## Defining the Environment and Presenting the Problem

The Warehouse Environment (WarehouseEnv) is a custom simulated environment crafted to mirror the operations of a warehouse. While it draws inspiration from environments in the OpenAI Gym framework, it operates independently. Within this environment, a virtual robot is tasked with traversing the warehouse to collect packages and transport them to specified delivery points.

A screenshot of a game

Description automatically generatedThe warehouse layout is represented as a 5x5 grid, with each cell denoting a distinct location within the warehouse.

Key components of the Warehouse Environment include:

* A robot which maneuvers within the warehouse grid, interacting with packages, delivery points, and obstacles.
* Packages which are randomly dispersed throughout the warehouse, symbolizing goods awaiting pickup and delivery.
* Delivery points that are also randomly situated within the warehouse, marking destinations where packages must be delivered.
* Obstacles introduced to replicate potential obstructions or impediments within the warehouse environment, necessitating the robot to navigate around them.

The objective of the Warehouse Environment simulation remains consistent, focusing on the optimization of warehouse operations, particularly in the realm of package retrieval and delivery.

Key challenges addressed by the Warehouse Environment include:

1. **Navigation:** The robot must adeptly navigate the warehouse grid, circumventing obstacles while efficiently progressing towards packages and delivery points.
2. **Package Handling:** Upon reaching a package, the robot must collect it and update its inventory accordingly.
3. **Delivery Execution:** After securing a package, the robot must transport it to the designated delivery point. Failure to deliver a package to the correct destination may result in penalties.
4. **Inventory Management:** The robot must judiciously manage its inventory, ensuring it retrieves and delivers packages as required while minimizing unnecessary movements.

The overarching aim of this project is to devise AI-driven strategies, particularly leveraging reinforcement learning techniques, to tackle these challenges and optimize warehouse operations. By training the virtual robot to make informed decisions based on its observations of the environment, the endeavor seeks to bolster efficiency, curtail costs, and elevate the overall efficacy of warehouse logistics.

## State Transitions and Rewards

### State Transitions

In Figure 1, the warehouse is depicted as a 5x5 grid enclosed by impassable walls, creating a defined space for navigation. Within this grid, the robot has the freedom to move vertically and horizontally, unhindered by obstacles. To emulate real-world scenarios, we've introduced two obstacles into the environment, adding complexity to the robot's path.

In addition to basic movements, the robot can also attempt to pick up or drop off items at any location within the warehouse. Before executing any action, our algorithm rigorously checks its legality, ensuring that the robot doesn't breach any boundaries or collide with obstacles. If collision happens the episode ends. If the selected action involves picking up or dropping off a parcel, the algorithm verifies the presence of a package at the designated location. Failure to find a package results in a negative reward, a concept we'll discuss further below.

Upon successful validation, the algorithm updates the robot's current position. If the action entails dropping off a package, the algorithm further verifies whether the robot is placing the package in the correct delivery point. Correct placement is rewarded positively, whereas attempting to place the package in the wrong location incurs a negative reward. Notably, a package can only be delivered to its designated destination, ensuring precise delivery.

The state transition function is essential for guiding the learning process in reinforcement learning algorithms. By defining a well-designed function, we enable the robot to learn optimal policies for navigating the warehouse and completing its tasks efficiently.

These validation policies may inadvertently lead our agent to repeatedly select the same action, impeding learning progress. In stochastic environments, this repetitive action selection behavior can delay learning, necessitating adaptive strategies for effective navigation.

### Rewards

The reward function in our Warehouse Environment assigns numerical rewards to the robot based on its actions and the resulting state of the environment. This function serves as a feedback mechanism, guiding the robot's behavior by reinforcing desirable actions and discouraging undesirable ones.

The rewards attribution is defined by the following structure:

|  |  |
| --- | --- |
| Action | Reward |
| Pick up package in correct place | + 10 |
| Attempting to Pick Up in empty cell | - 1 |
| Drop off package in correct place | + 100 |
| Attempting to Drop off in wrong place | - 10 |
| Collision with Obstacles | - 5 |
| Positive reward for reaching intermediate steps (packages or delivery points) | + 1 |

The reward function plays a crucial role in reinforcement learning algorithms, shaping the agent's learning process by incentivizing actions that lead to desirable outcomes. By carefully designing the reward function, we can encourage the robot to learn effective strategies for navigating the warehouse, managing inventory, and optimizing its performance in achieving its objectives.