ANDROCHI

EVGENI KOVALEV PAVEL OSIPOV

CONTENTS

1. PROBLEM

The problem was to make the first software based on Android powered mobile devices in one week with no earlier experience in Android ever, having well planned lectures of Android guru from Digia and good consultations.

Seems like no big problem for two engineer students, who have good programming skills and visual eye. After planning for a while and measured what we already knew and what we need to learn for this project, we started to implement the Androchi project.

Along the project we planned to learn some basic Android skills with Android SDK like the use of basic objects as textboxes, dropdown lists and others. For sure, we had to try some database things with SQLite.

After that we came up with a simple toy, which reminds the Japanese Tamagotchi, where user must train, feed, teach and keep his virtual pet satisfied.

2. Solution

First we learned the Android basics, after that we tried to implement some basic things binding them with already known java skills. We also made some sketches of which kind of software we are going to deal with.

Finally we came up with the tasks and their priorities:

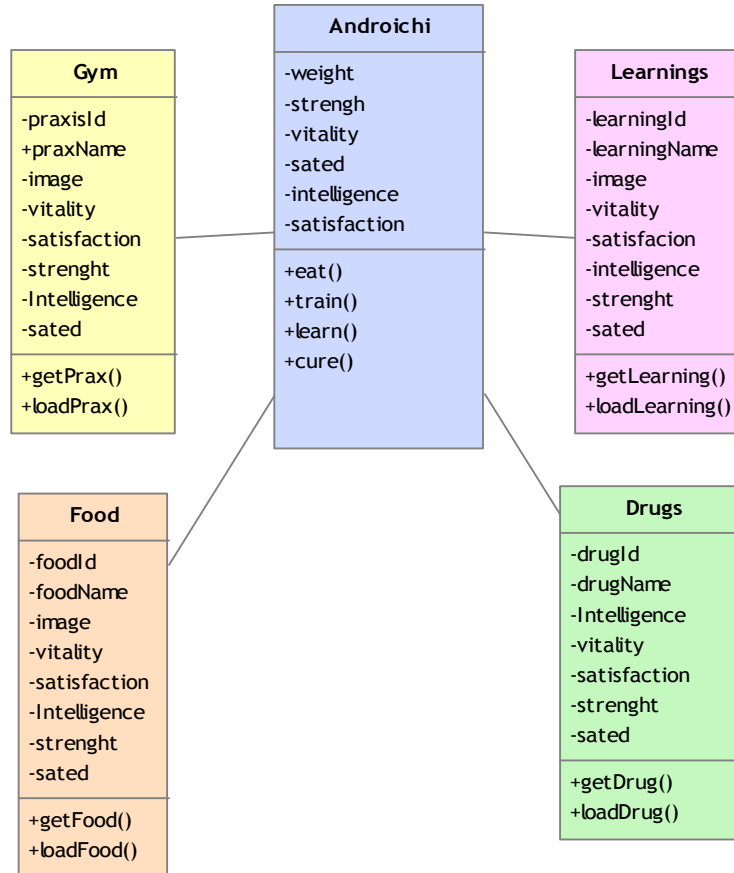| FEATURE	PRIORITY
Learn Android basics	1
Artifical intelligence classes	2
SQLite issues	3
Interface design	4

3. APPLICATION FUNCTIONALITIES

The application operates on Android based mobile device and uses real time for spending out resources. User follows up resource statistics of his pet and gives him food, education, drugs and gym practices depending on which one of them is decreasing. All resources are being retrieved from SQLite database and being filled up as user gives some resource. Every piece of recourses contains the following multipliers: vitality, weight, satisfaction and satiety. The following example shows a piece of resource with its multipliers:

As user gives Androchi a piece of cake, the eating animation rolls on and statistic bar of sated increases by 5%, satisfaction bar increases by 10%, weight bar increases by 3% and vitality bar increases by 2%. Otherwise if user gives Androchi to read the Windows manual satisfaction and vitality bar will decrease by 10% and 5%.

4. IMPLEMENTATION TECHNOLOGIES

First we implemented some AI classes and inherited them in our main class as the following class diagram shows:

Gym
-praxisId
+praxName
-image
-vitality
-satisfaction
-strenght
-Intelligence
-sated
+getPrax()
+loadPrax()

Androichi
-weight
-strengh
-vitality
-sated
-intelligence
-satisfaction
+eat()
+train()
+learn()
+cure()

Learnings
-learningId
-learningName
-image
-vitality
-satisfacion
-intelligence
-strenght
-sated
+getLearning()
+loadLearning()

Food
-foodId
-foodName
-image
-vitality
-satisfaction
-Intelligence
-strenght
-sated
+getFood()
+loadFood()

Drugs
-drugId
-drugName
-Intelligence
-vitality
-satisfaction
-strenght
-sated
+getDrug()
+loadDrug()

After that we worked on SQLite server and database adapter. As connection to database was established, we started to work on SQL – statements and databases.

As we made database aspects work, we moved to animation part. We modeled the Android character with 3D Studio and animated it for every action of our software. Within help of our Android guru we managed out how to make animations for Android using xml – file.

After making some menus and placed the resource items in them, we finally got the first working version of our software.

5. DESCRIPTION

Finally we got the mobile software, which could be described as following:

Androchi is an artificial intelligence (AI) software for Android powered mobile devices, that intends as a digital pet for children, which could be kept satisfied within few aspects as vitality, strength, satisfaction and satiety. Androchi is a very simple toy which could be used as a pet simulator for children who want to get a real pet in future. The satisfaction aspects of Androchi are engaged to real time and decrease along the time.

6. ARCHITECTURE

The Androchi mobile software uses SQLite database and spends resources from it on real time. User fills up resources and updates the database.

7. WORKGROUP

Evgeni Kovalev (0500765) Saimia University of applied science

Pavel Osipov (0600767) Saimia University of applied science