ST10445672 Graeme Salvage IPRG511: A1 05/04/2024

ACADEMIC HONESTY DECLARATION

Declaration	**SIGN**
I have read the assessment rules provided in this declaration.	✓
This assessment is my own work.	✓
I have not copied any other student's work in this assessment.	✓
I have not uploaded the assessment question to any website or App offering assessment assistance.	✓
I have not downloaded my assessment with any other student.	✓
I have not used any AI tool without reviewing, re-writing, and re-working this information, and referencing any AI tools in my work.	✓
I have not shared this assessment with any other student.	✓
I have not presented the work of published sources as my own work.	✓
I have correctly cited all my sources of information.	✓
My referencing id technically correct, consistent, and congruent.	✓
I have acted in an acadeicaly honest way in this assessment.	✓

Report

Introduction:

The provided code is for an Android application developed in Kotlin, witch replicates the famous Japanese Tamagotchi game. The application prompts users engage with the app to maintain their pet. Additionally, it incorporates visual feedback to the user in various forms.

Functionality:

The app was made so users can easily interact with the pet to keep all the pets statistics as full as possible. If the user inter acts with the pet the appropriate action is shown by the pet as well updating the correct information fields.

How the app works:

Error Handling:

Errors happen when the user does not attend to the dog. The user is then encouraged to interact with the app by prompting the user with a message, giving the user some suggestion.

Data Retrieval:

Because of the nature of the application all relevant data in puts are received by button inputs. Then depending on the button, an image is displayed and statistics are updated.

UI Updates:

Every button click updates the UI in a different way. While the second screen is open the pets statistics decrease over time. This also means that the progress bars are also changing constantly.

btnContinue:

The only function of this button is to go from the home screen to the second page where the user can interact with the pet.

btnExit:

The functions of this button is to stop and reset the pet on the second page and to take the user back to the home page.

btnFeed:

When this button is clicked the pets hunger increases by 50 and decreases the cleanliness and happiness by 10.

The image showing the pet also shows the pet eating then going back to the waiting image.

The progress bar and the text is then updated accordingly

btnClean:

When this button is clicked the pets cleanliness increases by 50 and decreases the hunger and happiness by 10.

The image showing the pet also shows the pet bathing then going back to the waiting image.

The progress bar and the text is then updated accordingly btnPlay:

When this button is clicked the pet’s happiness increases by 50 and decreases the hunger and cleanliness by 10.

The image showing the pet also shows the pet playing then going back to the waiting image.

The progress bar and the text are then updated accordingly.

The i object:

This is used to check the pet’s statistics and the lower it over time. All the statistics of the pet gets decreases by 1 every 3 seconds.

The j object:

This is used to change the pets state from what it currently is to the appropriate image.

If any of the pet’s statistics are 0, the pet is displayed as sad.

If none of the pet’s statistics are 0, the pet is shown as happy.

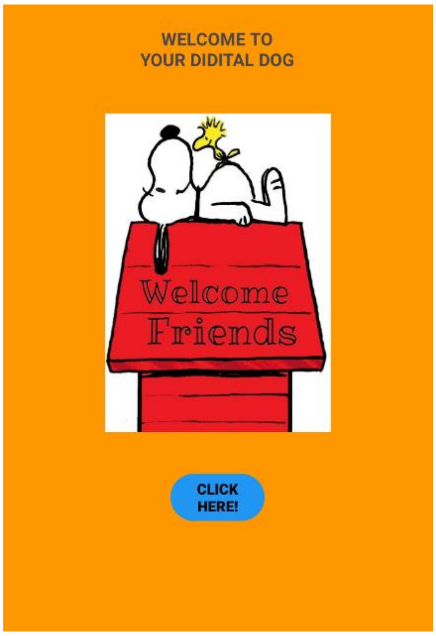
User interface design:

The design of the UI is targeted to entice interaction from the user.

The home screen colours, and image creates a feeling of joy and curiosity.

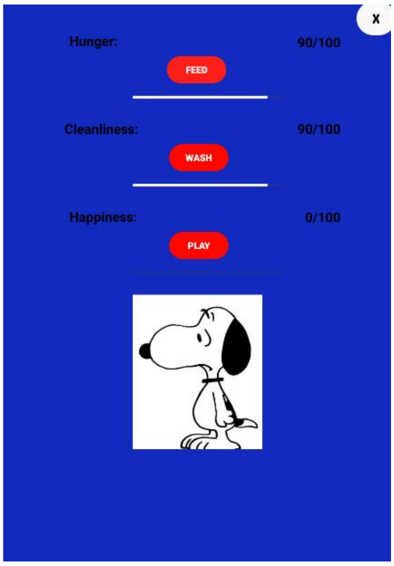
The second page colours create a sense of calm whilst the progress bar and text, showing the pets statistics ticking down, create excitement and keep the users attention.

The changing images the adds a sense of personalisation and ownership so the user feel responsible for the pet.



WELCOME TO
YOUR DIDITAL DOG

The above image shows the welcome page containing the continue button and welcoming the user.

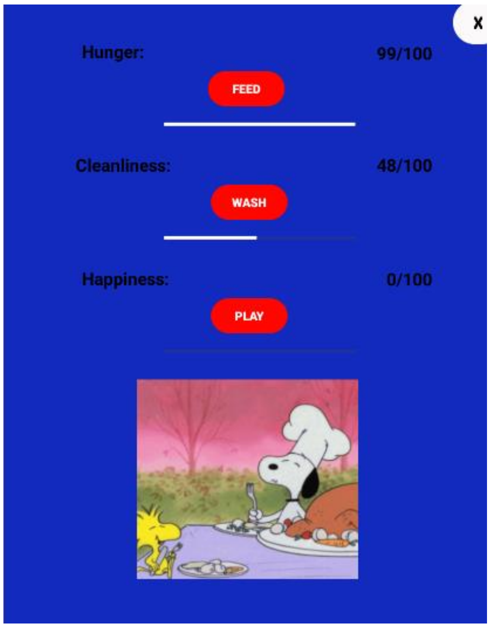


Hunger: 90/100

Cleanliness: 90/100

Happiness: 0/100

The above image shows the second page. The page starts off with the happiness at 0 which means the pet is sad, prompting the user to interact with the app.

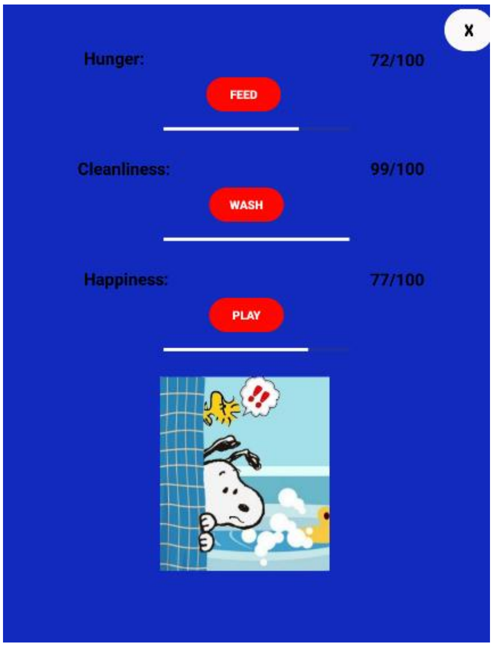


Hunger: 99/100

Cleanliness: 48/100

Happiness: 0/100

The above image shows the second page after the user clicked the feed button. The image now shows the pet eating, the hunger level is increased by 50, cleanliness and happiness are decreased by 10 and the progress bars are updated.

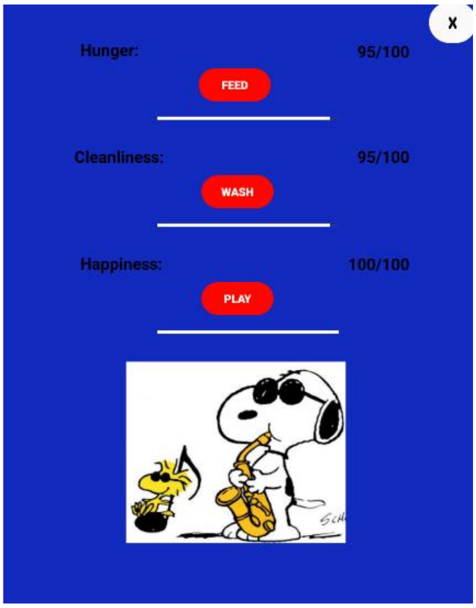


Hunger: 72/100

Cleanliness: 99/100

Happiness: 77/100

The above image shows the second page after the user clicked the wash button. The image now shows the pet bathing, the cleanliness level is increased by 50, hunger and happiness are decreased by 10 and the progress bars are updated.



Hunger: 95/100

Cleanliness: 95/100

Happiness: 100/100

The above image shows the second page after the user clicked the play button. The image now shows the pet eating, the happiness level is increased by 50, cleanliness and hunger are decreased by 10 and the progress bars are updated.

Conclusion

In conclusion, the provided code implements a solution that engages users with an interesting design and an interactive pet. This app replicates the famous Japanese Tamagotchi game in an unique way.

Reference list

All the above mentioned coding was done in:

Google. 2013. Android Studio. 1.9.22.[App](Accessed 22 April 2024).

Screenshots were taken in:

now.gg. 2011. BlueStacks. 5.20.10.1003. [App](Accessed 22 April 2024).

Video demo link

https://youtu.be/Fu59nPT5vpo

GitHub link

https://github.com/VCST10445672/ST10445672_IMADA2/tree/master