arms):
        self.num_arms = num_arms
        self.mean_rewards = np.zeros(num_arms)
        self.variance_rewards = np.ones(num_arms)

    def update_distribution(self, arm, reward):
        # Update mean and variance of rewards for the selected arm
        n = np.sum(self.mean_rewards != 0)
        self.mean_rewards[arm] = (self.mean_rewards[arm] * (n - 1) + reward) / n
        self.variance_rewards[arm] = np.var(reward)
```

Distribution Models Example

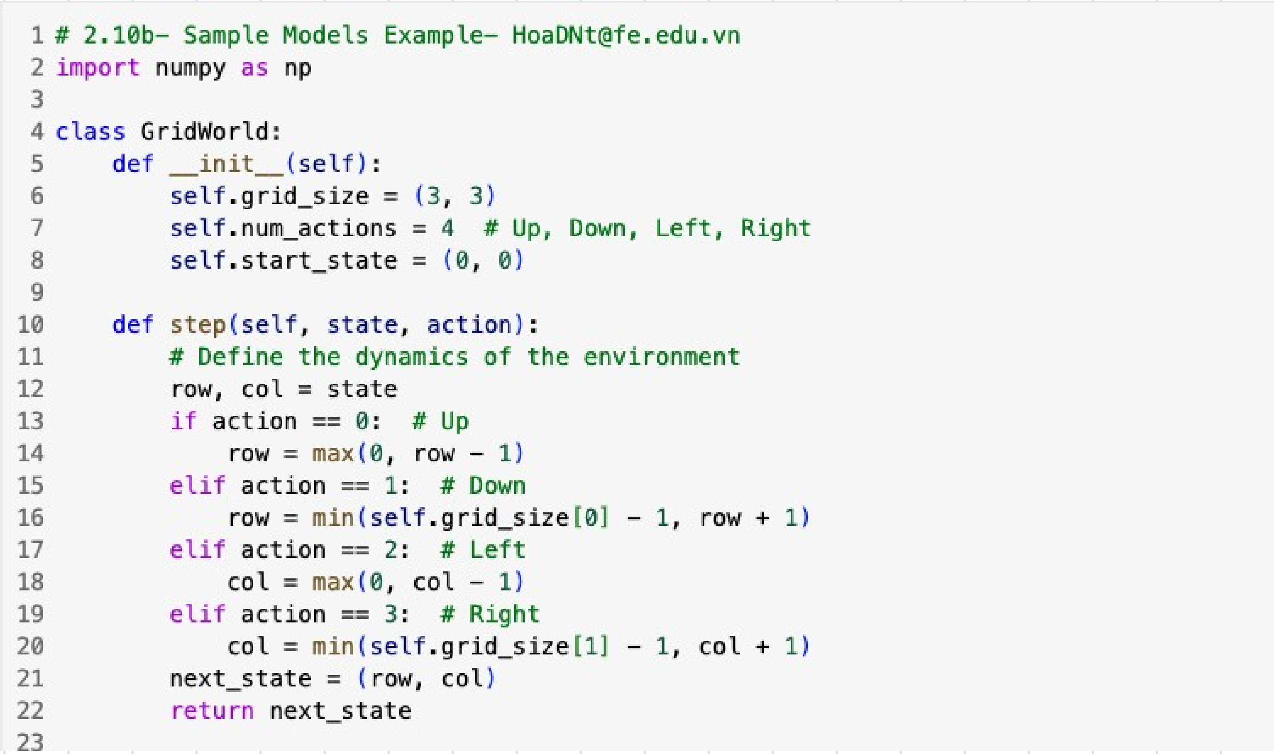


```python

# Define the true means of the bandit arms
true_means = [1.0, 2.0]

# Create a bandit environment with the true means
bandit = Bandit(true_means)

# Create a distribution model for the bandit
distribution_model = DistributionModel(len(true_means))

# Pull arms and update distribution model
num_pulls = 1000
for _ in range(num_pulls):
    arm = np.random.randint(len(true_means))  # Randomly select an arm to pull
    reward = bandit.pull_arm(arm)  # Pull the selected arm and observe reward
    distribution_model.update_distribution(arm, reward)  # Update distribution model

# Print the updated distribution model
print("Updated Distribution Model:")
print("Mean Rewards:", distribution_model.mean_rewards)
print("Variance of Rewards:", distribution_model.variance_rewards)

```

```
Updated Distribution Model:
Mean Rewards: [0.25347768        inf]
Variance of Rewards: [0. 0.]
<ipython-input-9-dafd609bb44a>:20: RuntimeWarning: divide by zero encountered in scalar divide
  self.mean_rewards[arm] = (self.mean_rewards[arm] * (n - 1) + reward) / n
```

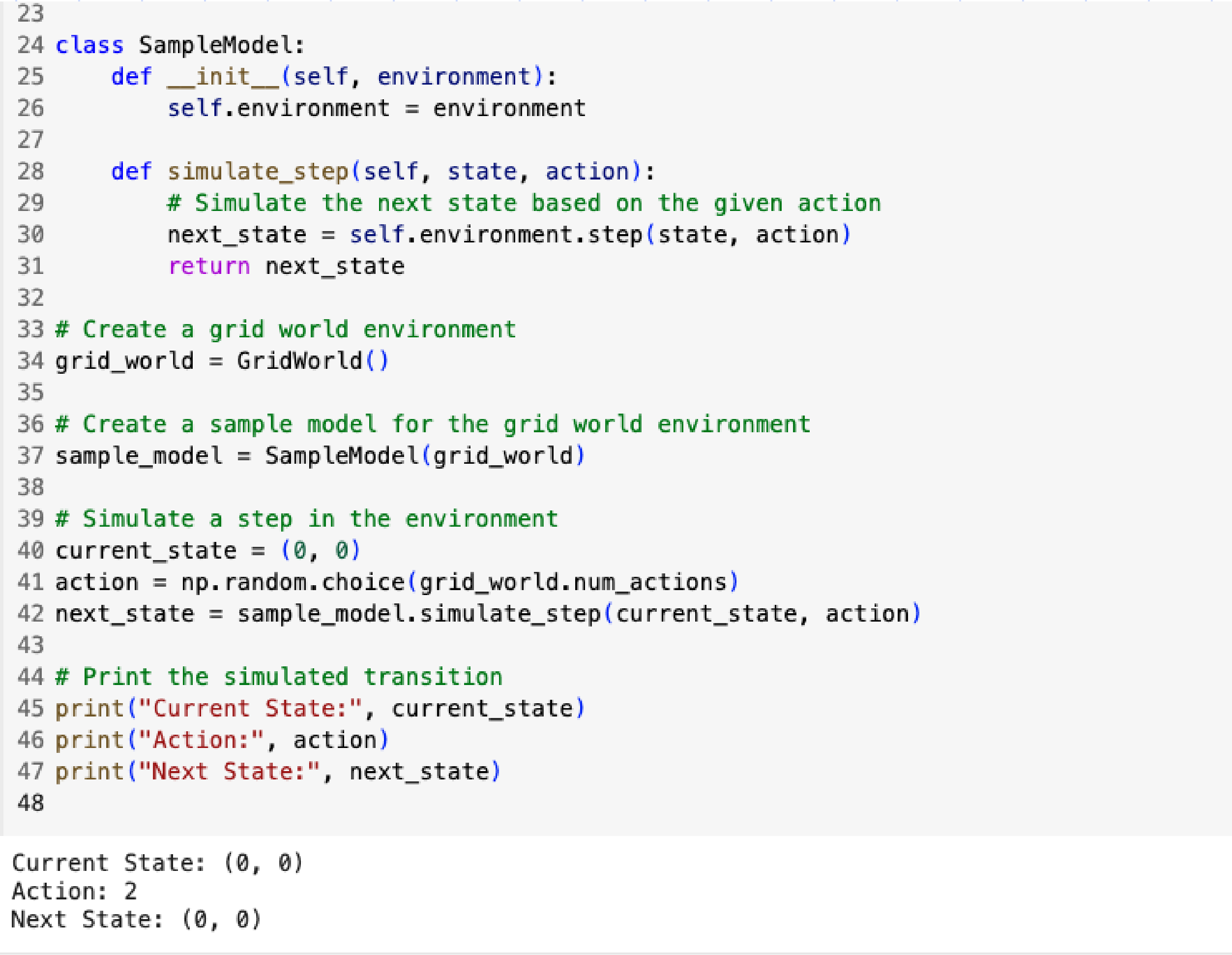
Sample Models Example

- □ Example of a sample model for a simple grid world environment



```python
# 2.10b- Sample Models Example- HoaDNt@fe.edu.vn
import numpy as np

class GridWorld:
    def __init__(self):
        self.grid_size = (3, 3)
        self.num_actions = 4  # Up, Down, Left, Right
        self.start_state = (0, 0)

    def step(self, state, action):
        # Define the dynamics of the environment
        row, col = state
        if action == 0:  # Up
            row = max(0, row - 1)
        elif action == 1:  # Down
            row = min(self.grid_size[0] - 1, row + 1)
        elif action == 2:  # Left
            col = max(0, col - 1)
        elif action == 3:  # Right
            col = min(self.grid_size[1] - 1, col + 1)
        next_state = (row, col)
        return next_state

```

Sample Models Example



```python

class SampleModel:
    def __init__(self, environment):
        self.environment = environment

    def simulate_step(self, state, action):
        # Simulate the next state based on the given action
        next_state = self.environment.step(state, action)
        return next_state

# Create a grid world environment
grid_world = GridWorld()

# Create a sample model for the grid world environment
sample_model = SampleModel(grid_world)

# Simulate a step in the environment
current_state = (0, 0)
action = np.random.choice(grid_world.num_actions)
next_state = sample_model.simulate_step(current_state, action)

# Print the simulated transition
print("Current State:", current_state)
print("Action:", action)
print("Next State:", next_state)

```

```
Current State: (0, 0)
Action: 2
Next State: (0, 0)
```

Summary

- Describe what a model is and how they can be used
- Classify models as distribution models or sample models
- Identify when to use a distribution model or sample model
- Describe the advantages and disadvantages of sample models and distribution models

Q & A