If somebody wants editing rights, let me know. You can suggest changes and new ideas in our forum or IRC. - deliwien

Table of Contents

1. Introduction

1.1 Purpose and scope

The purpose of the app is to work as a starter project for Reddit Android Developers. The application is going to be open source, free, and it's going to be published on Google Play.

1.2 Product and environment

The product is an improved version of a Tamagotchi. It's going to be an app for Android phones and tablets. The user has to take care of a creature or it will get sick or die. When the user starts the game, the creature is going to be a baby, and if it's being taken care of properly, it will grow until it will die of old age.

The theme of the app is aliens, all of the creatures will be aliens, and the theme needs to be taken into consideration when designing the UI and the sounds.

We're using libgdx as our game engine.

1.3 Definitions, acronyms and abbreviations

- Tamagotchi: A handheld, digital pet.
- RAD: Reddit Android Developers
- Pockaz: A name for the application, which was chosen in a name vote.

1.4 References

- Instruction booklet for Tamagotchi
  http://www.bandai.com/support/pdfs/tama/tama_v3.pdf
- GUI design standards: http://developer.android.com/design/index.html
- libgdx: http://code.google.com/p/libgdx/
- A simulator of the original Tamagotchi:
  http://www.redditandroiddevelopers.com/downloads/tamagotchi/tamagotchisimulator.zip
- Redmine: http://redmine.redditandroiddevelopers.com/

2. General description

2.1 Product functions

At the beginning, the user will create a creature. The creature needs to be taken care of, or it will get sick and die. If the creature has died, it cannot be revived and the user has to start all over. The app will have sound effects and music, which the user can disable if he wants to.

A more detailed description of the functions can be found at the section 4.

2.2 User characteristics

The user owns an Android phone with a touch screen. Maybe he has used a real Tamagotchi before.

2.3 General constraints

The phone has to have at least Android version 2.2.

2.4 Assumptions and dependencies

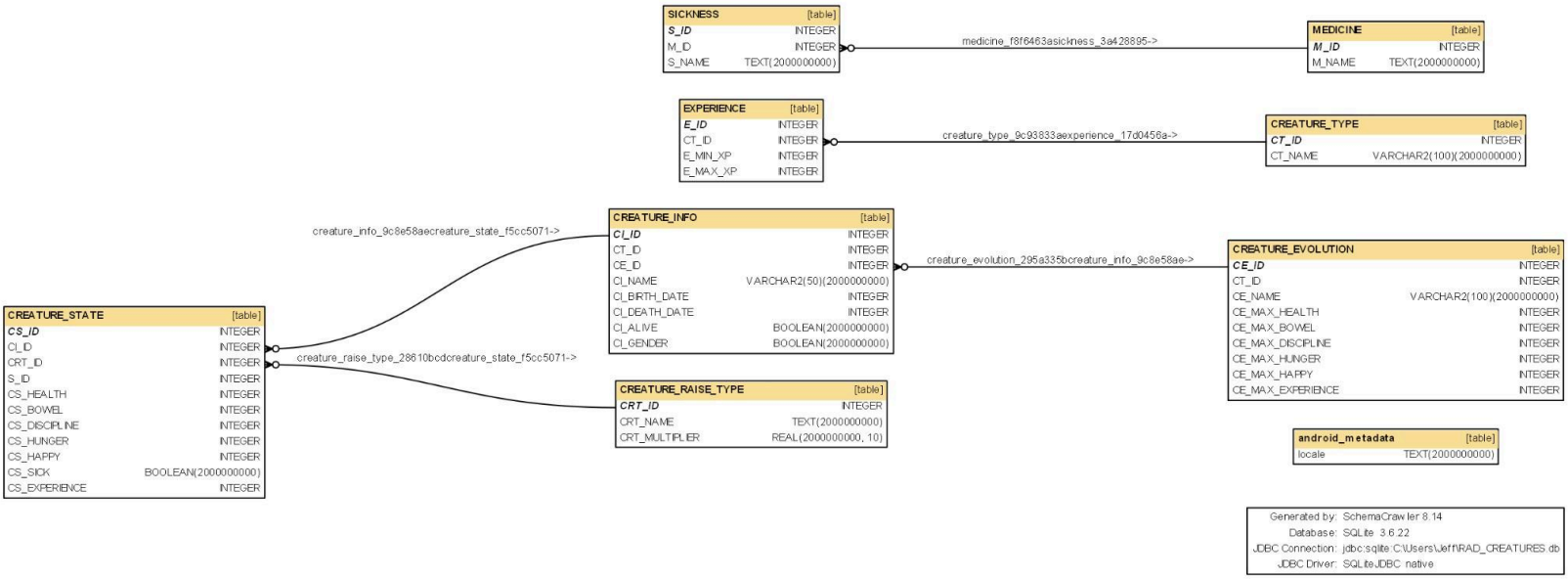
The layout needs to be created for both horizontal & vertical.

3. Data and databases

3.1 UML diagram of the database

An image with a better resolution can be found here:

http://www.redditandroiddevelopers.com/files/ (Misc → Database diagrams)



3.2 Contents of information

A description on how the data will be stored to the database is needed.

3.2.1 Creature

3.2.1.1 Age

The creature is going to age at some speed.

3.2.1.2 Gender

Male / Female

3.2.1.3 Name

The creature needs a name. This is given when creating the creature.

3.2.1.4 Hunger

A value which tells the creatures current hunger level. If the creature is hungry for too long, it can get sick, lose Health (hit points) or even die. If the creature gets too much food, it will gain weight and too much weight will lead to health problems.

3.2.1.5 Health (hit points)

Amount of health the creature currently has. It can decrease if the creature is hurt or ill. It can increase if the creature is feeling well for a certain amount of time. The total amount of Health (hit points) can also be increased if the creature is feeling well over time.

3.2.1.6 Cleanliness

A value which tells the creatures current cleanliness level. If the creature is dirty, it's more likely to get sick and be less happy.

3.2.1.7 Happiness

A value which tells the creatures current happiness level. It can increase through playing with the creature, feeding it treats, etc.

3.2.1.8 Alive

A boolean which tells if the creature is alive. False = dead, true = alive.

3.2.1.9 Sleepiness

A value which tells the creatures current alertness level. If the creature's sleepiness is low, the creature's reaction time and movement will be slowed down. The creature could faint.

3.2.1.10 Skills

The creature can have special skills if he's trained accordingly.

3.2.1.11 Diseases

If the creature is sick, it has a disease of some type. A disease can be cured with some kind of a medicine. This can be diagnosed through a creature doctor/specialist where s/he recommends getting this specific medication where the user chooses the dosage, etc. Wrong medication will cause problems.

3.2.1.12 Race

This field tells us what kind of a creature it is. Different races have different attributes, for example some races don't like to get wet, some races can age at different speeds, some races like to misbehave more than the others and some races tend to get hungry very easily. Some races can also have special diseases.

3.2.1.13 Weight

The creature can be thin, normal or overweight. If it is too thin or too overweight, it's going to die. Creatures with weight issues are also more likely to get sick. How much the creature weights will affect their speed, agility and other factors when playing games, i.e. racing.

3.2.1.14 Discipline

If the creature behaves badly, it needs to be disciplined.

4. Functions (requirements)

about every function: purpose, inputs, handling, outputs, error handling, user interface, menu/screen hierarchy event list. UI guys / graphic designers, we're going to need your input here. We're also going to need animations for each of the functions.

4.1 Creating the creature

Have several different species of pet to choose from. From there, perhaps you are able to choose some of its basic characteristics or alternatively you are simply given an egg which hatches to a random formation of the species you chose.

4.1.1 Giving a name to the creature

I'm thinking of an interface where you would be looking at a collar tag for the pet, this screen can also keep track of its statistics. The keyboard used for typing the name will be whatever the devices main keyboard is, no use in implementing a keyboard with some skin on it.

4.1.2 Choosing the race of the creature

Eventually, varying shapes/sizes of ears/tails/fur/color. First we're going to start with just a couple of different basic creatures.

4.1.3 Choosing the gender of the creature.

The creature can be male or female. The user has to choose which one he wants to create. The females can have a bow on their head or some other kind of graphics can be used.

4.2 Main screen

4.2.1 General description

Main screen is the place where player spends most of his time. The creature will walk around the screen, there will be idle animations, and the user can take care of his creature here.

A prototype of the main screen: http://dl.dropbox.com/u/3325290/MainScreenUI.pdf

4.2.2 Feeding the creature

If the creature is hungry, it needs to eat. If it doesn't eat often enough, it will get too thin and eventually die. If it eats when it isn't hungry, it's going to gain weight, possibly get sick and die. Balance is key. The user has the option of what kind of food to give the creature, at what amount, at what style and how (whether to just slap in on a bowl and serve or make the creature do tricks to get the food or throw it in the air where the creature can jump to get it).

4.2.3 Washing the creature

If the creature is too dirty, it needs to be washed. If the user ignores it, or is late to the cleaning, the creatures happiness would decrease and can potentially catch a disease or fleas. There would be options on what to use to clean the creature and how, the user intereacts with the cleaning items to clean the creature.

4.2.4 Exercising the creature

Needs to take a jog, or do some squats otherwise it will get fat, sick, and die. Training exhausts the creature and it will get hungry and/or tired.This is where attributes can increase. Exercising can be as simple as playing with the creature (let's say fetching a ball) or as complex as playing a game where there could be awards.

4.2.5 Giving medicine to a sick creature

Different types of medicines / illnesses? Simple cold to a heavy influenza, cough medicine and much rest is needed. Also it might not want to eat if its sick.

4.2.6 Putting the creature to bed

The creature needs to rest when it's sleepiness gets too high. An initial starter bed will be given, where the user has the ability to upgrade the bed, the home, the stuff on the bed and inside the house, etc.

4.2.7 Playing with the creature

Playing with the creature makes it happier and makes it lose weight. We can have multiple different mini-games from which the player can choose how he wants to play with his creature. Playing exhausts the creature and it will get hungry and/or tired. Games can include basic racing, retrieval of a ball/object thrown somewhere and more.

4.2.8 Discipling the creature

If the creature behaves badly, it needs to be disciplined. It can also be praised if it is behaving nicely. Appearance and behavior might change based on the the disciplinary actions taken. If it is being bullied by the user, it will look scared and sad, and might get sick and die. Failing to discipline the creature when it has done something bad, might have long-term repercussions .

4.3 Death of a creature

If the creature has died, it can not be revived. Some kind of statistics about the creature will be shown to the user. The only option is to start the game again by creating a new creature.

4.4 Starting over

The user can delete his creature and start again whenever he wants to. This has to be verified so it can't happen on accident. Possibly by password, at least something more complex than a yes/no.

4.5 Changing layouts

When changing from a vertical layout to a horizontal layout (or back to vertical layout),

something funny happens, like throwing the creature around the screen or something.

4.6 Tutorial

Some kind of a mascot / helper creature, which helps the user while creating a new creature / learning how to take care of his creature.

4.7 Upgrades

The user should be able to “buy” upgrades for the creature’s home: toys, more space or a window maybe.

5. Program behaviour

This chapter will describe when certain things from chapter 4. are shown or executed.

5.1 Splash screen

- Splash screen should be shown for a certain time. The AssetManager will start loading at this point.

5.2 Main menu

- On the first start, the user should be asked if he wants music/sound effects BEFORE any sounds are played.
- The menu is shown and lets the player choose what to do next.
- AssetManager is still loading and should finish sometime during this screen.

5.3 Creature creation screen

- A text field is shown where the user can write a name for the creature.
- Some kind of a race selection option is needed, so the user can choose a race for the creature. Different races will behave differently.
- The user needs to choose a gender for the creature.

5.4 Creature selection screen

- If the user has multiple creatures, he needs to select which creature he wants to play with.

5.5 Creature main screen

- **Idle animations**

    - I want an animation where the creature puts his thumb in his mouth, starts blowing, becomes bigger and bigger every time he breathes out and at some point he can't keep the thumb in his mouth anymore and starts flying around the screen like a balloon. (member68)
    - Creature scratches himself
    - Creature sneezes, yawn, rolls over, falls asleep
    - Creature walks around and might trip over his own feet from time to time
        - Maybe use a reference to Dinner for one
    - Creature starts jumping on the spot
    - Creature reads a book.
    - Creature plays with a Tamagotchi. (Tamaception!)

5.6 Settings screen

- Deleting a creature
- Enable / disable notifications

6. Changing the attributes over time / Formulas

In this section, we should define the formulas needed for changing the values of the attributes, such as hunger, sleepiness and so on. Feel free to add sections when needed.

6.1 Hunger

Will be developed in later sprints.

6.2 Sleepiness

Will be developed in later sprints.

6.3 Age

Aging is the same as leveling. Feeding, playing, cleaning etc the creature gives X experience points. Each minute (? or hour?) gives X experience points. When the total experience points hits Y the creature reaches level / age Z. This means that each action (feeding etc) needs to be assigned X worth of points. Also Sickness, hunger and other negative statuses lowers the total experience / level cap. So if you create a creature it will have a random (or static for that matter) level cap, which we can subtract points from (which makes it reach different max ages before it dies, depending on the creatures total health during its life).

6.4 Cleanliness

Will be developed in later sprints.

6.5 Happiness

Will be developed in later sprints.

7. Other requirements

7.1 Performance and response times

The animations need to look smooth, and the FPS needs to be high enough.

7.2 Security, recovery, availability

When high score tables are implemented, some kind of an online database is needed to save the scores so people can't cheat.

7.3 Maintainability

The app will receive updates as new features are developed.

7.4 Transferability/portability

Per now: none, only works on Android. (And there is also a desktop version for the developers)

8. Graphics

8.1 Graphics style

The Art Style, which has been discussed on our IRC, is going to be a 2D, cute, cartoon art style. We choose this art style because it is easier to execute and this kind of art style has been successful with games such as Angry Birds, Cut the Rope and Where's my Water? Also, this allows us to stylize the art design and would be visually appealing and attractive to our main audience. Also, less is more with cartoons.

The theme of the app is aliens, so this should be considered when designing the graphics.

9. Design constraints

9.1 Standards

The GUI Design Standards of the app follow the Android Design Guidelines which can be found here: http://developer.android.com/design/index.html. The designers who take the role of GUI design should read this and understand the concepts of the guidelines. This applies to mostly to creating icons. The app itself won't use the GUI standards, which have been created for apps, not games.

9.2 Hardware constraints

The product has to work with a phone or a tablet with Android version 2.2 or newer and a touch screen. All screen sizes should supported (handled mostly by our framework: libGDX)

9.3 Software constraints

Minimum Android version: 2.2.

9.4 Other constraints

10. Rejected ideas

10.1 A tree Tamagotchi

"Heres my idea for a Tamagotchi-like game: Instead of having a animal which you nurture, we follow the life cycle of a tree. As we all know, trees can get really old, so the game could take a while. Just like the normal Tamagotchi.

We start out by "eating an apple" and then spitting the seeds on the ground (for planting!). We then have to dig a hole for the seeds, put them in and cover it back up. Alternatively we can just have the seeds on the ground and we then have to fight off birds while it "grows". We then have to water it regularly and wait patiently for it to grow into a little twig. We now have to water and "feed it" (by feeding i mean fertilized with manure). You also have to guide the roots down to get a good grip, so it won't blow over.

Possible "danger scenario" later is a dry summer, and the tree have to find water in the ground. When the tree finally has grown into a "small tree", flowers start to pop out. And we need to "fertilize the flowers" by clicking on the flowers, signaling a bee or a bumblebee to come over and visit the clicked flower. Danger: BEWARE OF THE WASP! Also we need to watch out for

snails or lice. This flower in time grows into a fruit (crazy, i know), say an apple (since it was apple seeds.. :D). Not every flower grows into a fruit. And now we have to watch out for birds, punching them with your finger if they appear. Also worms can be a problem.

Eventually the apples rot and fall down. Which in turn fertilizes the tree. Tree trimming might also be needed so that the branches or fruits don't die. Ok what happens if you can't fight off the different types of dangers? Well maybe your leaves, flowers and fruits fall off. Maybe a twig or branch breaks, maybe it starts to burn, dries up, gets eaten by insects, blows over (because you forgot to extend your roots), or something else entirely."

10.2 Andengine as the game engine

The idea was to use Andengine as the game engine. After a long discussion with the developers and trying it out, it was clear that it wasn't the right choice because of the lack of documentation. Libgdx took its place as the game engine.

11. Ideas for further development

11.1 Customizing the character

The creature could be customized by creating the creature from different parts, such as hands, legs, bodies, heads and accessories. (hats!) Also, the home of the creature can also be customized such as how big the home is to what is inside of it.

11.2 Achievements

- Killing your first creature (on purpose).
- Getting a rare disease.
- Viewing the credits after a character has died (see section 11.4 Easter eggs)

11.3 Hardware Use

I was thinking of doing something similar to the Pikachu Tamagotchi. The unit contained a pedometer which would give more rewards depending on how many steps you took, encouraging exercise.

11.4 Easter eggs

- Reddit alien & Android alien as creatures
- One of the easter eggs would be unlocked after the credits have been viewed after a character has died

11.5 Graveyard / heaven Screen

A graveyard screen so that you can view all the tombstones of your dead pets, the tombstones will have various stats of the pets when they died.

Alternate idea, if graveyard screen proves to be too depressing, we can have a heaven screen. Where you can see all the old pets frolic around, you can click on one of them to view stats.

11.6 Adoption center for mistreated pets

Pets can move out if you treat them badly. online based of course with a trading system. "Get your pet back from the orphanage by completing this mini game"

11.7 Multiplayer

Have friends' pets visit yours and play together. This can occur through Bluetooth, WiFi LAN or online multiplayer (if we get servers, etc.)

11.7.1 Breeding <3

Have the pets mate with each other. Contacting other creatures in other phones? Visiting them? Making relationships with them?

11.7.2 Games

Play multiplayer games with others.

11.7.3 Battling the other creatures (suggested by balomus on Reddit)

Reddit link "I would love something like this, but with the old school Digimon handheld things that were a brick wall. Especially if you could battle with your friends who had the app like the good old days."

11.8 Petting the creature

The user should be able to pet the creature by moving his finger on it. The creature should move accordingly.

11.9 Social media support

Tweeting about how the pet is doing and other things of that nature. Facebook?

11.10 Widget

A widget, so the user can see his creature and interact with it on the homescreen.

11.11 Virtual photo album with your pet

Add a virtual camera that captures important moments from the life of your pet. You can later view those photos, add captions and share them via 10.9 social networks or any other Android share option.

11.12 High scores

Allow people to upload high scores to a global ranking system. The high scores can range from longest living creature, most healthy, most sick, the creature with the largest home, etc.

11.13 Anti cheat system

Add a online auth system, to stop people from editing their database (only for users with rooted phones). What could be done is to make this game require the user to create a user account where their creatures are synced to it. This is not a requirement but this could prevent cheating for users who use an account. This can also allow the user to access their creatures on different devices.

11.14 Ask the user before playing music/sound

The user should be asked if he wants music before the music starts playing. A good place to ask this would be before entering the main menu. The choice should be remembered after that. The playing of the music/sound can be toggled on or off (mute) in the menu during gameplay, where under Options, is the more advanced music/sound options such as music/sound volume.

11.15 Things to avoid

http://www.reddit.com/r/Android/comments/sc8vv/what_makes_you_hate_a_game_app/

11.16 Hardcore/Classic mode (suggested by serrghi)

The original Tamagotchi was a bit moody and could die anytime if you weren't paying attention. A similar "mode" could be added to the game to make it more difficult.

11.17 Show current weather in game

The current weather of the user could be shown in the game and could influence the creature. Cold temperatures could make it sick, etc...

11.18 Custom creature colors

We could allow users to select their creature's color. Every color that isn't black could be changed like this. Note: This affects the entire texture, not just certain parts.

12. Open issues

12.1 Creatures

We have to figure out how many creatures we're going to have and how much customization we're going to do.