# Task 1: Massive Report (Qianchen’s Version) Inquisit Verison for M-Turk

1. Experiment Procedure

This experiment is a version which replicate Qianchen’s version by Inquisit. But this this time each experiment has less trials (14 trials each) and the order of the Image presentation can be random.

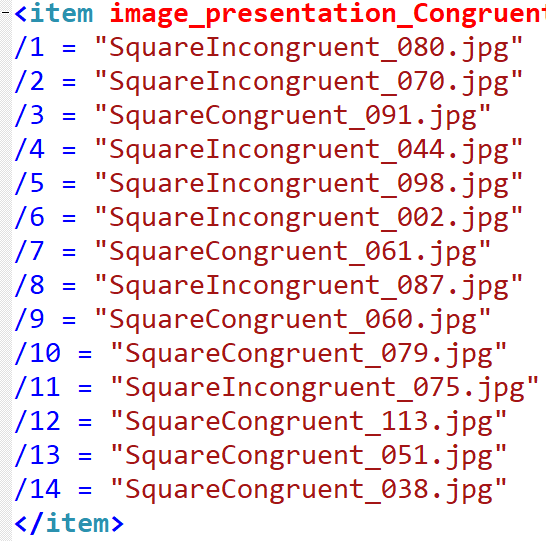
In each trial, a big Image will be followed by 500ms cross fixation at the center. The image will flash for 133ms and masked by 5 masks. Each mask takes 60ms. Then, it will accompany 21 questions. (21 patches) These patches are made of 4 or 5 patches from the present Image, 1 patch contain object item and 16/15 absent patches. (Depend on the critical patch location)

1. Scripts Description:

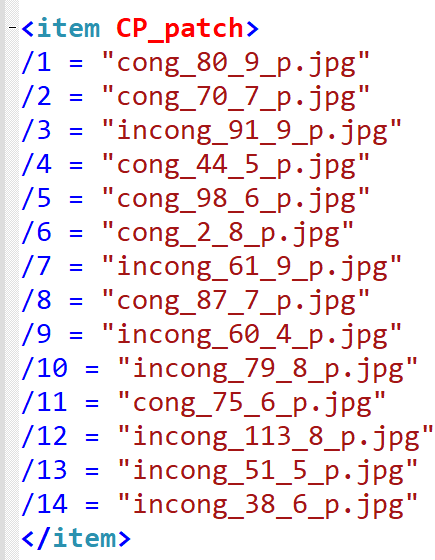
* BaseScript\_B1\_Gx.iqx

This Script created by an .m file named ‘ListCreate\_individual\_fix.m’. It contain lists of images need during this experiment. For example:

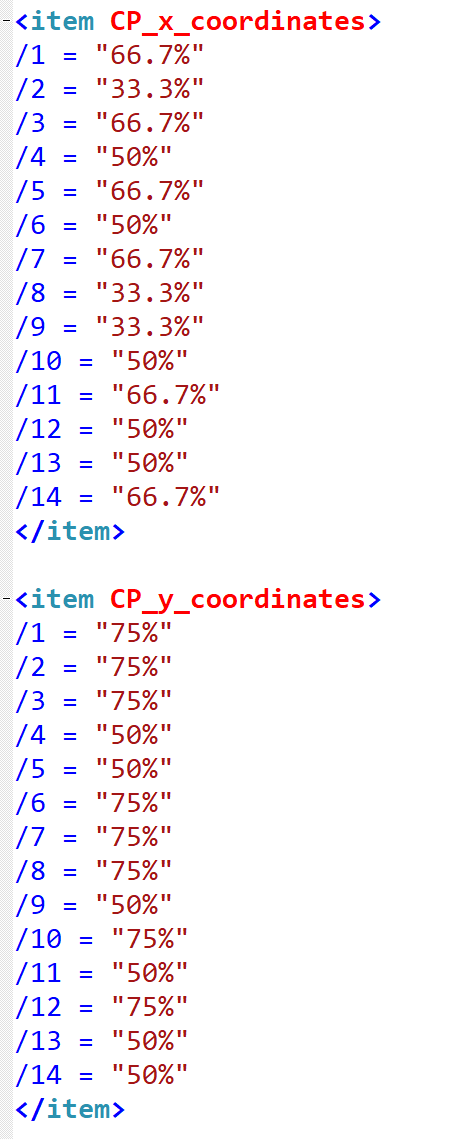
List of present Images:



List of critical patches correspond to the same order



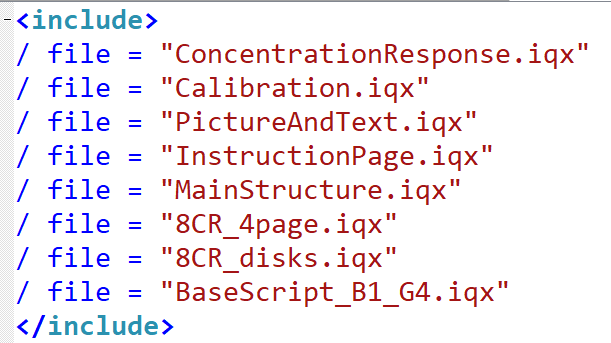
List of position of the critical patch:



And so on.

* Script\_B1\_Gx.iqx

This script is used to start the experiment. It includes all files the experiment needs.



* Mainstructure.iqx

The main body of the experiment. It contains the strategy of trials present and collect correspond data:

Fixation Step:

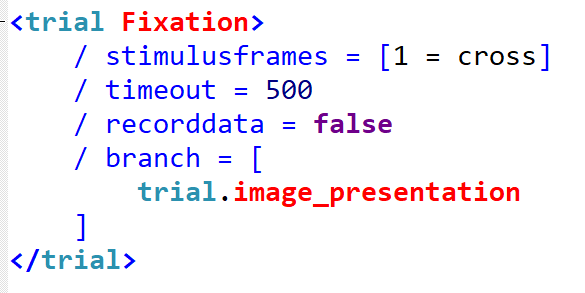
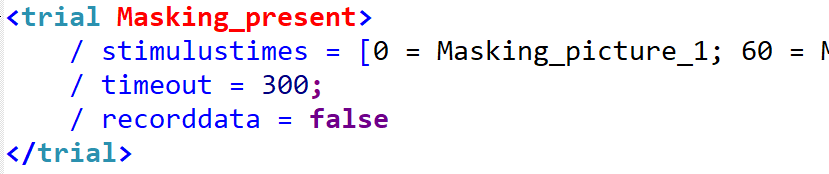


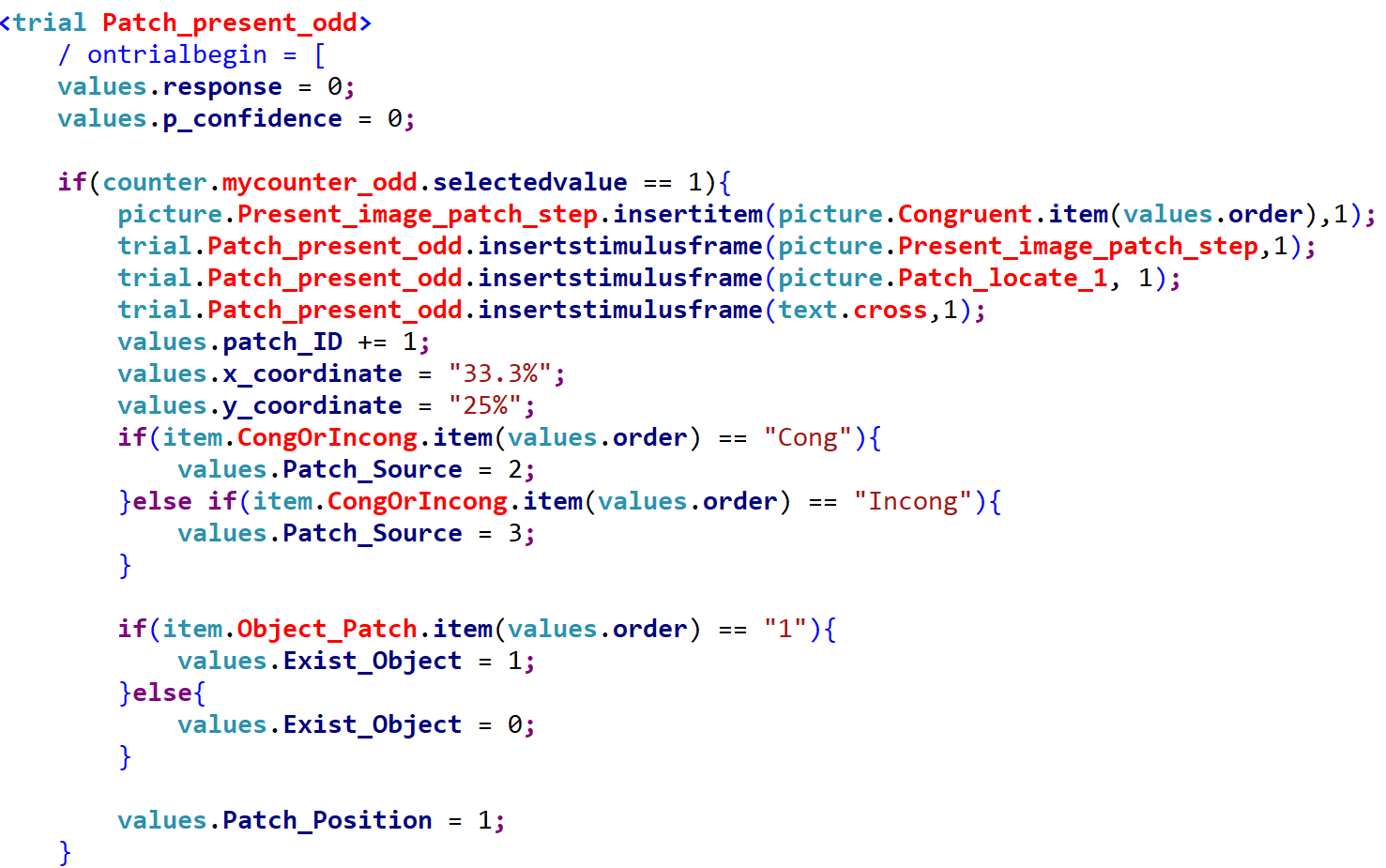
Image Presentation Step:



Masking (Big Image) Step:



Patch Present Step:



Here codes have 2 cases, <Patch\_present\_odd> and <Patch\_present\_even>. They both operate similar. ‘Odd’ points to the critical patch locate on odd position and ‘even’ means even position.

By using <counter mycounter\_odd>(or even) to realise random following questions:

Odd Case:

Counter.selectedvalue == 1: patch position 1;

Counter.selectedvalue == 2: patch position 3;

Counter.selectedvalue == 3: patch position 5;

Counter.selectedvalue == 4: patch position 7;

Counter.selectedvalue == 5: patch position 9;

Counter.selectedvalue == 6: Critical patch;

Counter.selectedvalue == 7-21: Absent patch;

Even Case:

Counter.selectedvalue == 1: patch position 2;

Counter.selectedvalue == 2: patch position 4;

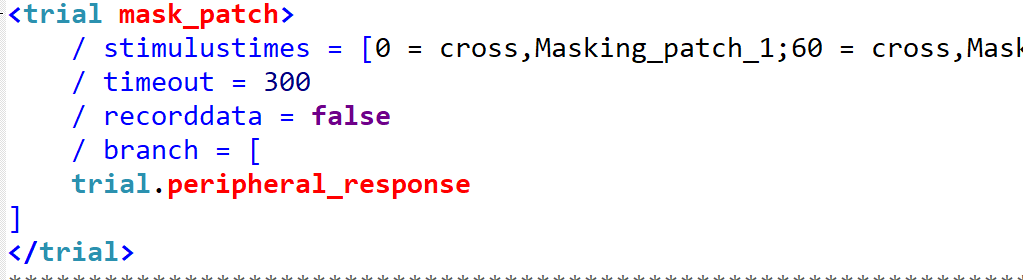
Counter.selectedvalue == 3: patch position 6;

Counter.selectedvalue == 4: patch position 8;

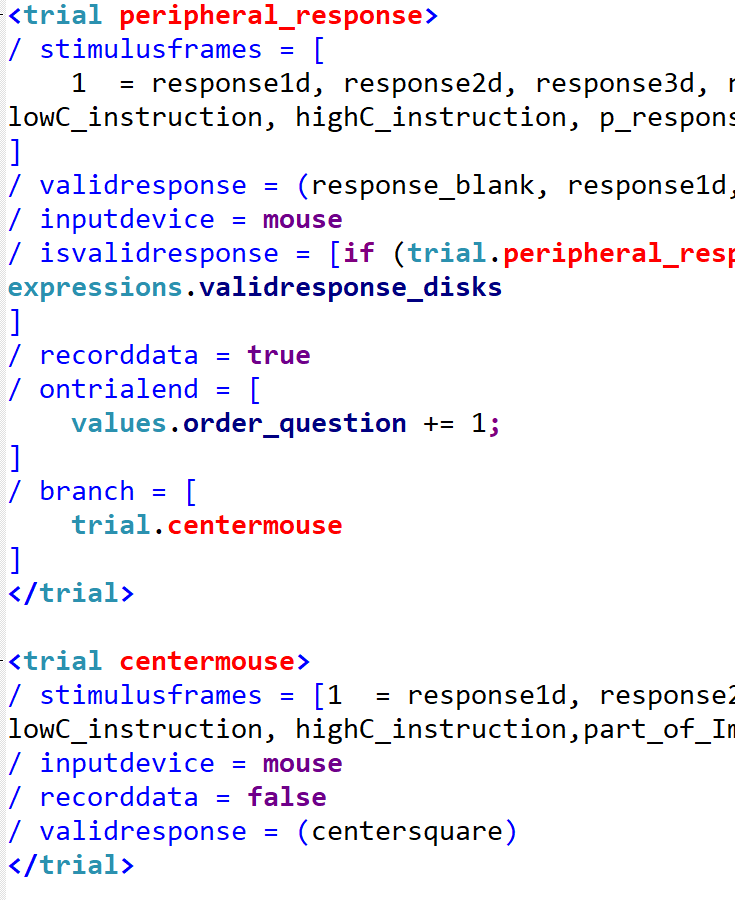
Counter.selectedvalue == 5: Critical patch

Counter.selectedvalue == 6-21: Absent patch;

Masking Step (Patch)



Response Page and Re-centre cursor:



* PictureAndText.iqx

This file contains most of <picture> and <text> syntax which used in the ‘MainStructure.iqx’

* InstructionPage.iqx

Contents for Instruction Pages

* Calibration.iqx

Contents for Calibration pages

To avoid the influence of different size of subjects’ screen and different resolution, we want to keep the same size when subjects see the images.

When a man keeps his arm straight, the visual angle of his thumb and eyes is 2 degrees. There will be a line shown on the screen as reference line. Subjects adjust their body position until they see their thumb has the same width of that line. Then keep this position throughout the experiment.

* ConcentrationResponse.iqx

This page comes up after one trial. It collects the concentration level of the subject when they did the previous trial.

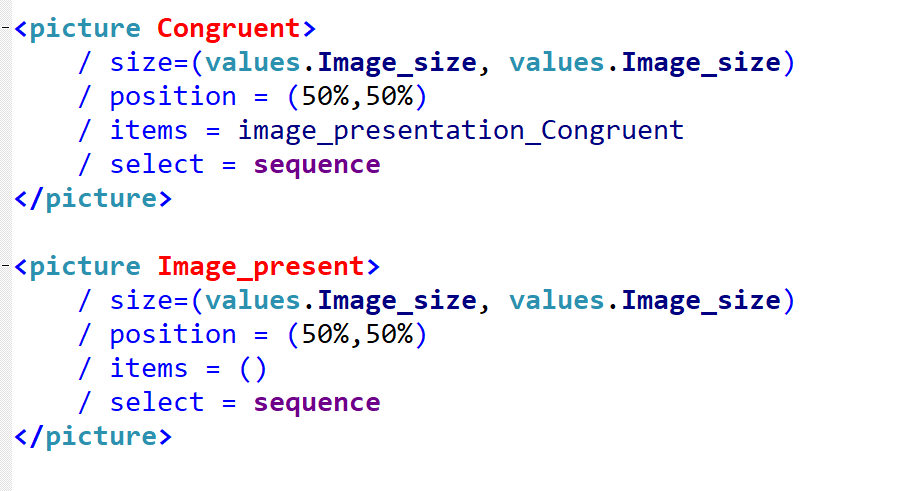
1. Strategy

* Self-Random Select

Create multiple lists in file ‘BaseScript.iqx’, each list will have same order number for each image. For instance, big image #1 has the order ‘/4’ then, its critical patch and other items relate to this image will have the same order ‘/4’ as well.

After creating these lists, using a syntax <counter> to pick a value randomly. According to this value, take out the particular item and insert to correspond trial.

For instance:



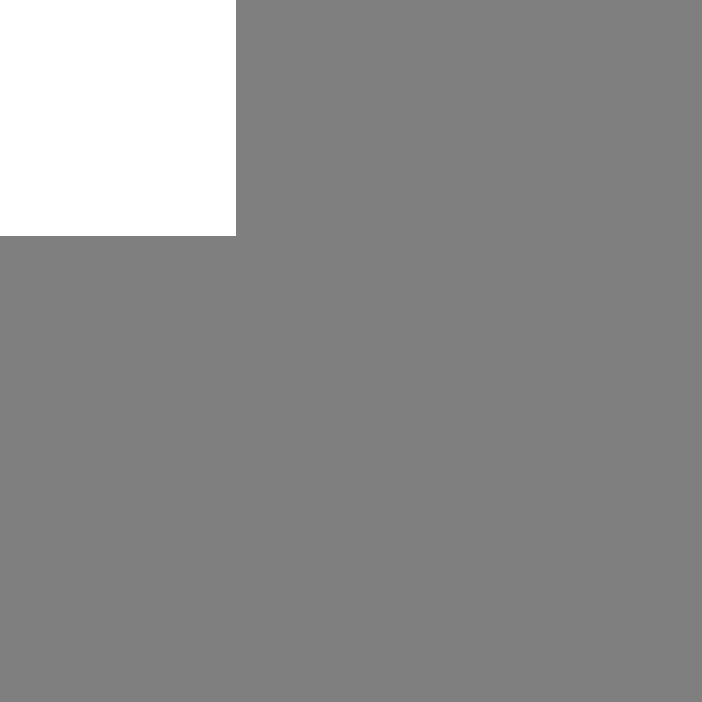
<picture Congruent> works like a source which take all the image through the experiment. If the counter choose order ‘/5’, then take out number 5 item in <picture Congruent> by using ‘picture.Congruent.item(5)’ and insert this item into <picture Image\_present>. In this method, the experiment can operate random select every time subject start the script.

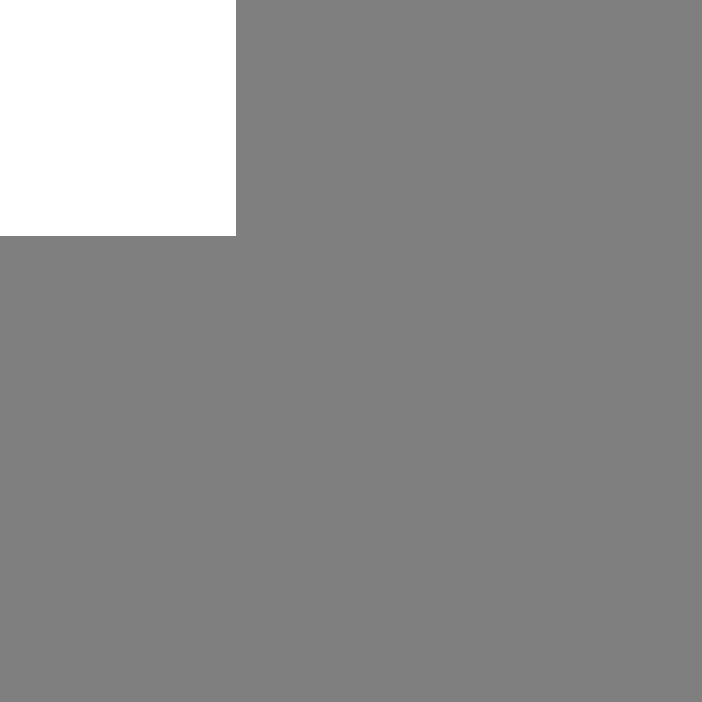
Another important thing is you need to remember clear the items in <picture Image\_present> after finishing showing the correspond images.

* Trick on patch present

In the beginning, I copy all small patches which may be used in each experiment file. But it caused the codes complicate and the file is too big to upload on Mturk. Then I use a trick to reduce the number of patches you need to contain in the file.

For the patched comes from the original image, instead of showing the correspond patch individually, we show the present big picture and a particular same size .png picture which has part of transparent in particular position.

For example:



+ =

The cover picture has the background colour.

So in this method, the patches comes from present image for all trials can use 9 different png pictures to replace. Highly reduce the size of the file and time to prepare the experiment online.

* Matlab files operate order:

Firstly, run the ‘ListCreate\_individual\_fix.m’ to create base list files.

Secondly, run the ‘IncludeOrderCreate.m’ to create start scripts

At last, run the ‘CopyFiles.m’ to create individual experiment files.

They are located in the folder named ‘WebVersion’

* Patch Location Coordinate:

|  |  |  |
| --- | --- | --- |
| 33.3%,25% | 50%,25% | 66.7%,25% |
| 33.3%,50% | 50%,50% | 66.7%,50% |
| 33.3%,75% | 50%,75% | 66.7%,75% |

These coordinates values only feasible when the present image size set to 75%. If size has been changed, these values need to calibration again.

1. Tips

Online test can’t have nested folders. You need to put all files and scripts into the same folder and then upload online.

Some syntax may cause an unexpected problem when you do the online test. Even you find it correct on your own laptop.

# Task 2: Massive Report (2nd Version) Inquisit Verison for M-Turk

1. Experiment Procedure

This experiment is quite similar with Qianchen’s version. But this time, each trial has only 6 questions (6 patches).

These 6 questions consist of 40% patches from present image, 5% Critical patch(From Present Image), 5% Critical patch(Inverse) and 50% Absent patches.

So, the content types of 6 patches may not fixed in different trials.

1. Scripts Description:

* BaseScript\_B1\_Gx.iqx

Same structure of Qianchen’s Version

* Script\_B1\_Gx.iqx

Same structure of Qianchen’s Version

Run this script to start the experiment

* Mainstructure\_V2.iqx

Similar structure of Qianchen’s Version

* PictureAndText\_V2.iqx

Similar structure of Qianchen’s Version

* Instruction.iqx

Similar structure of Qianchen’s Version

1. Strategy
2. This task has similar strategy with Qianchen’s version. But the only different part is the patch present step.

We use a counter which choose from 1 to 20:

When the selected value is from 1 to 8: the patch will be the present patch

When the selected value equals to 9: the patch will be the critical patch (Present)

When the selected value equals to 10: the patch will be the critical patch (Inverse)

When the selected value equals to 11-20: the patch will be absent patch.

1. The absent patches are presented at any position of 9 positions in the Qianchen’s version. In this 2nd version has some differences. The absent patch is cut from an image by 3 x 4. Now we only take 9 patches on the left side. (So the patch ID in the image will be 1,2,3,5,6,7,9,10,11) And each absent patch will be present in the same position it cut from. For example, if the absent patch is cut by location 3, in the patch present step this patch will be presented at location 3 as well.
2. Firstly, run the ‘ListCreate\_individual\_fix.m’ to create base list files.

Secondly, run the ‘IncludeOrderCreate.m’ to create start scripts

At last, run the ‘CopyFiles.m’ to create individual experiment files.

They are located in the folder named ‘WebVersion’