Embedded Systems 3
Individual Project Report

Konstantinos Dadamis
1002224d

March 14, 2012

Contents

1 Introduction

This document is the report of the individual project which was carried out as part of the Embedded Systems 3 course. This project consisted of developing an application on a portable device (mobile phone or tablet, iOS or Android). The application (app) I developed is a fight game (with minimum violence) called "Photo Fighter" and developed on Android 2.3.5 Gingerbread. Through this app, the user can create his own fighters by taking photos and recording voices and afterwards play the fight game with the characters he created.

The full functionality of the game is explained in section 2 where the reader can also find screenshots of the app. Section 3 contains the design of the system and explains how the functionality was implemented. Section 4 lists the aims contained in the project proposal that were achieved, those abandoned and the future additions that could be made to the game. Finally, the conclusions can be found in section 5.

2 User Interface

When the app is installed to an Android device, it is listed along with the other apps at the "All apps" menu. It has an icon named "Photo Fighter" (Figure 1).

When the user taps on the icon, a splash screen appears (Figure 2) with the logo of the app.

The splash screen lasts 5 seconds but if the user does not want to wait, a tap to the screen makes the splash screen go away and the main menu appears (Figure 3).

The main menu has three options which separate the functionality of the game. The "Fight" option contains all the fight–related functionality and the "Personalize" option contains the functionality which is relevant to creating new fighters and arenas. These two functionalities are described at the two following sub–sections. Finally, the "Help" option contains a list of instuctions the user must follow in order to benefit from the full functionality of the game.

2.1 Personalizalion

The "Personalize" menu (Figure 4) has three options. The user can personalize either the characters who will be fighting by pressing "Manage Fighters" or the arenas where the fight will take place by pressing "Manage Arenas". The user can also press "Back" to return to the previous screen.

By pressing "Manage Fighters" the user advances to the respective sub–sub–menu (Figure 5). There, he can choose either to create a new fighter or edit an existing one.

If he chooses to create a new fighter, the respective screen (Figure 6) appears where the user can enter a non–existing fighter name and advance to the "Insert Media" screen (Figure 7).

Figure 1: The Android home screen with the PhotoFighter icon

Figure 2: The splash screen

Figure 3: The main menu

Figure 4: The Personalize sub–menu

Figure 5: The Manage Fighters sub–sub–menu

Figure 6: The Create Fighter screen

At this screen, the user is able to insert his photos and voice recordings to the game in order to create his character. Initially, 8 cameras and 5 orange microphone icons appear which act as buttons. The user can press them and insert his photos and recordings respectively. Upon a successful insertion, the camera is replaced by the photo taken and the orange microphone is replaced by a blue one. All insertions can be changed by pressing the same buttons.

The game needs 8 photos of the user[1]standing in different positions for reproducing the fighter behaviour. The 8 positions are "Walk", "Stand", "Punch", "Kick", "Crouch", "Take hit", "Winning" and "Losing". When the user tries to take a certain photo, a margin is provided by the app so that the right photo is taken. The user is also notified by text about the position he should take.

Figure 7: The Insert Media screen

The "Voice Record" screen (Figure 8) allows the user to make a recording and review it before pressing back and continue to the next one. The recording can be overwritten if the "Voice Record" button is pressed twice. The recordings that must be made are when hitting, receiving a hit and winning (before the fight is over) and two more for victory and loss.

After the Personalize sub–menu (Figure 4) the user can also advance to the "Manage Arenas" screen (Figure 9) where he can add arenas by taking photos and edit them. Unfortunately, at this version of the app, managing arenas is not implemented.

2.2 Fight

When the user presses "Fight" at the main menu (Figure 3), he advances to the "Select Fighters" screen (Figure 10) only if there is at least one fighter inserted to the app through the personalize process described above. There, he has two

[1]A screenshot taking a photo could not be provided as the "Dalvik Debug Monitor" cannot take screenshots while in taking photo mode

Figure 8: The Voice Record screen

Figure 9: The Manage Arenas sub–sub–menu

same scrollable lists of fighters added to the game and he has to choose his fighter and his opponent which will be handled by the game. The entries of the fighters in the lists consist of their name and their "Winning" photo. When he presses on one fighter of a list, his name and photo appear above the list which means that this character was selected either as the user fighter or as his opponent.

Figure 10: The Select Fighters screen

When the user is ready, he presses on "Fight" and the fight begins (Figures 11 and 12). Each user has a life bar and when the fight begins it is full. When a user receives a hit, his life bar gets reduced by 10 or 15 percent depending on if he receives a punch or a kick respectively. The life bar color is initially green, gets orange when it drops to less than 75 percent and gets red when it drops to less than 25 percent.

The user can move his fighter by the up, down, left and right by pressing the cursors placed at the bottom left part of the screen. The punch and kick buttons are placed at the bottom right part of the screen. When a fighter moves left and right a fake sense of walking is produced by the game by interchanging the fighter's "Walk" and "Stand" photos. When up is pressed the fighter jumps and he can dodge kicks and when the user presses down, the fighter crouches and he can dodge punches. The fighters are limited to move inside the screen limits and they can change sides which makes their photos mirrored to face each other again.

The fight can last up to one minute. If some fighter's life bar does not get depleted by the end of this minute, then the user with larger life bar wins. If the two fighters have the same percentage of their initial bar, then we have a draw and noone wins.

When the game ends, the name of the winner appears at the center of the screen along with the winner's bouncing "Winner" photo and the loser's bouncing "Loser" photo (Figure 13). The relevant recordings of each fighter are also

Figure 11: Fight screenshot

Figure 12: Fight screenshot

played.

Figure 13: End of fight screenshot

3 Design features

This section lists the special design features used during the implementation of the functionality described at section 2. The code submitted along with this report is needed by the reader in order to understand the following paragraphs.

3.1 General

This subsection lists the special design features implemented at the initial screens and are part of the `uk.ac.gla.photofight` package:

- `SplashScreenActivity` is the activity behind the initial splash screen. It is implemented by creating a thread which waits five seconds and starts the "Main Menu" activity. This thread can also be interrupted when the user touches the screen which throws an `InterruptedException` and starts the activity without waiting.
- `Storage` is a special class with `static` functions is used which takes care of all the accesses to the SD card of the Android device.
- `Fighter` is a class which takes care of loading all the `Bitmap`s of a fighter and provides functions for mirroring the `Bitmap`s and loading the sound recordings. Its instance variables are all public as Google advises when using an object in animations where the accesses need to be fast without intermediate accessors.

- `FighterArrayAdapter` is an `ArrayAdapter` which takes care of listing the fighters with their name and photo. It is used at the `ListViews` in `EditFighterActivity` and `SelectFightersActivity`.

3.2 Personalizalion

This subsection lists the special design features implemented for the Personalization part of the app and are located inside the `uk.ac.gla.photofight.personalize` package:

- `CreateFighterActivity` is using `SharedPreferences` for passing the new fighter name to `InsertMediaToCharacterActivity`, `TakePhotoActivity` and `RecordVoiceActivity`.
- `TakePhotoActivity` was created by following the instructions for using the Camera provided by Google and modified where necessary. Photos are stored with a substantially lower quality (10 percent of the original quality) because of the limitations of memory in an Android device. For the same reason, each time the photos are loaded to `Bitmaps`, their size is reduced 8 times.

  Special `.png` images were used for filtering the photo which are located in the `res/drawable` folder with `bg` and `border` prefixes. These images act as masks for the photos. The `border`–prefixed ones are used for taking the photo and have their interior transparent. The `bg`– prefixed ones are used for removing the background of a photo and they contain the same images as the `border` ones but with the backgound transparent and the interior opaque. `Xfermode` objects are used for removing the background.

3.3 Fight

This subsection lists the special design features implemented for the Fight part of the app and are located inside the `uk.ac.gla.photofight.personalize.fight` package:

- `FightActivity` is a simple activity which uses the custom `View`, `FightView`. It also takes care of placing the buttons on top of the view and sending the touch events to the `FightView` object which also handles the animations.
- `ControlFighterThread` is the `Thread` which controls the fighter handled by the app and provides the Artificial Intelligence needed. The AI implemented is not high level (which was not needed for this project) and it contains a simple state machine which switches between four different states (`Plans`). There is a `DEFENSIVE`, an `OFFENSIVE`, a `NEUTRAL` and a `CRAZY` plan. Each plan has its own random set of moves and every five seconds, the thread changes `Plan` randomly. Only when the fighter controlled by the thread has less than 25 percent of his health, the `Plan` followed is switched permanently to the defensive one in order not to lose more health.

- `FightView` is a custom `View` which receives the user's moves from the `FightActivity` and runs the `FightThread` (which is an inner class). The `FightThread` uses the `SurfaceHolder` of the view which has the `Canvas` where the animation is drawn and draws every frame of the animation. It is also responsible for creating the AI thread and receiving movements from both the user and the AI thread. After receiving their movements, appropriate changes are made to the respective `Fighter` objects and the physics (gravity, movement speed) are updated in each frame. Again, almost no accessors and mutators were used in order to achieve more Frames Per Second.

4 Future Improvements and Extensions

Most of the requirements set at the proposal were implemented and the goals that were set were realistic. Nevertheless, some improvements and extensions to the functionality could also be introduced:

- "Manage Arenas" feature should be implemented which would allow the user to create his own arenas and edit them, This would fit very well to the game since its main feature is personalizing.
- Playing and recording sounds is sometimes buggy in Android Gingerbread, so the app could be updated to use a more recent Android release such as Android 4.0 Ice Cream Sandwich which could resolve these problems.
- A major feature that the app could have is multiplayer capabilities. This can be implemented either throught direct Wi-Fi in Android ICS or through an XMPP server.
- Score tables of players could be kept inside the game and possibly social networks like Facebook or Google+ could be used for posting these tables and advertising the app.

5 Conclusions

The development of this game has been a wondeful experience because it is developed on Android and Java and also because it is a project that I came up with. Its first stage of development is complete and I will probably spend more time in polishing it, checking how it behaves in other Android devices and adding more features during the summer. At that point, I might be able to publish it to "Google play" and hopefully get some downloads.