

## **CS566: SPEECH PROCESSING**

### **Group 5: Tile Flipping- Memory Game**

Ayasha Agrawal	184101008
Megha Jain	184101024
Shubham Jain	184101035

## Tile Flipping- Memory Game

In this game there are 3 levels level 1, level 2 and level 3 comprising of 4, 6 and 8 tiles respectively. Each tile is of different colour and behind every tile there is a symbol. The player can see the symbols behind the tile by speaking the colour of that tile. The symbols are present in pairs means the same symbol is present behind two tiles and the game is to remember the tiles behind which the same symbol is present.

### RULES:

The player has to choose two tiles say choice 1 and choice 2. The player can choose tile by speaking its colour and then the symbol behind it will be shown to player. If the symbol behind both the choices are same then 5 points are awarded and both the tiles will get permanently flipped (symbols will be shown) and now the player has to match symbols among the remaining tiles and if the symbols behind the chosen tiles are different then 2 points are deducted as penalty and both the chosen tiles will again get flipped and again you have to choose 2 tiles.

### LEVEL 1:

LEVEL 1:

ORANGE

YELLOW

PINK

WHITE

Enter your 1st color choice

@

YELLOW

PINK

@

As you can see there are 4 tiles of different colours and the player choose orange and white tile and the symbols got matched so now these two tiles got flipped permanently.

```
congratulation you have won the game with a score of: 10

      @           !
      !           @
```

Here all the symbols got matched so this level is cleared and the player moved to level 2.

### Level 2:

```
LEVEL 2:

      ORANGE      YELLOW      PINK
      WHITE       PURPLE      INDIGO

Enter your 1st color choice
```

In this level there are 6 tiles of different colours.

```
ORANGE      YELLOW      PINK
!           !           INDIGO
```

```
Not matched , Try again

      ORANGE      YELLOW      @
      !           !           #
```

```
ORANGE      YELLOW      PINK
!           !           INDIGO
```

Here white and purple tiles got matched and got flipped permanently but then pink and indigo didn't get matched so they again got flipped back to tiles.

Congratulation you have won the game with a score of: 11

@	#	@
!	!	#

When all the symbols get matched the player reached the last last level

### Level 3:

LEVEL 3:

ORANGE	YELLOW	PINK	WHITE
PURPLE	INDIGO	GOLDEN	BLACK

Enter your 1st color choice

congratulation you have won the game with a score of: 18

!	#	\$	#
%	%	\$	!

There are 8 tiles in this level .When all the symbols get matched the player wins the game and the final score is displayed on the screen.

### Training the Model

The training was done with 90 utterances of each color with three different speaker- one male and two female. The data was pre – recorded with cool edit tool. The frame limit was set to 125 at max. The average of model was done for three iterations, taking model generated at previous iteration as base. The threshold value of 10 to the power of -40 (difference in P\* in consecutive iteration) was used to train the model for particular observation sequence.

### **Steps Followed:**

1. Pre-recorded utterances of colors are used to generate the observation sequences.
2. The obtained observation sequences are used to train the models for different colors.
3. The trained model are then used for testing purpose.

### **Live Training**

A separate model is used to train the model for new user. The pre-trained model was used for adjusting the model for new user.

Steps:

1. New user is asked to speak colors for some n number of times.
2. The recorded data is used to generate the observation sequences followed by training the model.
3. The model obtained is used for testing purpose.

### **Testing the Model**

With the pre-recorded files the accuracy obtained was 100%. On live testing from the same speakers the accuracy around 96 % – 100 % is achieved.

For new speaker the accuracy is around 95 % – 100 %, varying on background noise recorded.

### **Architecture of the System**

Hidden Markov Model is used to develop the tile flipping game based on speech.

HMM Element considered:

1. Number of states: 5
2. Number of Observation symbol: 32 ( codebook size)
3. Feed forward (baker's) model is used.

## Recording

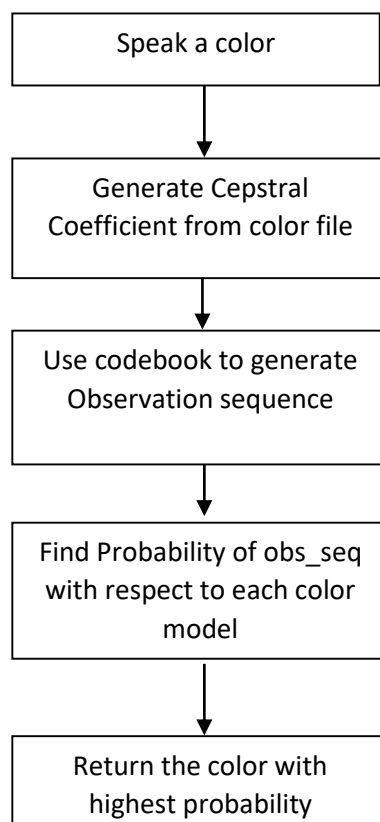
Following color are used to train and test the game. The recording is done with cool edit software. The sampling rate used was 16000. Each sample of size 16bit, mono channel.

1. ORANGE
2. YELLOW
3. PINK
4. WHITE
5. INDIGO
6. PURPLE
7. GOLDEN
8. BLACK

For live recording, a separate recording module is used which dump file in .txt form.

## CodeBook

A 12 dimensional codebook of size 32 is used. It is generated using the cepstral coefficient which were generated from the pre-recorded utterance of different colors spoken by different users. For generation of codebook, binary split method along with k-means clustering is used.



**Fig.** High level Flow of system

## Using the code

The project source code is present in the folder named TILE\_FLIP.

It contains code for testing and training separately.

Testing folder name is: Final\_project\_tile\_flip

Training folder name is: Train\_model\_for\_color

## Executing Live Training:

1. Open the Final\_project\_tile\_flip.sln file present in Train\_model\_for\_color.sln present in Train\_model\_for\_color/ Train\_model\_for\_color/ folder.
2. Execute the code by tapping F5.
3. User will be prompt to record 15 utterance of each color(8). Once the recording is done, it will train the model.
4. The updated model is stored in Train\_model\_for\_color\ Train\_model\_for\_color \Output\Average\_Models.
5. The final model file names are a\_0.txt – a\_7.txt , b\_0.txt – b\_7.txt
6. Copy these files in Final\_project\_tile\_flip/ Final\_project\_tile\_flip/Input/ folder.

Now the testing model is updated with new model.

## Executing Game(Testing):

1. Open the **Final\_project\_tile\_flip.sln** file present in Final\_project\_tile\_flip/ Final\_project\_tile\_flip/ folder.
2. The project is compatible with visual studio 2010 and 2013.
3. Run the project, by tapping F5. (Debug-> Start Debugging).
4. The system uses the pre-trained model for the game.