

Project – Pokemon Turtle

1. Introduction

In this class, you have to complete a final task in the end of semester. You shall master the skill of coding ROS programs and using ROS tools for general projects. The final project will let you show your intelligence and passion in robot project, meanwhile it is very interesting!

The name of this final project is called Pokemon Turtle. You are asked to use turtlebot to get Pokemons. In Researching Building I (because of the coronavirus, we may do it in Gazebo), we will provide a scene of a little maze constructed by sponge squares as shown in Figure 1. Somewhere in the maze hidden some Pokemons (some pokemon pictures posted upon the walls). You don't know what the maze looks like, it may be dangerous (you should **stand away from the maze without seeing it** and you should not crash the wall). So you send your turtlebot to explore the maze by remote control or automatic control. Then get some Pokemons (save some pokemon images in your computer under some rules).

Pokemon Get daze!



Figure 1 The sponge maze in Researching Building I

2. Rules

Because some of the wild Pokemons are very cautious and timid. While the others are fierce. It is not easy to catch a Pokemon. However, Professor Haokido gave some instructions that you should keep a certain distance and angles to catch a Pokemon. He gave us a target device shown as Figure 2. If we can keep a Pokemon in the center of the red rectangle and fit the size well, it will be easily caught.

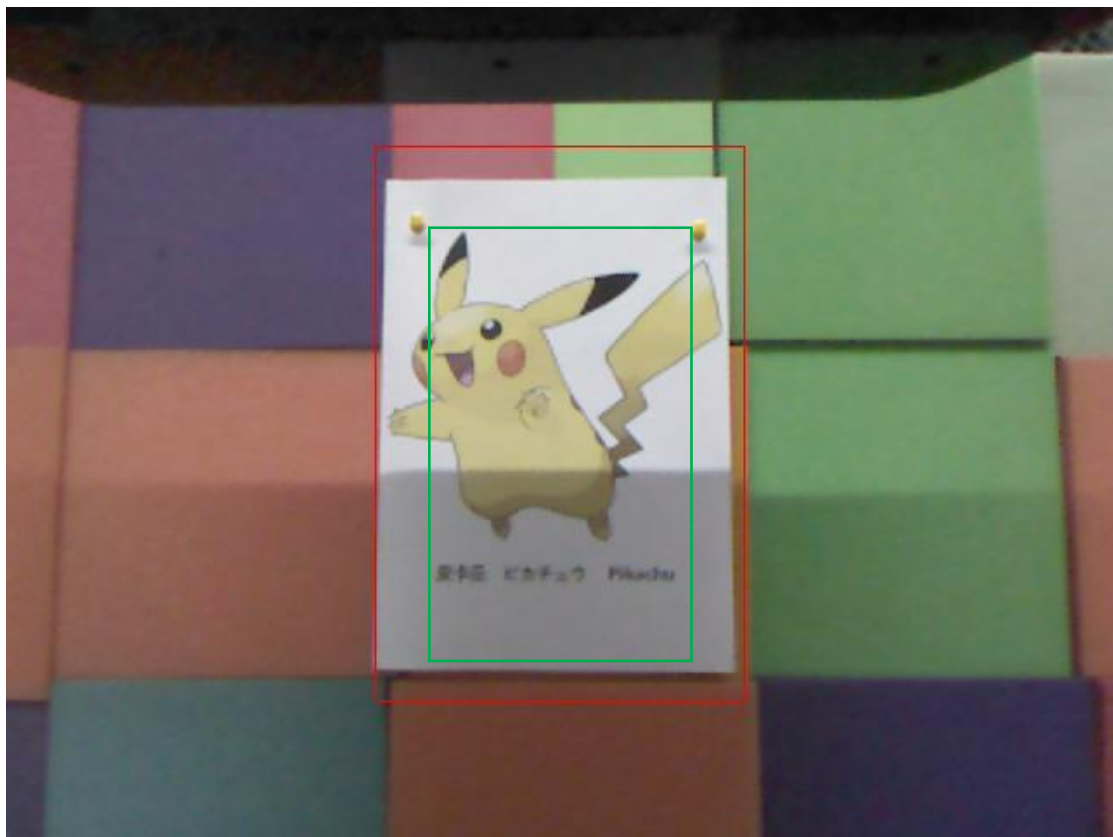


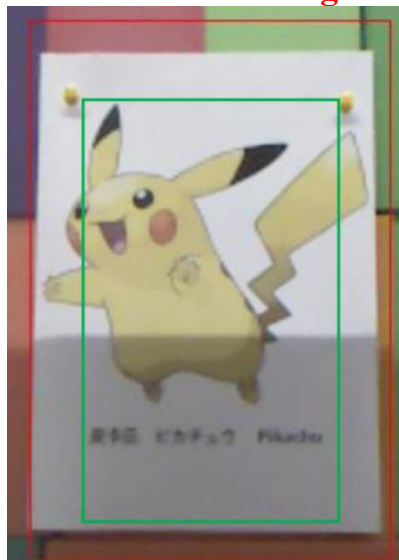
Figure 2 Pokemon catching target

Here is the rules:

2.1 Six Pokemons are qualified but you can get as many as you can.

2.2 A valid catching is shown in **Figure2** , The red box is the picture, and the green box is the pokemon. Since pokemon is in the picture, the green box must be in the red box, and if you encounter two boxes whose coordinates are almost the same, you can enlarge the red box a little bit.

Remember a valid catching as follows:



2.3 Your score depends on the following things:

- 1) Number: one valid catching gets 1 point until 15 points. Less than six gets 0 point.

2) Time: total 15 points. Every group has at most 7 minutes.

TIME (min)	POINTS
< 3	15
$3 \leq \&\& < 5$	10
$5 \leq \&\& \leq 7$	5

3) Strategy : we encourage you to use novel and high level techniques. Here are the bonus:

TECHNIQUE	Point
Auto identification	10
Auto driving	30
Multiple robots	70

Auto identification means automatic identifying a Pokemon picture then modifying the pose and save a valid picture.

Auto driving means you can not give any control to the robot during the game.

Multiple robots means use more than one robots.

these techniques are upward compatible, which means the high score technique should include all the lower ones. You should achieve them one by one.

Note that whether you choose to use a single robot or multiple robots, you need to put them all at the entrance (the maze has two doors, you can choose either one as the entrance).

4) Penalty: any crash of the wall or two robots (if you use multi-robot) gets a deduction of 5 points.

3. Pre-project(mid-project)

Before the final project, let's do the pre-project. It is a simplified version of final project and will be seen as a common lab task. So you have to do this and submit the results. In this task, you are required to complete 1) 、 2) 、 4) of 2.3, which means as follows: you manually control turtlebot to search for pokemon and use a catching node to catch 6 pokemons within a specified time, which is the basic requirement of mid-project.



3.1. Guidance of pre-project

- 1) Firstly, you should download the provided file, [pokemon_searching.cpp](#). Create a new package in your workspace and put it into the src/ subdirectory in your package. Please read the file carefully to figure out what it does and what depends it requires. Edit your CMakeLists.txt and Package.xml (if needed) and then build it. Run the executable as a ROS node, this node requires a topic /camera/rgb/image_raw in sensor_msgs/Image message type and publishes the same type. So be sure this topic exists in your ROS network while running this node, whatever a Kinect or a rosbag provide it.

- 2) If you successfully run the node, you will see it just provide a window showing the original image flow except a red square frame added on it (as Figure 2). That is your searcher. You should **manipulate your robot by keyboard** using the window to see. You can also **run a gmapping** on you robot so that you can also know where you are.
- 3) Once you meet a pokemon and want to catch it, the first thing you should do is to rotate your robot and let the pokemon picture fall in the middle of the frame. After that, the only thing you are expected to do is **run a or a punch of node(s), which modifies the distance between the picture and robot until it meets a specific range and saves the image (with the red rectangle).**

Considering computer configuration, mid-project requires you catch 6 pokemons and record a video. About the point of time, we will relax the requirements appropriately.

4. Final Project

In this task, you are required to complete 1) 、 2) 、 3) 4) of 2.3.