COSC 345 Assignment 4

Nathan Kennedy 1006384, Kat Lilly 8867318, Tobyn Packer 5003352, and Daniel Davidson 375621

October 1, 2018

Contents

1 The NKTD app

Our app now contains three functioning games which can be opened via speech commands from the main menu and controlled either by speech commands or through touch. Two of the games (2048 and Frozen Bubble) are open source games we have manipulated to work with voice commands and Teragram is an original game, a maths practice game for children.

The application uses pocketSphinx to interpret speech commands: https://github.com/cmusphinx/pocketsphinx

1.1 Since the Beta

We have made the following improvements and enhancements since our beta release:

- Bugfixes:
  - Fixed memory leaks upon closing Activities.
  - Frozenbubble: Fixed launcher rotation speed.
  - Teragram: Voice Recognition only restarts when it is supposed to.
- The whole app now has a consistent design and colour scheme.
- Modified frozen bubble launcher so that gameplay is not negatively impacted by the time taken interpret commands. The launcher stops moving immediately on hearing a command, but waits to interpret it before launching the bubble, this means the delay in interpreting a speech command doesn't ruin the game.
- Frozenbubble speech implementation was finished, making it completely playable hands-free.
- Added an animation to the listening button, which is intuitively easy to understand and shows the user when the app has heard something.
- Changed the way the radio buttons are labelled in multiple choice quizzes in Teragram - we originally labelled them as "a", "b", "c" etc, but this was not suitable for voice recognition, so we changed it to numbers. This wasn't a satisfactory solution either, and we have now changed it to be labelled by colours instead.
- Speech recognition stops when focus shifts from NKTD such as when the phone goes to sleep.

2 Testing

Voice control

Testing an application that is run by voice commands was not something we felt could be done through automated methods without devoting a large amount of time that could instead be spent implementing features. Instead, frequent testing was done manually either through emulators or by installing the apk on our own phones. We note that the microphone on a typical phone is significantly better than the microphone on a typical laptop or desktop computer, and so the app is significantly easier to play on a phone than on an emulator.

Teragram

For the Teragram game, as well as testing it ourselves it has been user tested by two children, seven and four years old. They still love this game, and have still been playing it frequently. However, they do prefer to play it without using the voice control. This is partly because they do not often find themselves in a quiet enough place for the voice control to work reliably, but also because they get frustrated that voice controls are a bit slower and less reliable than entering answers via the touch screen. Unfortunately our app really does work less well with voice control than touch control.

3 User Documentation

3.1 Teragram

The game starts up at level one, and the user can move to the easiest level by tapping the 'too hard' button or go up levels by tapping the 'too easy' button. When ten questions in a row are answered correctly, the game will automatically level up, otherwise if two questions are incorrect in a row, it'll automatically level down. The user is not aware of the actual level, it's just that questions get easier or harder - the bound on the random number generator used in setting new questions is related to the level.

The game launches in addition mode, and the user can chose to move to subtraction problems, or practice times tables or powers of two.

3.2 2048

2048 is a popular number name released on March 20, 2014. This game is very simple; you can choose to slide the tiles up, down, left or right by swiping the screen or saying 'left', 'right' 'up' or 'down'. Every slide, all the digital squares will move in the direction the screen was slid. This is a really neat game, and there are many open source implementations out there, but what makes it especially suitable for our app is the simplicity of its controls - it's an interesting and challenging game controlled with only four commands. Also there is no time pressure in this game, so if it takes half a second for the speech recogniser to work, this does not adversely affect the game play.

3.3 Frozen Bubble

The aim of this game is to clear the screen by shooting the current bubble from the launcher to its respective coloured bubbles on the level. Saying 'fire'/'now' or swiping up will shoot the bubble.

3.4 Voice Controls

You can control the game with the following voice commands:

3.4.1 Main menu

- **'Teragram'** or **'game two'** - this will take you to the game Teragram.
- **'Twenty Forty Eight'** or **'game three'** - this will take you to the game 2048.
- **'Frozen Bubble'** or **'game four'** - this will take you to the game Frozen Bubble.

3.4.2 Teragram

- **'help me'** - Brings up help menu with available voice commands.
- **'harder'** - Provides a harder question.
- **'easier'** - Provides an easier question.
- **'new question'** - Provides a new question of the same type
- **'addition'** -Provides an addition question
- **'subtraction'** - Provides a subtraction question
- **'times tables'** - Starts times tables practice mode
- **'powers of two'** - Starts powers of two quiz
  - Say the colour of the correct answer: **'blue'**, **'green'**, **'pink'**, or **'red'**
- **'number'** - Starts Number Input mode, see below
- **'exit'** - Prompts a message confirming if they want to exit

3.4.3 2048

- **'help me'** - Brings up help menu with available voice commands.
- **'left'** - All blocks will move to the left if possible.
- **'right'** - All blocks will move to the right if possible
- **'up'** - All blocks will move to up if possible.
- **'down'** - All blocks will move to down if possible
- **'exit'** - Prompts a message confirming if they want to exit

3.4.4 Frozen Bubble

- **'fire'** or **'now'** - Will fire the bubble
- **'exit'** - Prompts a message confirming if they want to exit
- **'continue'** - Continues the launcher's auto-rotation after a pause.
- **'colorblind'** - Toggles colorblind mode.
- **'full screen'** - Toggles fullscreen mode.
- **'(don't) rush me'** - Toggles 'Rush Me' mode.

- **'about frozen bubble'** - Displays Frozen Bubble credits.
- **'new game'** - Starts a new game.
- **'help me'** - Displays a help screen.

3.4.5 Number Input

- **numbers 0 to 9 inclusive** – Appends the spoken number to the end of your answer, i.e.. Speaking 'one', 'two' will give you 12. Note that 0 is to be spoken 'zero'
- **'okay'** - Submits the current answer and exits Number Input mode
- **'back'** - Removes the last number from your answer
- **'clear'** - Removes the entirety of your answer
- **'cancel'** - Exits Number Input mode

When entering numbers, please wait until the number you have just spoken appears in the box before trying to input the next one. Good Luck!

4 Developer Documentation

4.1 Build Instructions

The project can be built using Android Studio standard settings.

4.2 Application Architecture

The application consists of multiple RecognizedActivities that interact with a Recognizer service with assistance from a SpeechResultListener.

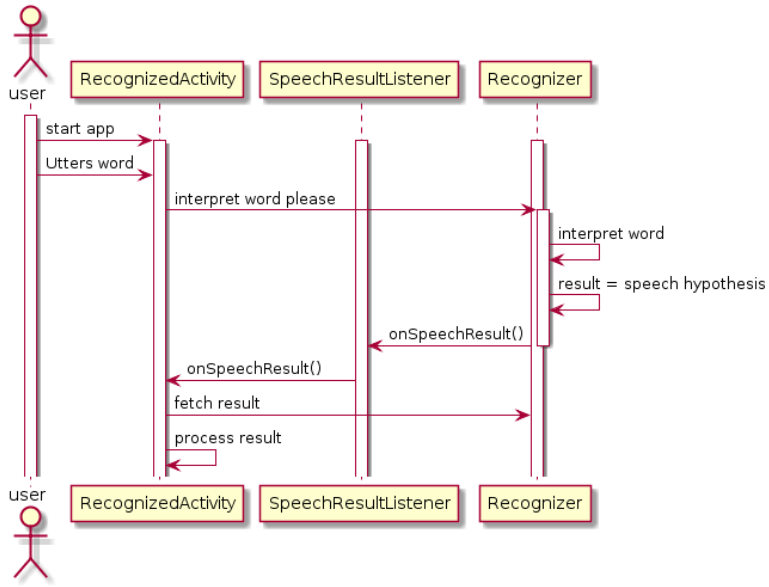

Figure 1: Interpreting a voice command.

4.2.1 RecognizedActivity

RecognizedActivity is an extension of AppCompatActivity that contains extra functionality for keeping the Recognizer service running in the correct mode throughout Activity switching and Dialog popups. In each implementation the method 'setup()' needs to be called in its 'onCreate()' method. Each RecognizedActivity needs to have an ImageButton in its layout with the id recognizerStatus and a listener that implements the method 'onSpeechResult()'. 'setup()' will add functionality to these for you as well as binding the Recognizer.

4.2.2 Recognizer

Recognizer is a Service that implements a CMUSphinx interpreter to interpret speech from a user and use the interpretations to control a RecognizedActivity. It is important that the Recognizer is able to switch between different modes of search to represent the RecognizedActivity that the user is currently interacting with. It does this by switching between 'grammar' files written in JSpeech Grammar Format. These files tell the interpreter what words and in what order it should expect them.

4.2.3 SpeechResultListener

SpeechResultListener listens for events in the Recognizer Service and informs the current RecognizedActivity that they have happened. It contains code that maintains the RecognizedActivity's recognizerButton such as swapping the image and initiating animations. The most important method is onSpeechResult() which informs the Activity that there is a new speech result for it to fetch. This method is abstract and needs to be implemented in each individual Activity and should contain code to handle each speech command the Activity can expect to find.

5 Open Source components

The code for the speech recognizer setup/permissions/asset copying is taken almost directly from the pocketSphinx demo application here: https://github.com/cmusphinx/pocketsphinx-android-demo

The phoneme definitions in our nktd.dic file are selectively taken from the cmudict-en-us.dic file in that same demo.

The acoustic model we use is provided by CMUSphinx at sourceforge.net

The game FrozenBubble taken from https://github.com/kthakore/frozen-bubble

The game 2048 taken from https://github.com/BuddyBuild/2048-Android