

KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATICS

T120B169 App Development for Smart Mobile Systems

*Tamagochi Project (temporary)*

|  |
| --- |
| *Group, Name and Surname:*  *IFF-8/8, Simas Kazakevičius*  *IFF-0/8, Ignas Banevičius*  *MER-1, Mark Travner* |
| Date: *2021.09.13* |

Kaunas, 2021

Tables of Contents

[Description of Your app 3](#_Toc49939969)

[Functionality of your app 4](#_Toc49939970)

[List of functions (adapt to your own app) 4](#_Toc49939971)

[Solution 5](#_Toc49939972)

[Task #1. Create a menu that is accessed by swiping. 5](#_Toc49939973)

[Task #2. Add the ability to feed the pet. 6](#_Toc49939973)

[Task #3. Petstats decrease over time. 7](#_Toc49939973)

[Task #4. CPet stats open when reopening the app. 7](#_Toc49939973)

[Task #0. Create a menu that is accessed by swiping. 0](#_Toc49939973)

[Reference list 4](#_Toc49939974)20

# **Description of Your app**

1. What type is your application/game?

[Game] Tamagotchi (with different animals, different mechanics, use notification/reminders, must use animations, with different difficulty, use phone sensors, with highscore shared in some leaderboard)

Will implement databases for leaderboards and saving player data. Might find use for services, player preferences and Google login.

1. Description.

Tamagotchi type of game where you take care of your pet, feed it, bathe it, pet it, etc.

# **Functionality of your app**

## **List of functions (adapt to your own app)**

1. Create a menu that is accessed by swiping.
2. Add the ability to feed the pet.
3. Pet stats decrease over time.
4. Pet stats update when reopening the app.

# **Solution**

## **Task #1. Create a menu that is accessed by swiping.**

The user is able to access a menu by swiping up from the bottom or left from the right side. Implemented by giving the main layout a custom OnTouchListener and Overwriting a OnScroll function.

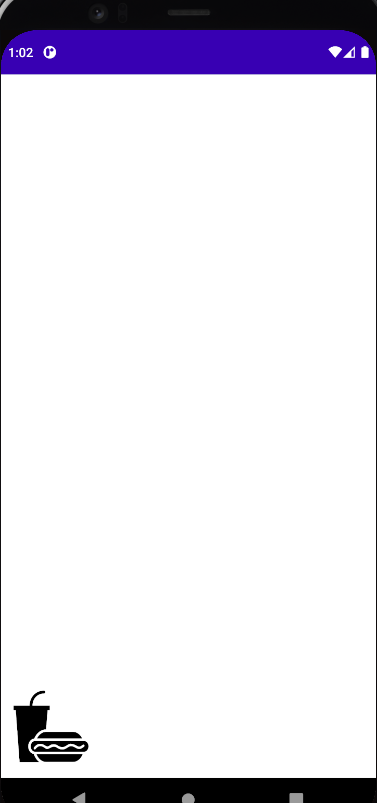


Figure 1. Food menu

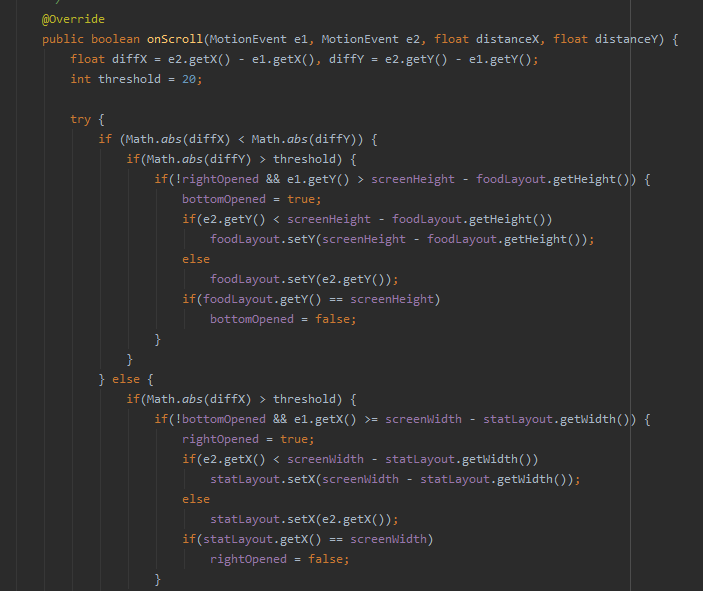


Figure 2. Food menu code fragment

## **Task #2. Add the ability to feed the pet**

The pet can be fed by accessing the food menu, long clicking the food icon then dragging the food onto the pet. This was done by creating a custom OnDragListener.



Figure 3. Food drag and drop



Figure 4. Drag and drop code fragment

## **Task #3. Pet stats decrease over time.**

When the app is active pet stats decrease and that information is displayed. This was done by setting up repeated timers.

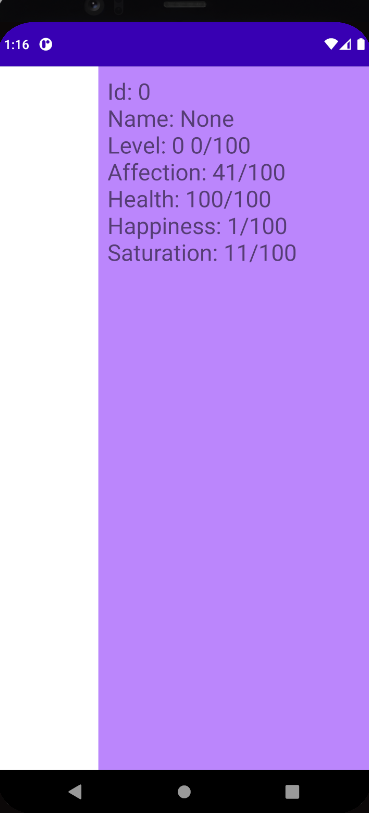


Figure 5. Stat menu

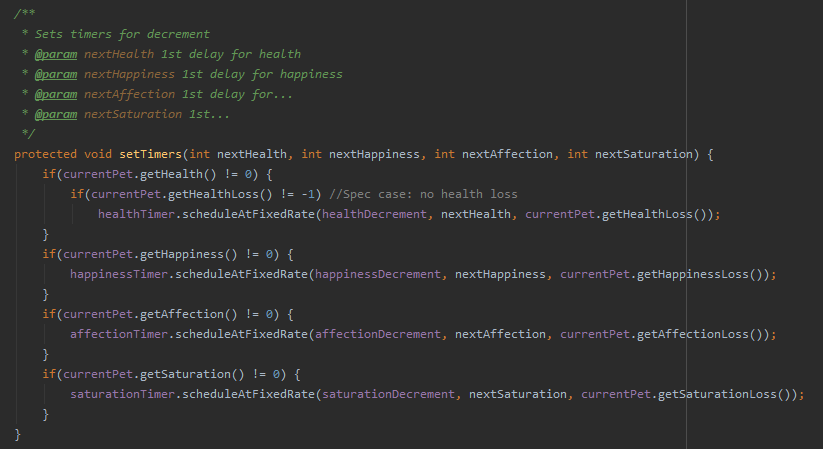


Figure 6. Stat menucode fragment

## **Task #4. Pet stats update when reopening the app.**

App saves your last login time. When reopening the app it calculates how many stat points were lost. This was implemented using shared preferences, dates and basic calculations.



Figure 7. App reopening stat calculation fragment

(Example task template, delete before submitting)

## **Task #1. Remove a UI component from the activity when a button is clicked.**

Description of the implementation (3-5 sentences). *Vestibulum hendrerit felis at turpis ultrices imperdiet. Nulla facilisi curabitur vitae semper nulla. Etiam rhoncus orci dolor [1], ac dictum erat iaculis sed. Aliquam pulvinar viverra consequat. Nam eu mi in mauris semper pellentesque eget ut erat [2].*

Screenshot

Figure 1. Screenshot #1

Each main function should be illustrated with the source code FRAGMENTS;

Fragment of Source Code

Figure 2. Source code #1

If you have used any external content or resources, make sure to refer to it [1]. Otherwise it will count as plagiarism.

# **Reference list**

1. Source #1. https://developer.android.com/reference/java/util/Timer
2. Source #2. https://developer.android.com/guide/topics/ui/drag-drop
3. Source #X. Url