

## Documentation

The code for the project can be broken into three separate parts.

1. Simulation
2. Image and Move Processing
3. Choregraphe Files

### NAO Setup

Before running our code, some directories must be created. To start ssh into the NAO using the following command:

```
ssh nao@<naos.ip.address>
```

This will take you into the NAO's home directory. Here you must run the following commands:

```
mkdir cs473
mkdir cs473/hesitation
mkdir cs473/hesitation/data
mkdir cs473/hesitation/sounds
```

This will create all necessary directories. You will then need to move all the necessary sound files onto the NAO as well. Navigate to the folder where the sound files are located, and run the following commands:

```
scp think1.wav nao@<naos.ip.address>:cs473/hesitation/sounds
scp think2.wav nao@<naos.ip.address>:cs473/hesitation/sounds
scp think3.wav nao@<naos.ip.address>:cs473/hesitation/sounds
scp think4.wav nao@<naos.ip.address>:cs473/hesitation/sounds
scp noise1.wav nao@<naos.ip.address>:cs473/hesitation/sounds
scp noise2.wav nao@<naos.ip.address>:cs473/hesitation/sounds
scp noise3.wav nao@<naos.ip.address>:cs473/hesitation/sounds
```

### Simulation

#### TakeTwo.java

This is a simple Java file which is meant to run on a computer. To run this file, first enter the terminal or the command line, navigate to the proper directory, and type the following:

```
javac TakeTwo.java
```

This will compile the simulation. To run the simulation then type:

```
java TakeTwo <participant number>
```

This will open the simulation, create a folder in the directory called Data, and record the move times in a file in this folder called <participant number>.

## Image and Move Processing

### **testImage.py**

This python file contains all the code for processing analyzing a picture of blocks, extracting their colors, and returning the number of blocks to take, and their point values. The code is run from Choregraphe, and must be placed on the NAO.

To place the code on the NAO, navigate to the folder containing the testImage.py, and run the following command:

```
scp testImage.py nao@<naos.ip.address>:cs473/hesitation
```

This will place the image testing file in the appropriate location on the NAO.

**Note:** The testImage.py code can be run locally on your computer only if PIL is installed. For the NAO, this should already be installed. To run locally, one must change the location of the block image your current directory. This change can be made on line 26 of the code.

## Choregraphe Files

**neutral.crg**

**thought.crg**

**hesitation-neutral.crg**

**hesitation-thought.crg**

Each of these four files can be executed in the same way, and if all the steps above are followed properly, should run without any issues. Open the files in Choregraphe and simply click the green play button at the top of the software will start the game play. The right foot must be tapped to initiate the NAO.

Data for move times is stored in the cs473/hesitation/data folder.

## Overall Structure

The simulation is completely independent of both the image processing and the Choregraphe files, and should run regardless of the whether these are correctly placed.

The image processing file can be run locally, without the Choregraphe files, however the Choregraphe files will not run properly if the image processing files are not placed in the correct folders on the NAO.

Every time a picture is taken, Choregraphe will execute the testImage.py file on the NAO, which reads in the new image. The image is processed, and the results are then returned back to Choregraphe. Without this interaction the Choregraphe files cannot run properly.

Choregraphe also uses the sound files to make the thought and neutral noises. These must be in the correct locations, or the sounds will not work properly.