Pokemon Report

11610310 Lu Ning

Program Design

We mainly implement the feature for catching.

The ideal process of pokemon go:

For catching the pokemon, we control the turtlebot in the front of the pokemon with keyboard. And we want to calculate the distance between turtlebot and pokemon. If the distance is too large to catch pokemon, turtlebot will go forward until the distance is suitable for catching. In the end, the turtlebot will take a photo for pokemon.

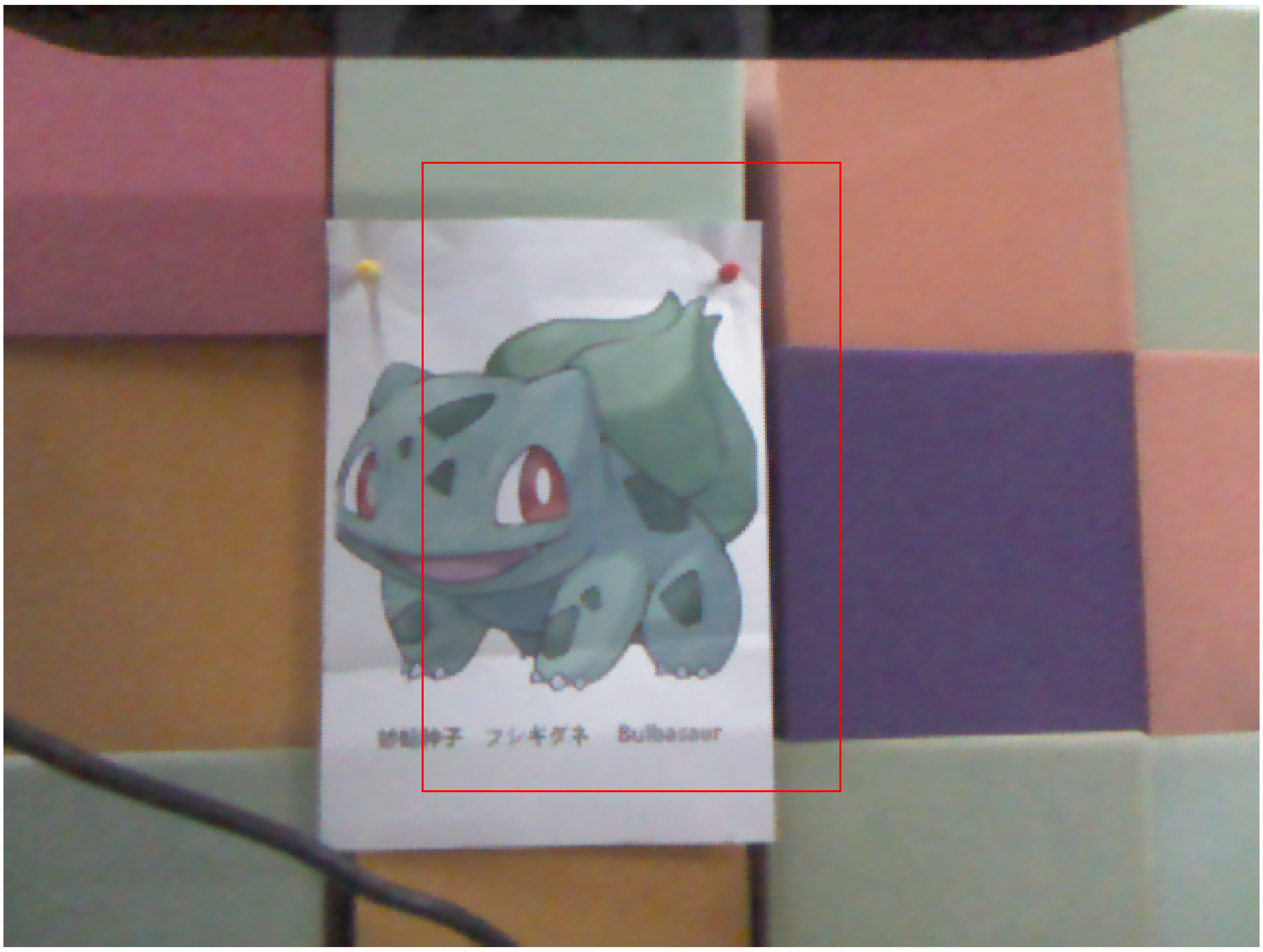
For the implementation code, we divide it into 3 main steps.

1. Run the searching code, and send the image information to "pokemon_go/search".
2. Getting the distance between turtlebot and pokemon by subscribing the laser scan to control the turtlebot. If the distance is too large to catch the pokemon, turtlebot will go forward until the distance is suitable for catching.
3. When the distance is suitable for catching, the turtlebot will stop and the sleep function call to avoid the blocked.
4. Getting the image of searching with subscribing topic:"pokemon_go/search" published by searching code. And when getting the image, we store the image.
5. To avoid image covered by the new one.

Pictures

噴火龍 リザードン Charizard

皮卡丘 ピカチュウ Pikachu


妙蛙种子 フシギダネ Bulbasaur

巴大蝶 バタフリー Butterfree

The control video

The videos are in the `video` folder.

We the control screen is in the `screen_record` folder and the `running` folder record how the robot running in the real world.

Ros Package

We built two packages:

- pokemon :

  This package use the file in sakai we don't modify it.

  Commond:

  ```
  rosran pokemon pokemon_search
  ```

- move :

This package uses the `move.cpp` source file written by us. We use the information in `laser_scan` to get distance information and control the robot through `cmd_vel_mux/input/navi`.

Commond:

```
rosrun move move_node
```

* Need to turn off the keyboard control process, because it occupies the control node.

* The source packages in the `src` folder.