11/4/2023

Fish and Sleep – Using Gamification to Enforce Consistent Sleep Schedules for Those with Mood Disorders

Jacob Bradford – 2011172
BSc Computer Science at the School of Computer Science, University of Birmingham. Supervised by Dr Alexander Krull.
Word count is 9095.

Contents

Abstract

Those with mood disorders struggle with both going to sleep and waking up and common solutions to these problems are less effective with depressed patients. Gamification (applying features of games to other activities) was identified as an area with few existing implementations to assist sleep and fewer targeting depressed people. A phone application was made that encouraged users to wake up by allowing them to catch virtual fish for a short period after a defined time. Caught fish were stored in an aquarium and would become progressively

‘unhappy’ while the phone was active past a defined sleep time. The app would be successfully released for general use despite missing some features defined in the specification. An experiment was performed to examine if gamification could help maintain a consistent sleep schedule, but poor management led to it being rushed and little data was found to support the hypothesis.

Glossary

Sleep inertia: “A transitional state of lowered arousal occurring immediately after awakening from sleep and producing a temporary decrement in subsequent performance” (Tassi, 2000)

Cognitive Behavioural Therapy (CBT): A talking therapy that can help people manage their problems by changing the way they think and behave. (NHS, 2022)

Cognitive Behavioural Therapy for insomnia (CBT-I): CBT combined with treatment for insomnia, encourages the use of actions like; sleeping at a regular time, going to bed only when sleepy, never lying awake in bed etc. (Walker, 2017)

Gamification: “a process of enhancing services with (motivational) affordances in order to invoke gameful experiences and further behavioural outcomes” (Hamari, 2014)

Introduction

This project aims to help people with mood disorders to wake up in the morning through a gamified application which provides rewards for keeping a consistent sleep schedule.

1.1 Motivation

Sleep is crucial for people’s health, despite this “Two-thirds of adults throughout all developed nations fail to develop the recommended eight hours of nightly sleep” (Walker, 2017). Sleeping protects newly acquired memories, refreshes the ability to learn and works to remove unneeded memories. Insufficient sleep can increase the chances of developing Alzheimer’s, lead to weight gain and increase the chances of the coronary arteries being blocked leading to cardiovascular disease, strokes, and congestive heart failure.

A key group that commonly suffers from sleeping problems are those with depression. 75% of depressed people have insomnia symptoms and these cause a “huge distress, have a major impact on quality of life, and are a strong risk

factor for suicide" (Nutt, 2008). Other studies echo these findings reporting sleep disturbances as the most prominent feature of depression and the principal complaint of depressed patients (Fang, 2019). Both studies indicate that insomnia can cause and cause reoccurrence of depression, a study on 7954 adults found that 14% of those with insomnia in an initial interview had developed depression by the second (Nutt, 2008). Insomnia is the most common residual symptom of depression and can cause customary treatments for the illness, such as Anti-depressants and Cognitive Behavioural Therapy (CBT) (NHS, 2019), to become less effective. Those with sleep disorders have a poorer response to CBT (Nutt, 2008) and some anti-depressants can exacerbate sleep issues (Fang, 2019).

Those with depression are also affected disproportionally by sleep inertia (see glossary). In a study of 600 people with unipolar depression, 74% reported difficulty getting out of bed (Trotti, 2017). An explanation for this is that people with mood disorders commonly report a reduced total sleep time, one of the strongest correlations with sleep inertia (Kanady, 2015). Furthermore, depressed people often express a wish not to be awakened and dread starting the day.

Sleeping pills are not a recommended treatment for insomnia (NHS, 2019), they sedate the brain similarly to alcohol causing the user to pass out, this leads to daytime forgetfulness and slowed reaction times. People who use sleeping pills are 4.6 times more likely to die over a 2.5-year period compared to non-users. Cognitive Behavioural Therapy for Insomnia (CBT-I) is recommended instead (Walker, 2017) (Fang, 2019) however, this solution does not usually include an element to address sleep inertia (Kaplan, 2018). This project aims to explore enforcing CBT-I habits through gamification while also creating a proactive solution to waking up that addresses sleep inertia.

1.2 Related Works

A literature review conducted on gamification explored the research of 24 empirical studies, the majority indicated positive results from the technique. However, there are caveats, user qualities can significantly affect their attitudes towards gamification and playing needs to be continuous, not sporadic (Hamari, 2014).

The app SuperBetter by Jane McGonigal (Figure 1) is described by her as the following: "use the psychology of gameplay in all of life to improve resilience, mental health and social-emotional learning". Users create, quests to complete, power-ups to activate and bad guys to vanquish, clearing these goals rewards the player with points and a pop-up congratulating them. Research indicates that

those who used SuperBetter for thirty days experienced significantly reduced symptoms of depression and anxiety along with increased optimism and social support (McGonigal, 2017).

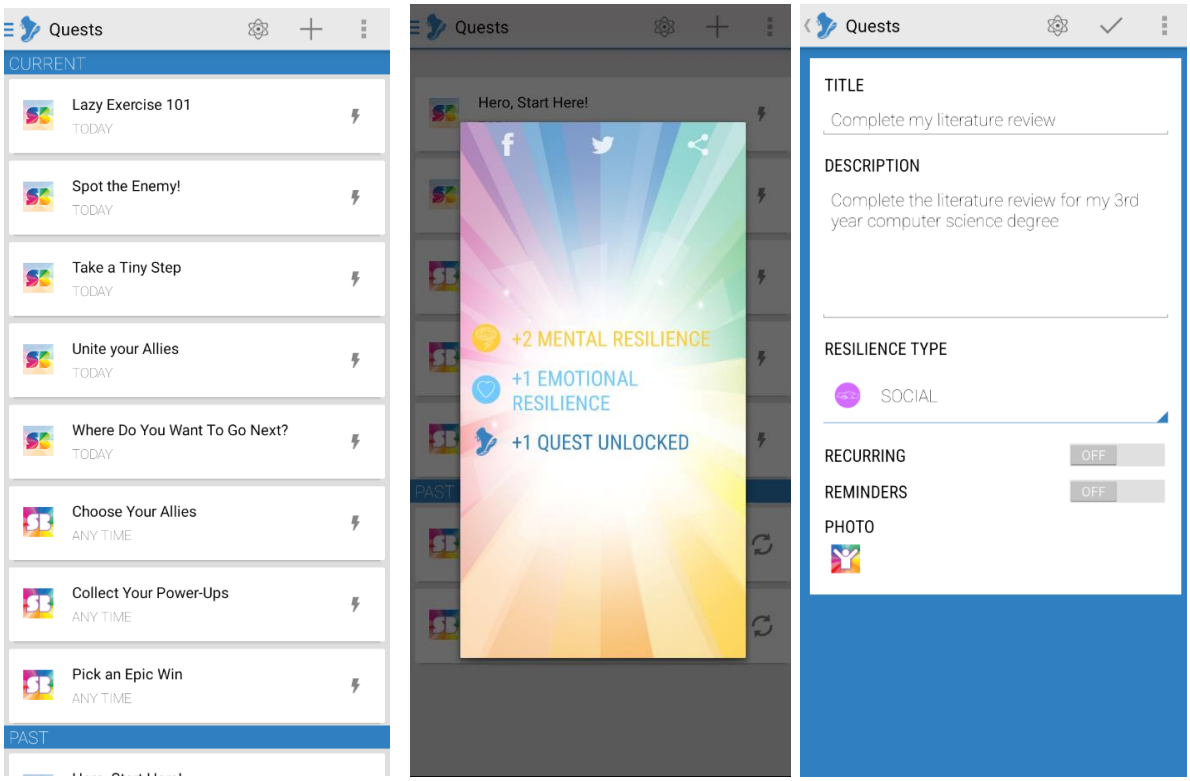


Figure 1 - Screenshots taken by the author in the SuperBetter app by SuperBetter, LLC. The leftmost picture shows the quest list, completing a quest leads to the next screen. The final picture shows the user creating a new quest called "Complete literature review".

Matthew Walker, a professor of neuroscience focusing on sleep, encourages the use of technology in sleep assistance stating "To battle against, rather than unite, with technology is the wrong approach in my mind" (Walker, 2017). Looking at the top 50 health apps on the Appstore 64% used gamification however, none of these were used to track sleep making this a lightly explored area (Cotton V, 2018).

Two gamified sleep assistance apps were studied, the first being SleepTown by Seekrtech (Figure 2), designed to encourage a regular sleep schedule using a city-building game. Players have a virtual city and construct buildings by leaving their phone off at night, using their phone during this period or waking up too late will cause the building to be destroyed. Users are encouraged to wake up and sleep at consistent times by adding a reward to this action (You, 2020).

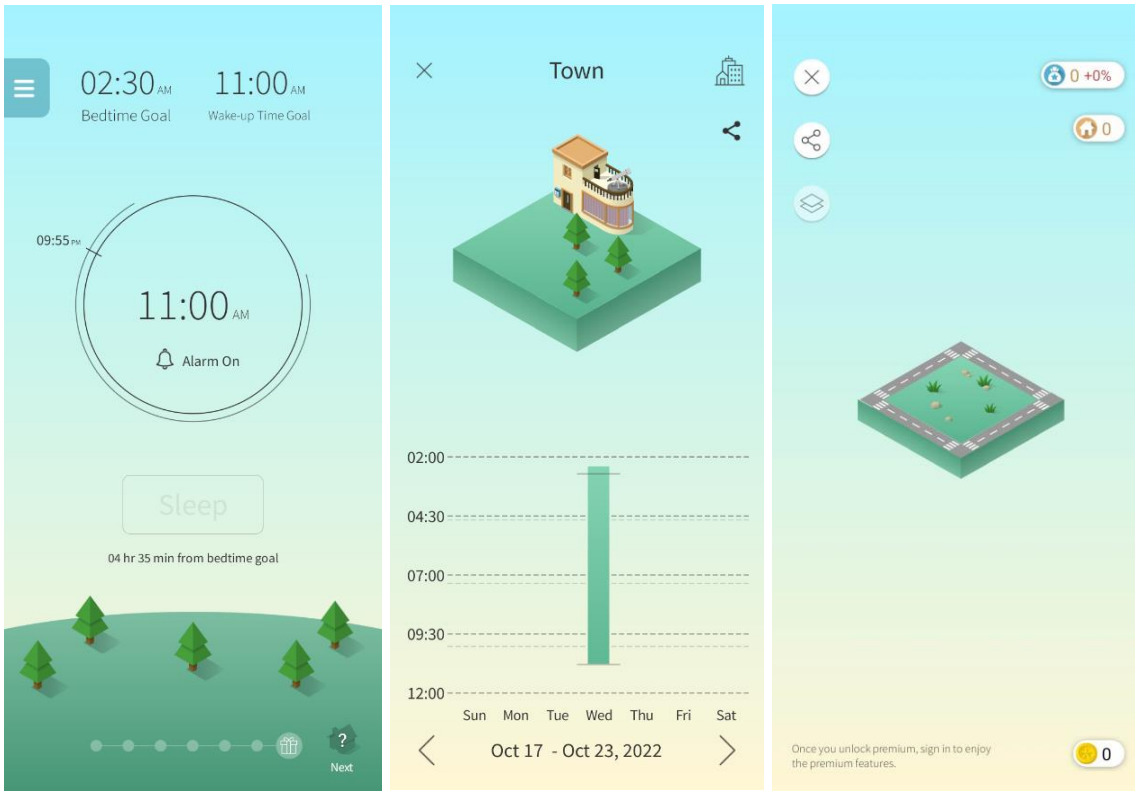


Figure 2 - Screenshots taken by the author in SleepTown by Seekrtech. The leftmost image shows the user's sleep schedule, they would need to start constructing their building before 2:30 am and wake up before 11:00 am to stop it from being destroyed. The next picture shows the records the app keeps. The final picture shows the tab "Big town" a gratifying way to view all constructed buildings.

The second app was Alarmy by Alarm Clock Alarmy (Figure 3), an alarm clock that rings consistently, requiring user engagement to turn it off through maths problems, memory problems, squatting and more. A study by the creators of Alarmy indicated large differences in snooze button usage depending on the morning task with the Maths challenge having the lowest. (Oh, 2020).

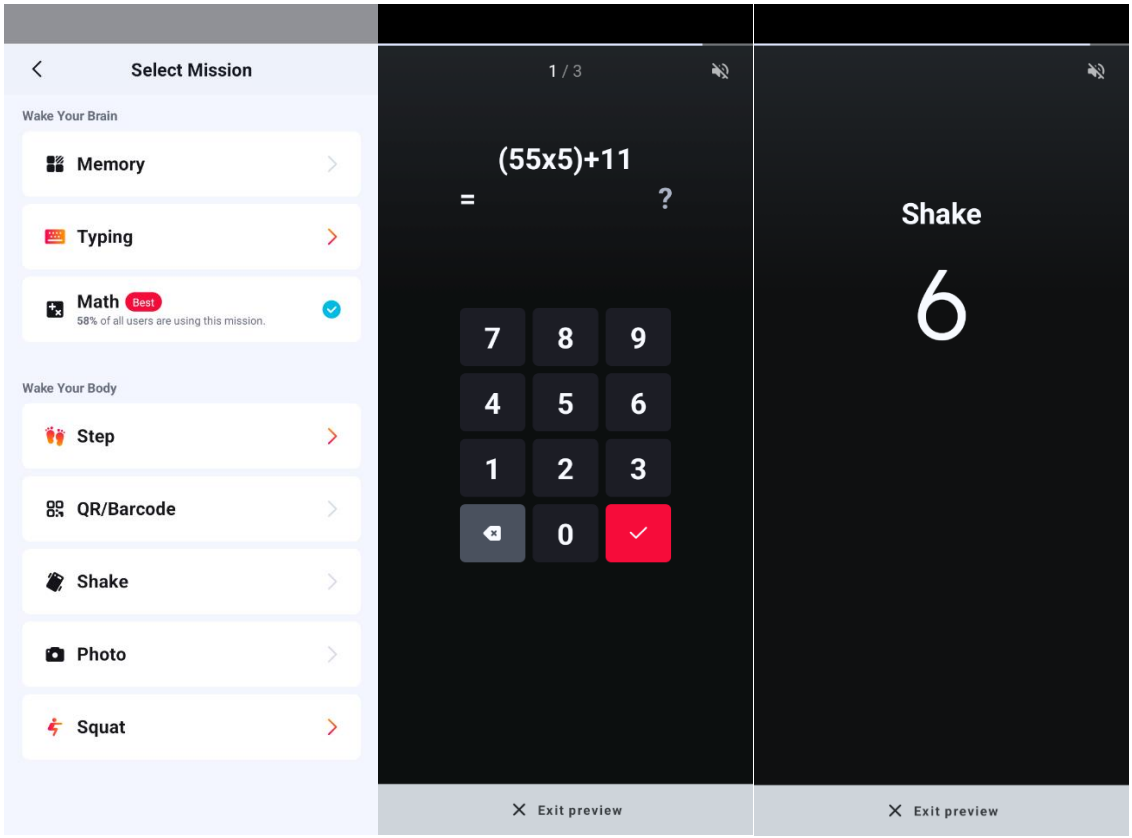


Figure 3 - Screenshots taken by the author in the Alarmy app by Alarm Clock Alarmy. The leftmost picture shows the different activities. The next two pictures show examples of a maths challenge and a shake challenge, the alarm will stop ringing when these tasks are completed.

An application created specifically for a study on gamified alarm clocks called ‘Sleepy Bird’ (Figure 4) was also analysed. Similar to the game ‘Flappy Bird’ by Dong Nguyen, an arcade-style game challenging players to avoid obstacles, Sleepy Bird added the feature that players’ scores would be affected by waking or sleeping late. When compared to a traditional alarm, participants were more likely to avoid the snooze button, go to sleep earlier and sleep closer to an optimum amount (7 hours) when using the gamified app (Ilhan, 2016).

Figure 4 - Screenshots of the app Sleepy Bird by Ezgi Ilhan, Bahar Sener and Hüseyin Hacihabiboğlu.

1.3 Project Aims

The solution aimed to build on these existing apps but target people with depression by addressing flaws in their design. SleepTown is punishing, a user opening their phone once during the night causes total failure which is disheartening to users, a proportional loss would be more appropriate for this project. Alarmy does not offer progression discouraging persistent play and using alarms to wake up can be stressful with Walker comparing them to "repeatedly assaulting your cardiovascular system (Heart and blood vessels)" (Walker, 2019). This project should aim to create an experience that users enjoy waking up to.

The proposed solution was an app that lets users play a fishing game for a short duration after a selected wake-up time, caught fish are placed inside a virtual aquarium where they can be grown and displayed. The initial focus was solely on waking users, but research showed that difficulty waking up normally came from poor sleep therefore, a mechanic where fish became progressively 'unhappy' while the phone was active overnight was introduced. A statistics page will show when a user's phone was used overnight allowing them to analyse their sleeping patterns. The setting of fishing was chosen as it was believed to be a relaxing environment that those with mood disorders would not dread waking up to.

1.4 Document Structure

The next section specifies requirements for a compelling game and how to implement these as features, a series of design documents are then presented. The functioning of every piece in the final design is then described along with a review of the final product. Details of an experiment testing the application on a set of participants to see if it helped their sleep will be discussed. Finally, the successes and failures of the project are examined with input on future possibilities.

Design and Implementation

Before implementation could begin, research was required on creating a compelling game along with how to develop phone applications.

2.1 User Requirements

People enjoy games for different reasons and it is making it difficult to appeal to everyone. The six user types hexad (Figure 5) by Andrzej Marczewski (Marczewski, 2015) groups people based on what they desire from games and was used to define possible features that would engage them (Figure 6). The Hexad was chosen over other player typing methods as it was designed specifically for gamification.

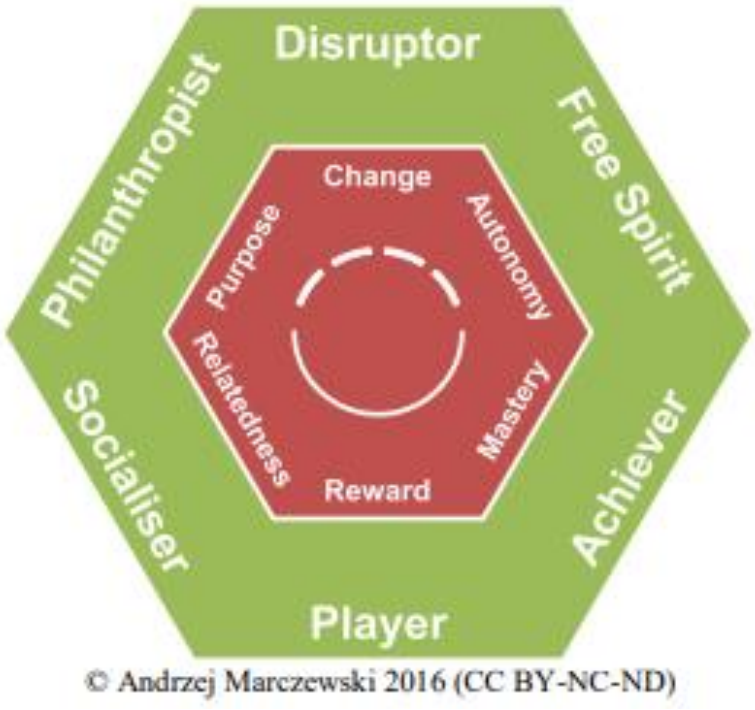


Figure 5 - The Gamification User Types Hexad by Andrzej Marczewski 2016

User Type	Desire	Possible Feature
Socialisers	Interaction with others	Friends list, online features
Philanthropists	Giving without reward	Gift fish to other players
Free Spirits	Freedom to express themselves	Fish can be customised
Achievers	Difficult challenges to overcome	Create rare fish that are harder to catch than others
Player	Beneficial rewards	Award money that can be spent in the game
Disruptor	Push the boundaries	Leave bugs and other exploits in the game

Figure 6 – This table shows the six user types from the User Types Hexad, the motivation they desire from games and a possible feature that could be included in the final product to appeal to them.

Disruptors are difficult to cater to as they desire to break the game itself, however, they only account for 1% of people so features appealing to them were not included in the specification.

The user types showed the essential features needed to create an engaging gameplay loop, but it was difficult to imagine the system in full. To give a better idea of the final product, two scenarios were written envisioning how the app would work for a student with depression and an older person working an office job. Both contain descriptions of features that would not be implemented but provided insight into how the app would work.

See Appendix for Scenarios

Using the scenarios, a specification was created showing what would be required for every part of the app. The MoSCoW system was used, assigning each requirement a priority (In descending order; Must, Should, Could and Would) allowing essential elements to be focused on. All online elements were placed in the ‘Should’ category as they were a priority but adding them would be difficult and creating the base game was more important. Only elements in the ‘Must’ category would be included in the final build, so this prioritization proved effective.

See appendix for MoSCoW requirements

2.2 Design Decisions

To save time, the development would focus on one OS (Operating System), Android which was selected because; it is the most common phone OS (StatCounter, 2022), my personal phone runs Android, and publishing to the Google play store is inexpensive. Time was set aside before development began to learn Android Studio, the official IDE (integrated development environment) for Android systems. Despite Kotlin being Google's preferred language, Java was chosen instead due to personal experience using Java and many online tutorials still employing it. The tutorial "Build Your First Android App in Java" from Google Codelabs was followed to learn the basic features of Android development (Java Code Lab tutorial link). Progress moved to prototype complex features like setting an alarm through an app, sending notifications, detecting when a user's phone is turned on and running code while an app is closed (How these were implemented is covered in implementation). Preliminary prototyping showed what was possible in Android before development began insuring it was the correct OS. The following sketches were made before this stage with an unclear idea of Android development (Figure 7), the middle image shows a custom alarm screen which was found to be difficult to implement and was therefore scrapped.

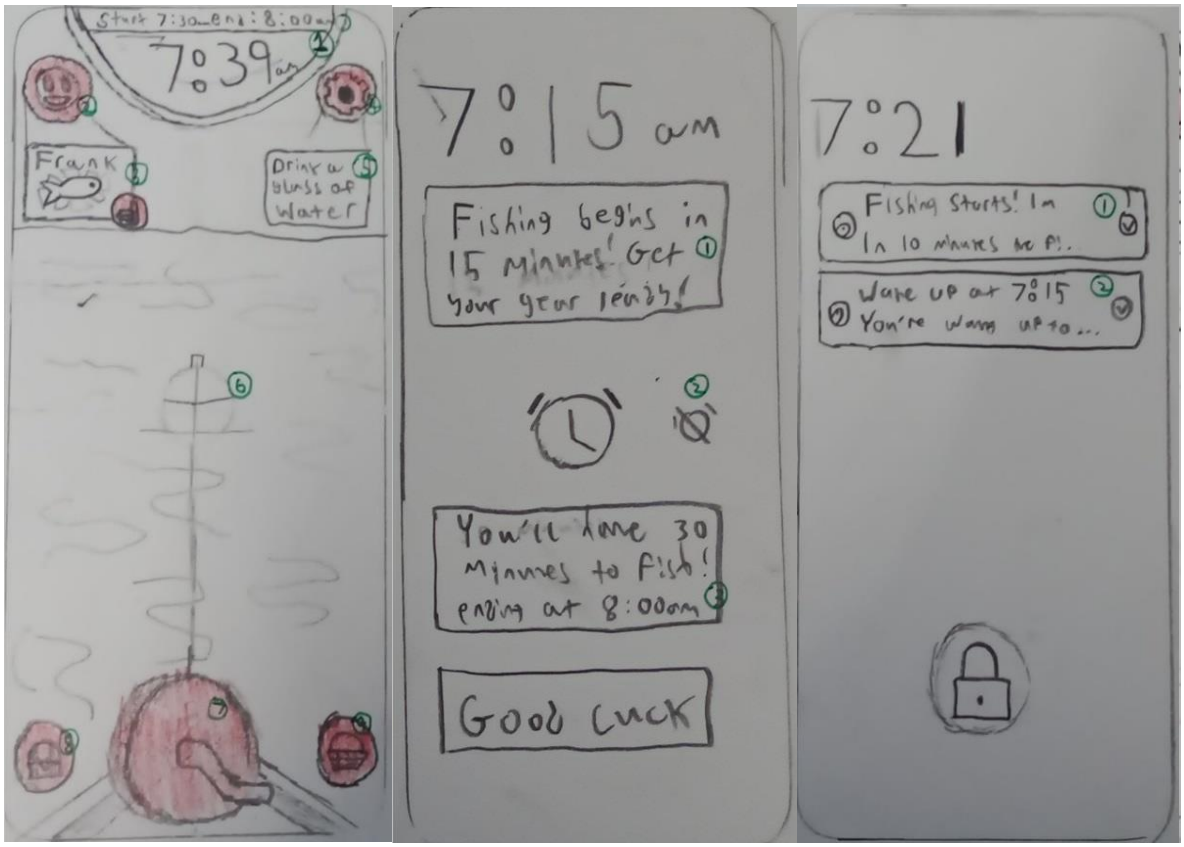


Figure 7 - Initial sketches of the app showing from left to right; the fishing game, an alarm screen and notifications.

With experience gained through prototyping, new sketches were created. An Android app works by having a series of activities which hold building blocks for UI (User interface) components (Developers, 2019), the goal for these sketches was to envision what the main screens would look like (Figure 8). Modularity was employed, elements such as the top and bottom bar would be identical across all screens meaning a user only has to learn the layout once.

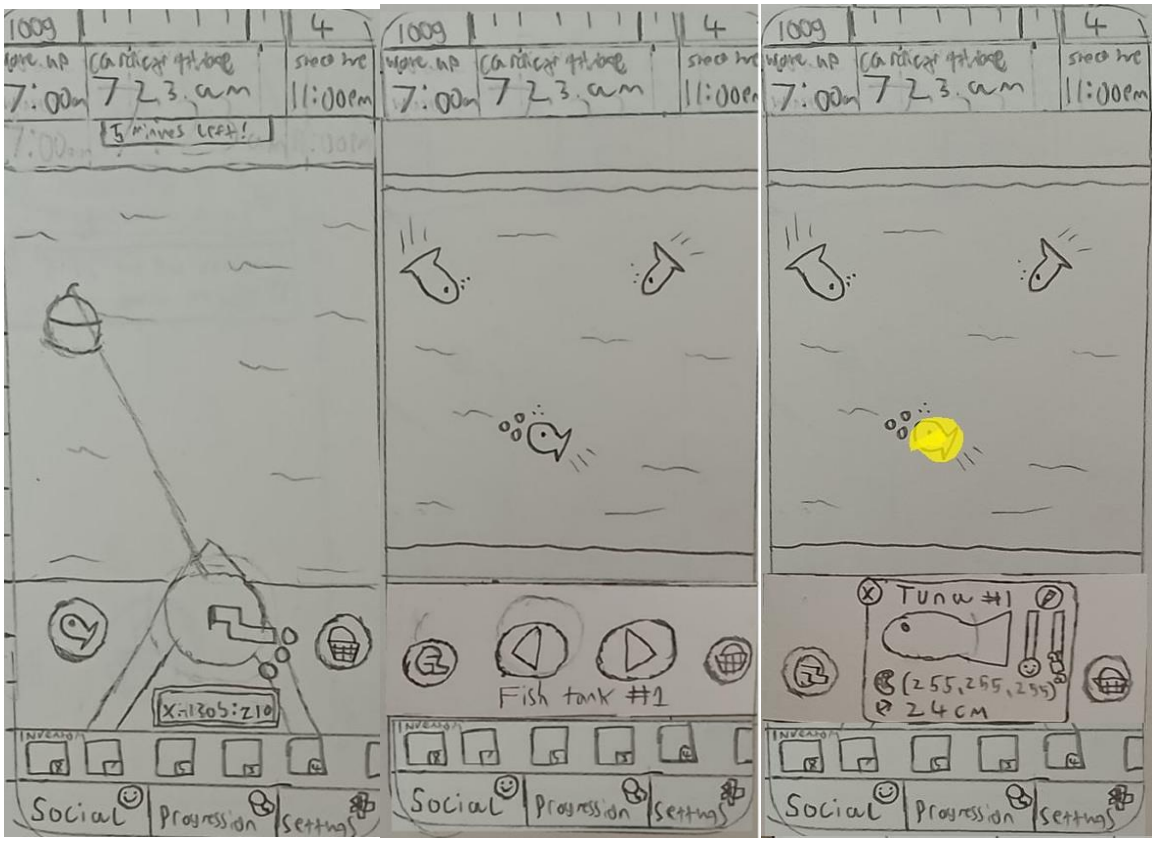


Figure 8 - Sketches of the app showing from left to right; the main fishing game in the morning, the aquarium without a fish selected and with a fish selected.

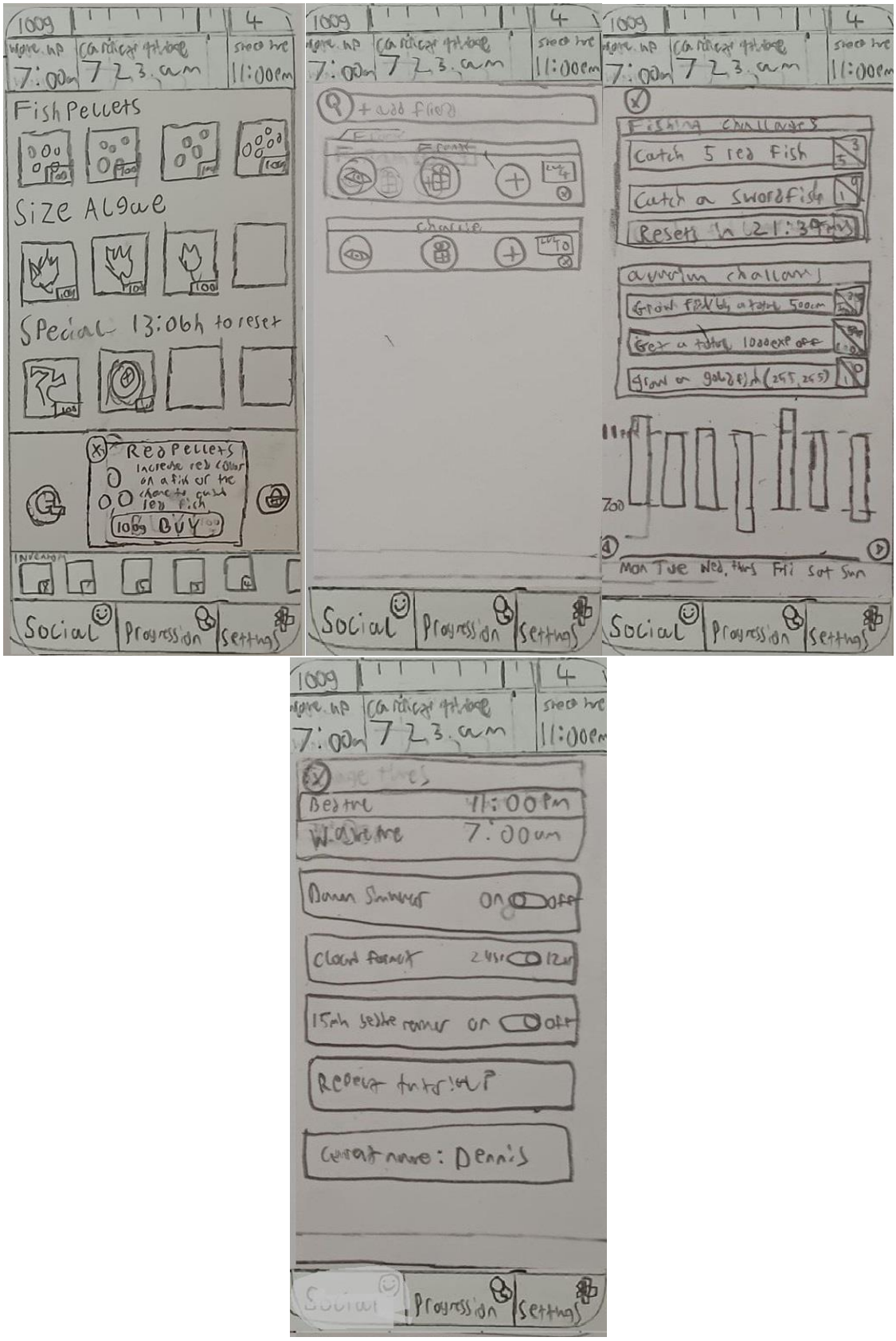


Figure 9 - Final sketches of the app showing from left to right, top to bottom; The shop screen, the friends list, the progression tab and the settings tab.

A set of storyboards were then created to explore how a player would interact with the game specifying the controls needed to perform actions. These would give a clear outline of what needed to be implemented in the finished product to meet the minimum requirements of being an interactive game. (Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15).

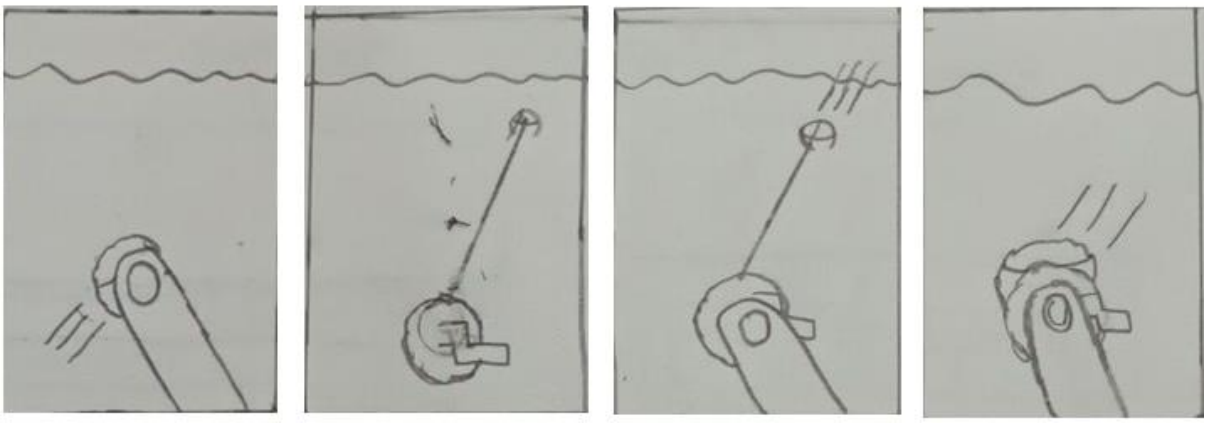


Figure 10 – The fish float is cast by swiping it and the fishing reel can be held down to progressively pull the float towards it.

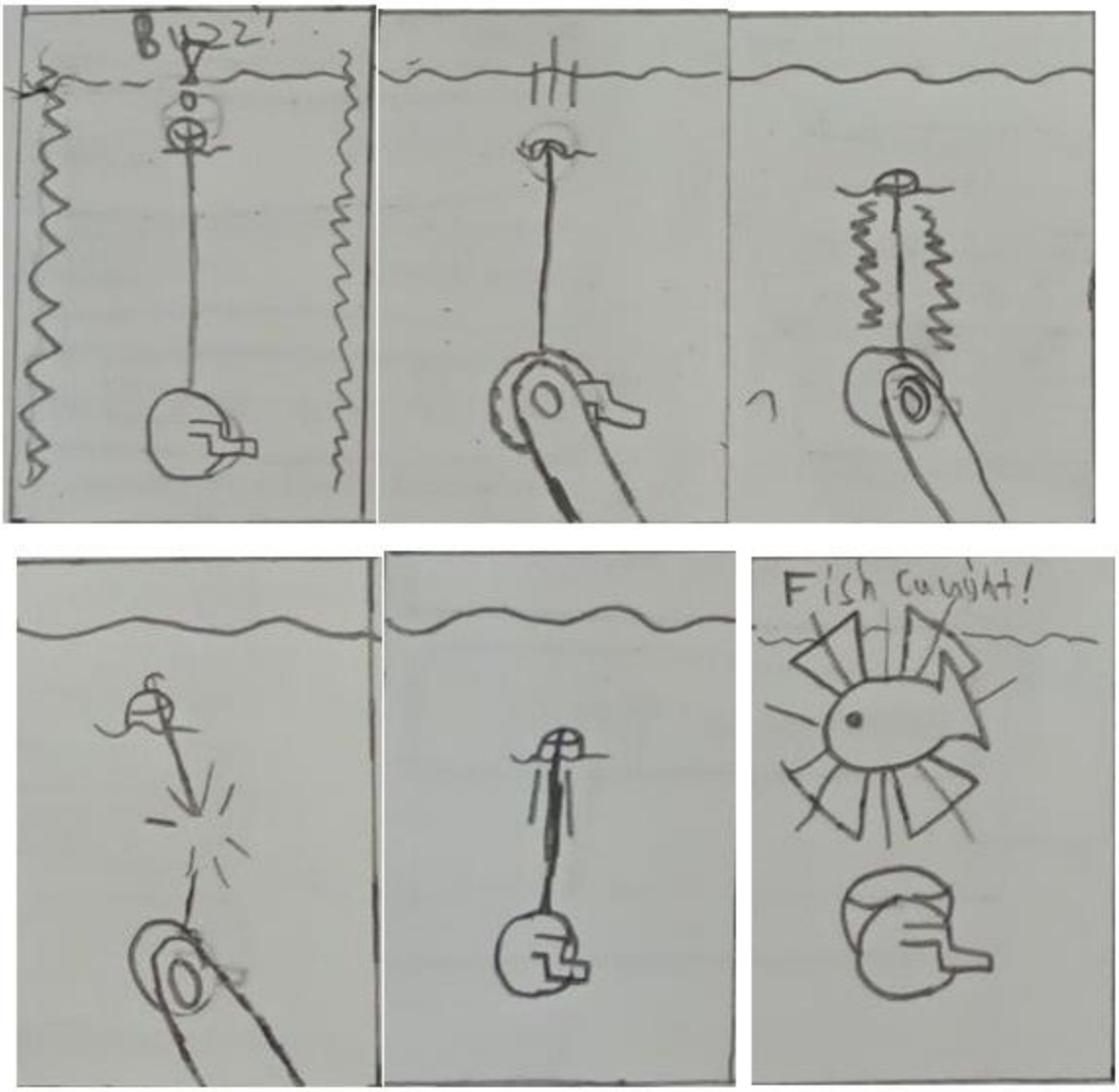


Figure 11 – A fish bites the reel creating an exclamation mark image and giving haptic feedback. Holding down the reel for too long will cause the line to snap, and not holding it will cause the fish to move away. When the float is returned to the reel a display shows the caught fish.

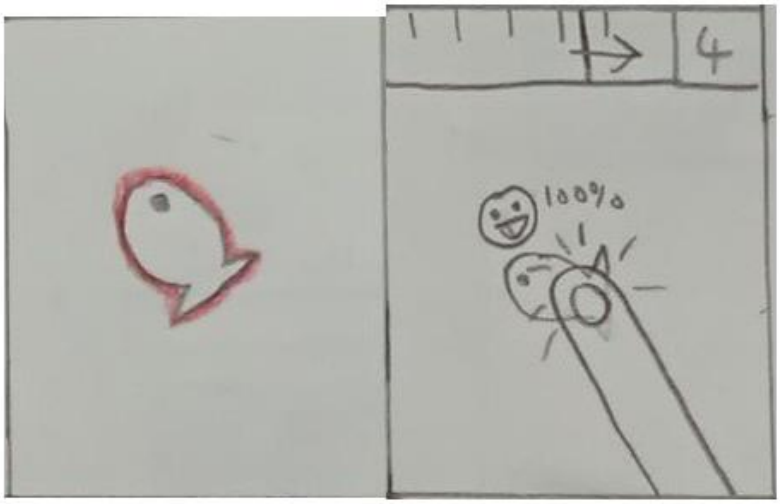


Figure 12 – Tapping a glowing fish gives experience based on how happy the fish is.

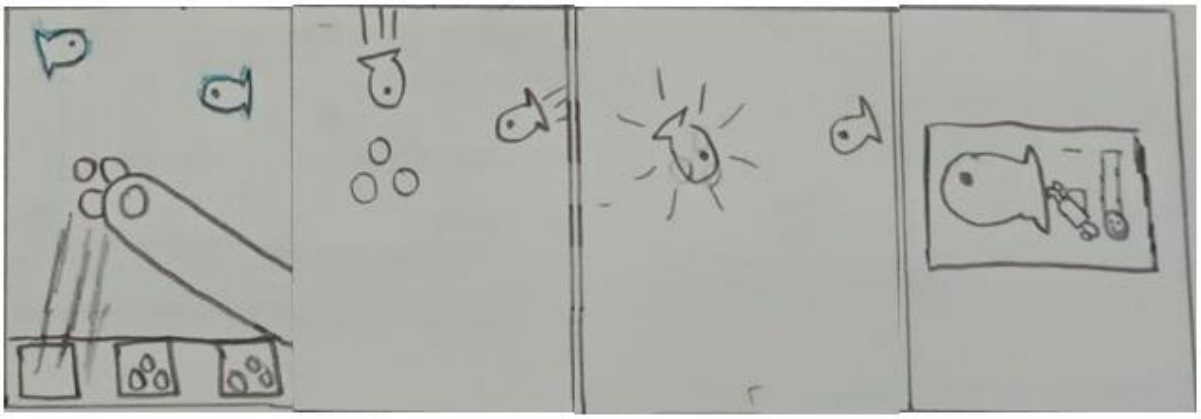


Figure 13 – Food can be dragged from the inventory into the aquarium and the first fish to eat it gains its effect. Fish can be tapped to see their stats such as their happiness and if they're hungry.

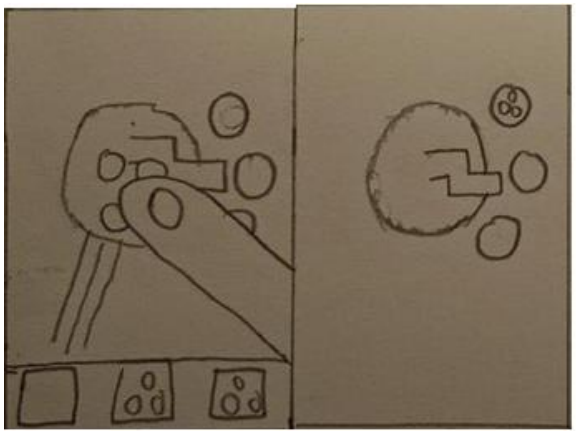


Figure 14 – Food can be dragged onto the reel to add effects to caught fish.

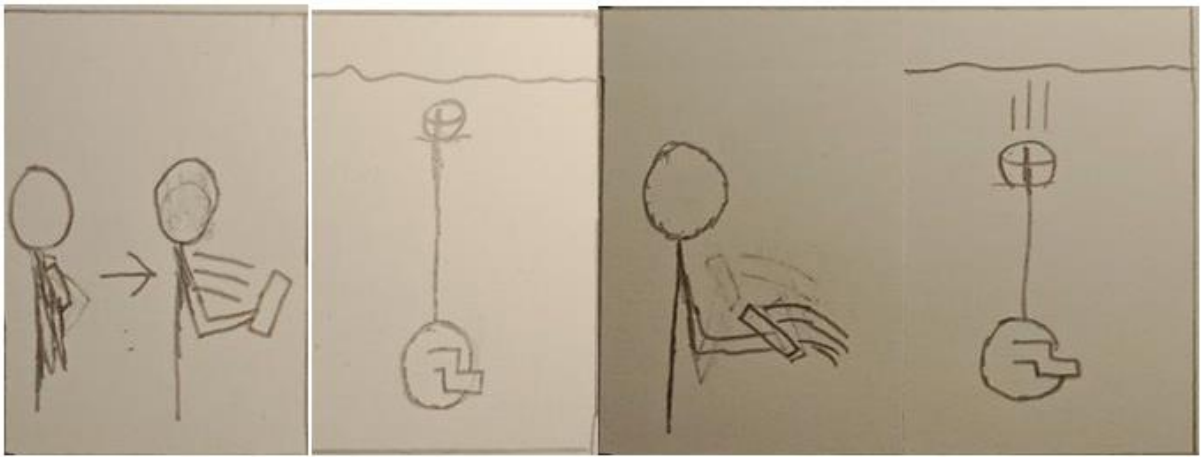


Figure 15 – Using the fishing game with motion controls; Throwing the phone forward casts out the float and tilting it backwards reels the float in.

Gameplay interaction was inspired by mechanics in existing video games that had already shown success. Throwing the fishing float is similar to throwing a Pokéball in Pokémon Go (Go, 2016) while the aquarium is very similar to phone games like The Simpson's Tapped Out (Out, 2012) requiring users to tap collectables every day to gain rewards.

Figure 16 – Pokémon Go by Niantic, screenshot taken by the author. The bottom ball is thrown at the creature in the middle by swiping.

Figure 17 - The Simpsons: Tapped Out by EA Mobile, screenshot taken by the author, notice the floating signs above objects, these release experience points when they are tapped.

Two UML diagrams were created to model the program. The class diagram (Figure 18) was an effective blueprint making implementation far easier although some classes would not be included in the final build. The activity diagram (Figure 19) proved less useful, it displayed functionality reserved for online features which would not be implemented due to time constraints.

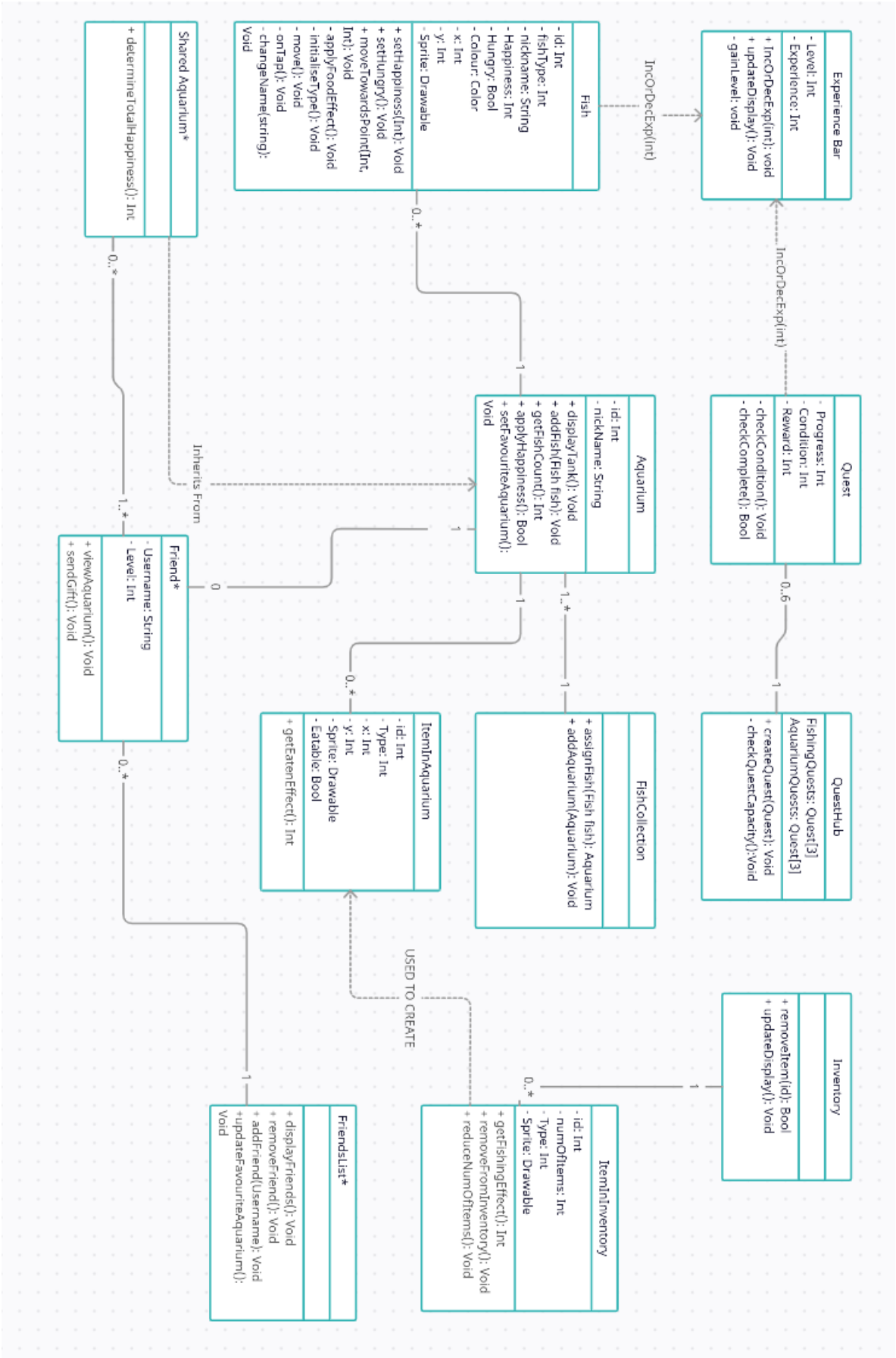


Figure 18 – Class diagram; classes with a '*' symbol in their name are online features.

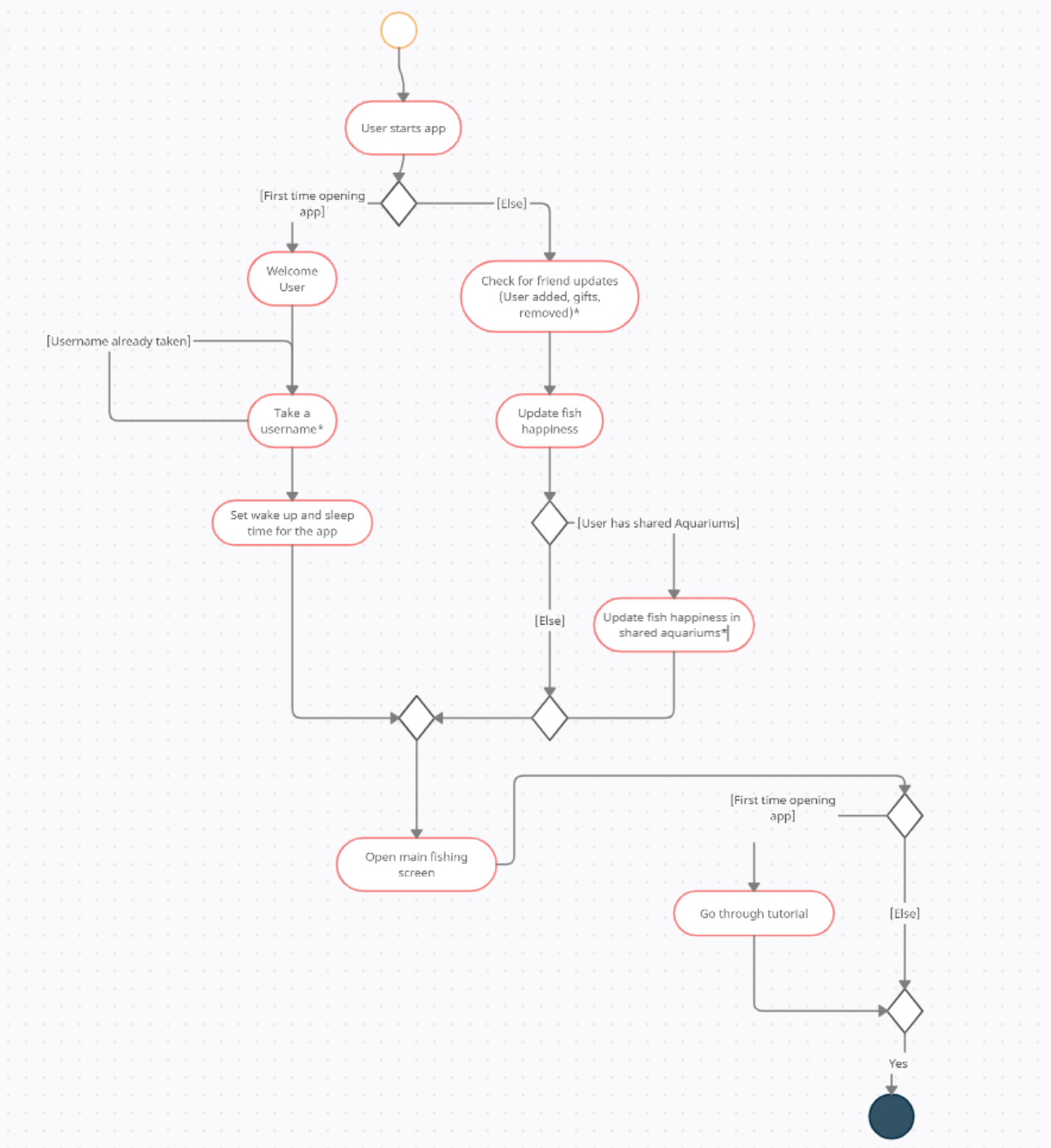


Figure 19 – Activity diagram; shows the processes performed when the app launches.

2.3 Implementation

2.3.1 Fishing Game

Android apps are split into activities which hold View classes, these occupy a rectangular space and are used to create interactive UI components (Developers, 2019), such as GUI elements, visual indicators for gameplay etc. Views can

respond to user input by assigning an onTouchListener class which performs a function when tapped, this worked for simple features like buttons but casting the fishing float required further work. Android registers MotionEvents which track the actions and coordinates of user interaction, complex input can then be interpreted through a GestureDetector. One was created with an overridden onFling method, if the system detects a swipe and it started within the fishing reel's boundary, then the float is cast. The position it moves to is calculated using an invisible View placed above the fishing reel, the distance the user swiped inside the box is scaled to calculate a position in a larger water View. The below diagram (Figure 20) shows an example.

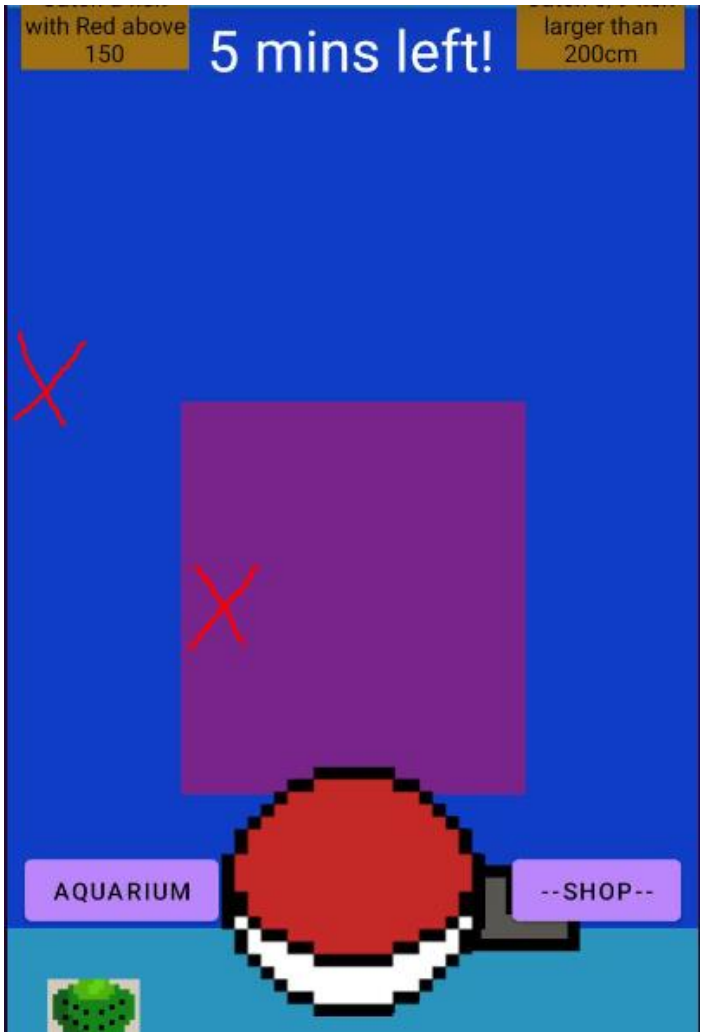


Figure 20 – A fling finishing at the first X will cause the fish float to move to the relative position on the water view at the second X. Note the red box is invisible in the final build.

Moving Views was accomplished using the Runnable interface, a method to implement threading. Runnables were created for actions such as casting the float, bringing the float back, checking for fish, fish pulling the float away and animations to display caught fish. These would be posted and removed at

different times, for example, touching the reel starts the bring float in Runnable and letting go removes it.

A class was created to manage the drawing of a fishing line between the reel and the float using Java Paint, this calls an onDraw method whenever it changed position to update the line's rendering. The line also shakes and turns progressively redder as the user holds down the reel to indicate when it will snap.

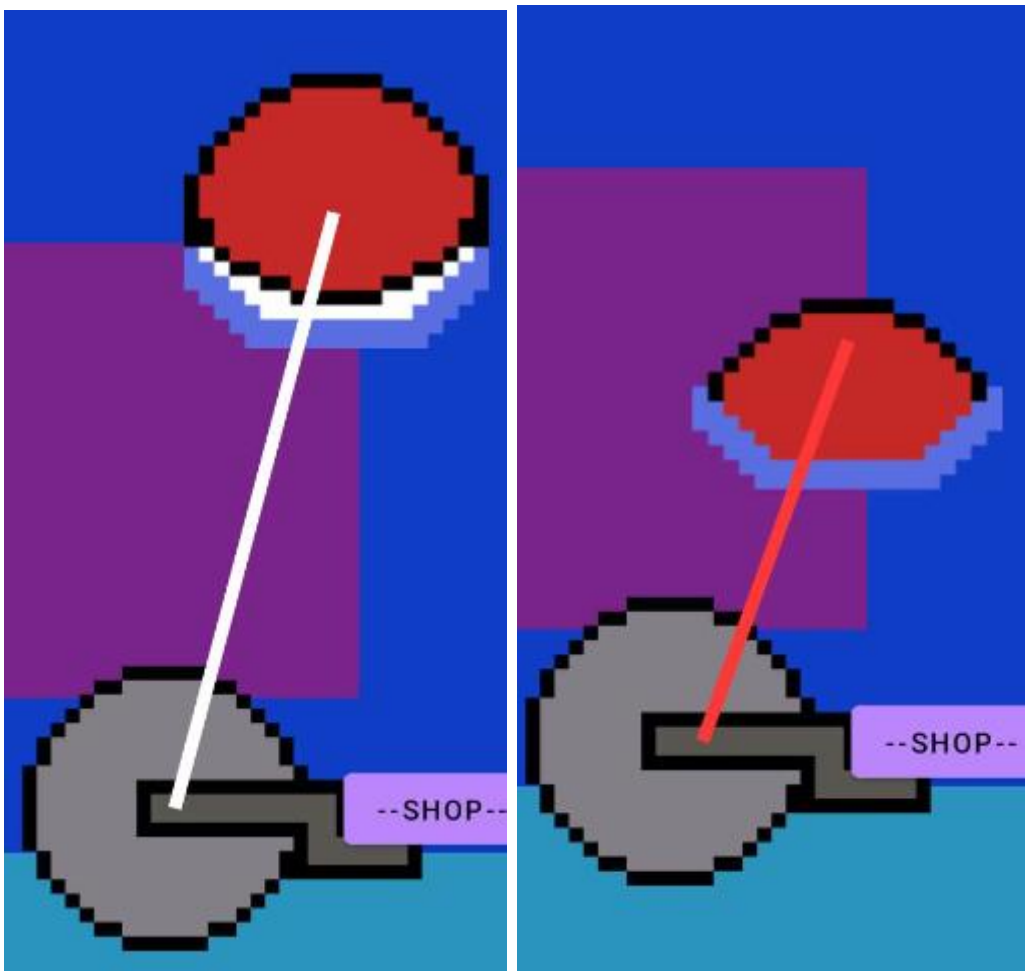


Figure 21 - The fishing line; the left image shows it normally, the right when the line is close to snapping

Quests were implemented as it was believed they would be an important incentive to keep players engaged in the game, they are updated and stored in the Global class (discussed further). The updateQuests' method is provided with a fish, if this meets the requirements of one of the quests (such as being above a certain size) its progress is incremented. The aquarium and fishing game each have unique conditions for quests. The quest's values are chosen randomly over a range to add variety, every possible combination is displayed below (Figure 22) along with examples (Figure 23).

Fishing Quests	Aquarium Quests
Catch 3/6/9 fish larger than 200/175/150cm	Grow fish by 30/60/90cm
Catch 3/6/9 fish smaller than 150/125/100cm	Set a fish's colour to above (100/150/200, 100/150/200, 100/150/200)
Catch 2/4/6 carp	Obtain 20/40/60 fish
Catch a fish with red/green/blue above 150/200/250	Receive max happiness from 10/20/30 fish

Figure 22 - Every quest that can be generated. The slashes indicate where values differ.

Figure 23 – Examples of quests; the left ones are fishing quests and the right are aquarium quests.

2.3.2 Aquarium

The fish object is represented by a View with the size, colour and sprite being defined by its internal properties. Movement is accomplished through two persistent Runnables; assignFishMove, which triggers sporadically calculating a random location for a fish to move to, and moveFish which adds the fish's speed to its coordinates and redraws it. Plans were for fish to move omnidirectionally however, this led to values being rounded which conflicted with the need for precise coordinate tracking to stop them from moving off-screen. The final build sees fish moving in four diagonal directions.

Along the bottom of the screen is the user's inventory holding food items to be attached to the fishing reel or fed to fish in the aquarium. Holding down on a box causes the sprite of the associated item to begin being drawn wherever the user is pressing. An onDragListener can be attached to Views to create drop zones that trigger functions when a user's drag finishes on top of it. The aquarium's water View has one attached which creates a representation of the dragged item in the dropped location when triggered (Figure 25). It also assigns the movement position of the closest fish to the item's location.

Figure 24 - The user's inventory bar; in this example, it holds 2 red food, 1 green food and 1 blue food

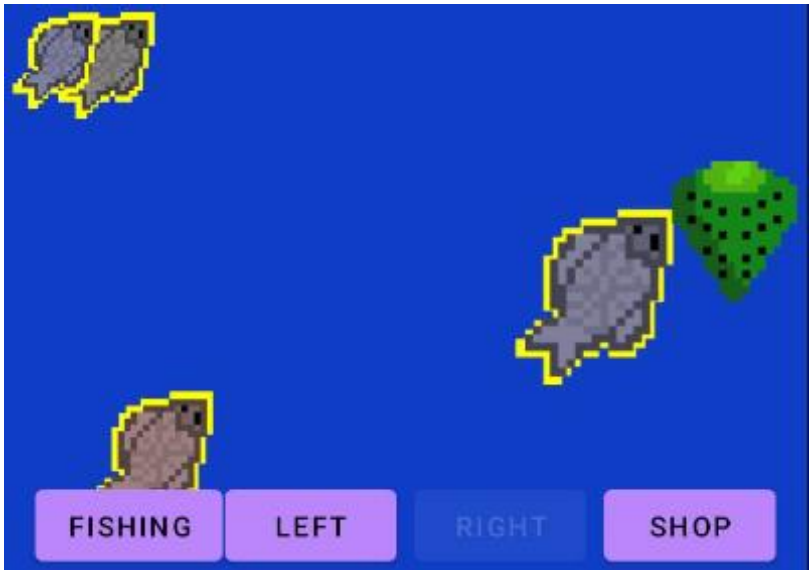


Figure 25 - A Fish moving towards 'Green food'

There are three food types, each increasing a different RGB colour value (Figure 26). Plans included a ‘size food’ to increase or decrease a fish’s size, but this was combined into the colour food to simplify development. Currently, internal size values are updated but the method to redraw the sprite with a new bitmap visualising changes does not work.

Figure 26 - The three types of food

2.3.3 The Global Class

Although not included in the class diagram, a way to manage global variables was necessary, this was implemented using the Application class which filled this role well, being omnipresent and created before other classes. Container classes for aquariums and quests from the class diagram worked better when wrapped into this Global class. It also holds functions maintaining these variables such as setting sleep timings, generating quests, managing inventory and tracking a user's sleep.

Global needed to function as though it was non-volatile as it held many key values that should not be reset when the app is closed such as fish, inventory items and records. A class called SaveObject was created to hold these values which Global holds an instance of, it is updated through two functions in the Common class (Discussed further). saveState updates the values of global's SaveObject, converts it to a JSON file and saves it in sharedPreferences, the recommended location to store key data (Developers, n.d.). When the app is launched, loadState is called which retrieves the JSON file, converts it back to a SaveObject and uses this to set the values of Global.

saveState is called in all activities' onPause function which is run whenever the activity is visible but not being interacted with. This function is run excessively but preserving user data is incredibly important, using onDestroy or the onStop function, which both run less frequently, was not possible as they are not called if Android closes the program to give higher priority apps more memory. This is displayed below with the Android activity life cycle (Figure 27) (Developers, 2019).

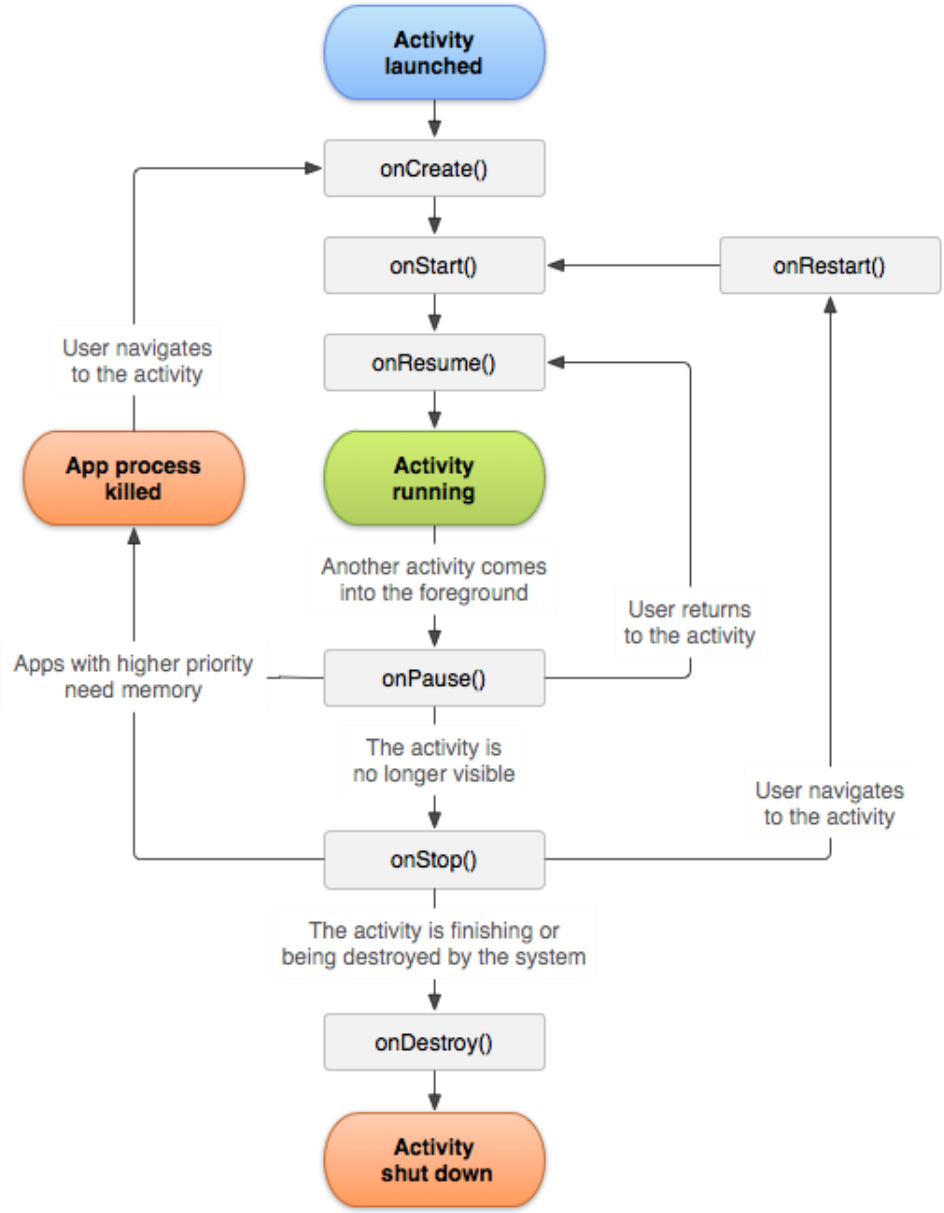


Figure 27 – The activity lifecycle; the rectangles show functions that can be implemented in an activity class and are called in the order indicated by the arrows.

2.3.4 Sleep Tracking

Detecting when a user's phone is active was the hardest part of implementation. Android provides the Service class which can run code while an app is closed. It also provides BroadcastReceivers, objects that run code when a condition is met on the device (for example, aeroplane mode being toggled). The first idea was to have a constantly running Service holding a BroadcastReciever responding to when a user's screen was active and increasing a counter if it was past their sleep time. However, a BroadcastReciever running constantly while the screen is active is battery intensive which risks Android ceasing the Service.

A new solution was created that used two BroadcastRecievers tracking when the phone screen was turned on and off respectively. When turned off, the difference between the time turned on and the current time is calculated to find how long the phone was active. Although this led to complexity in trying to deduce if a phone had been on before or after the sleep time, it was far more efficient.

Development was heavily slowed by the mistake of storing time as a pair of integers representing the hour and minute. Although it used up little space, telling if a new day had begun was difficult along with timings being ignored if they were less than a minute. The change was made to tracking time using Java's Time and Calendar classes which should have been used from the start.

BroadcastRecievers constantly gathering the status of a user's power button posed ethical concerns as there was no need to track this other than past their sleep time. The interaction was changed so that users would need to press a button before going to sleep that started the Services. Although this would lead to problems such as users forgetting to press the button before sleeping, it was important to only monitor a user's phone when necessary. A screen was also added, displayed when the app is first launched, showing the required permissions along with explanations for why each is needed. This was crucial for those with mood disorders who may become anxious at an app that tracks their phone while they sleep, explaining it only tracks when the power button is pressed should help ease these feelings. In the case a user is still uncomfortable giving these permissions, the app is still usable without them with some functionality removed.

Pressing this sleep button also initiates a function to decrease the user's screen brightness to its minimal amount through editing the system settings. Melatonin production (the brain chemical that makes people feel sleepy) is drastically decreased by LED light (Walker, 2017). Dimming screen brightness can help alleviate these effects and also works as a visual indicator of turning the lights off in the aquarium.

The times the user's phone was turned on and off are saved as a Record object holding these values in a list along with the time spent awake and the date the night was recorded over. A visual representation of stored Records can be seen on the statistics page to show user's how their sleep has changed (Figure 28).

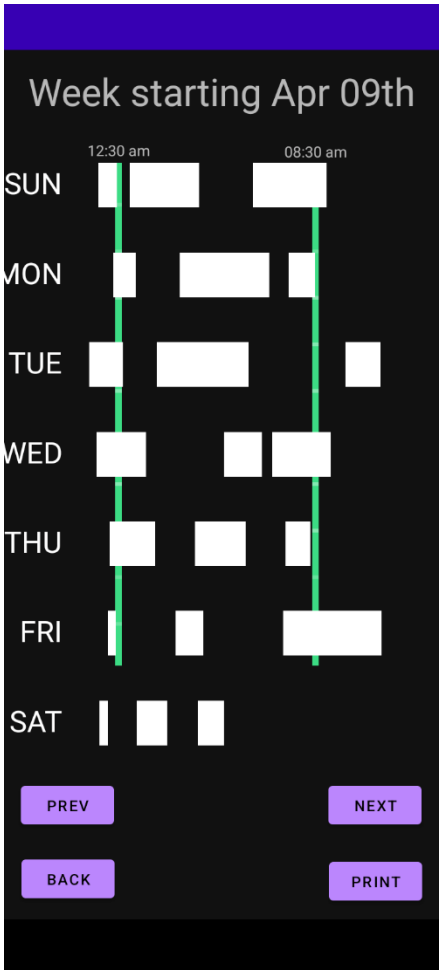


Figure 28 - An example statistics page representing multiple Record objects; The white bars show when this user's phone was off.

Push notifications sent 15 minutes before the user's sleep and wake time were also implemented using Services. They additionally use Intents, an object representing an abstract description of an operation to be performed, (Developers, 2019) which is triggered every twenty-four hours using an AlarmManager. The Intent activates a BroadcastReceiver which starts a Service to create and push the notification allowing it to be sent even if the app is closed.

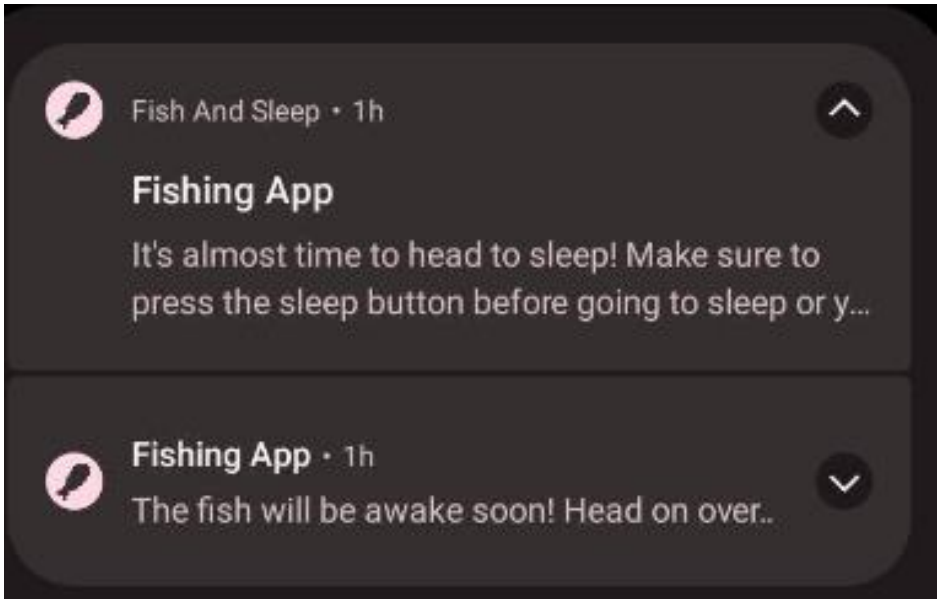


Figure 29 – The two push notifications; The top is sent before the user's sleep time and the bottom before the user's wake time.

2.3.5 GUI Elements

The Global class could provide functions across activities but concerns this would overburden it led to the creation of a static class called Common to fulfil this purpose instead. Some already discussed functions are contained within this class such as initiating sleep and saving data.

The Common class holds functions to visually update GUI objects, (user's money amount, the experience bar and quest boxes). This function is run whenever a change may have occurred for any of these values such as a player catching a fish or completing a quest. Although costly, it is important that the GUI always correctly represents internal values to a user.

Another Common function is the creation of a generic pop-up with a textbox and a button, these would be used extensively to display information. Pop-ups are themselves Activities and could serve as standalone screens but by using Android's DialogBuilder and LayoutInflater class, they can be displayed as sections of another Activity. Pop-ups were used extensively in the tutorial.

Users needed to understand how to use the app, so a large part of development was focused on creating an effective tutorial. When first launched, users must process through a series of slides explaining how the app operates, it then asks when a user would like to wake up and sleep and if they would like an alarm fifteen minutes beforehand. The already explained permissions screen is shown before moving to the fishing activity. Whenever an activity is opened for the first time a pop-up is displayed explaining its purpose. Pop-ups are also shown

when gameplay actions are required like tapping fish and casting the fishing rod, these also feature a hand sprite showcasing the action needed to give a visual guide to the user (Figure 30). After the first completion of the action, the guide will stop being shown.

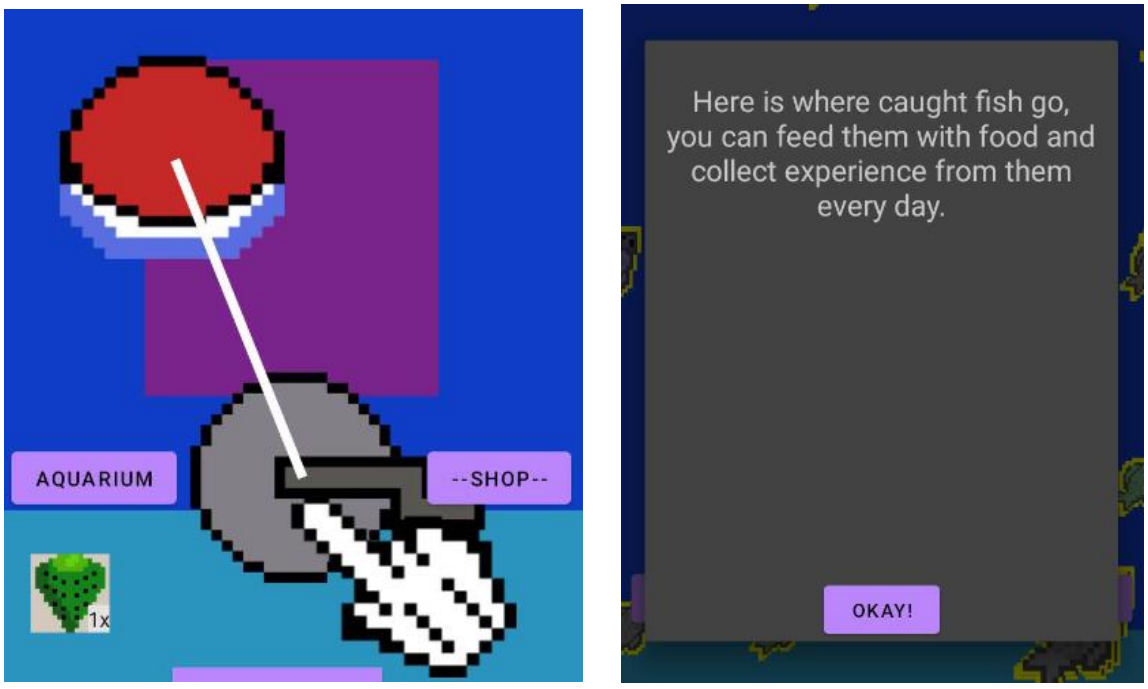


Figure 30 – The left image shows a hand sprite which appears the first time the user needs to reel in the fishing rod. The right shows a popup which appears when the user first opens the aquarium

2.4 Final Design

The final project was unable to include many features in the MoSCoW requirements. The ‘must’ statements were all accomplished excluding the size-changing food. Quests and the statistics page were the only ‘shoulds’ included in the final design, online features were scrapped in mid-January to allow time to fix problems with the app and perform the experiment (detailed next chapter). Motion controls were another feature not added as the focus had to be on completing the main game. No ‘could’ statements were implemented though this was always a possibility and did not cause drawbacks.

Referring back to the user types hexad, only the Free Spirit and Player types have features that appeal to them, using Marczewski’s study as a measure of distribution means 32% of people have been catered to (Marczewski, 2015). The creation of a stable build of the app was crucial despite it leading to online features being cut excluding the Philanthropist and Socialiser player types.

The layouts for both the fishing game (Figure 31) and aquarium (Figure 32) were similar to their final appearance however, the supplementary screens ended up looking fairly different.

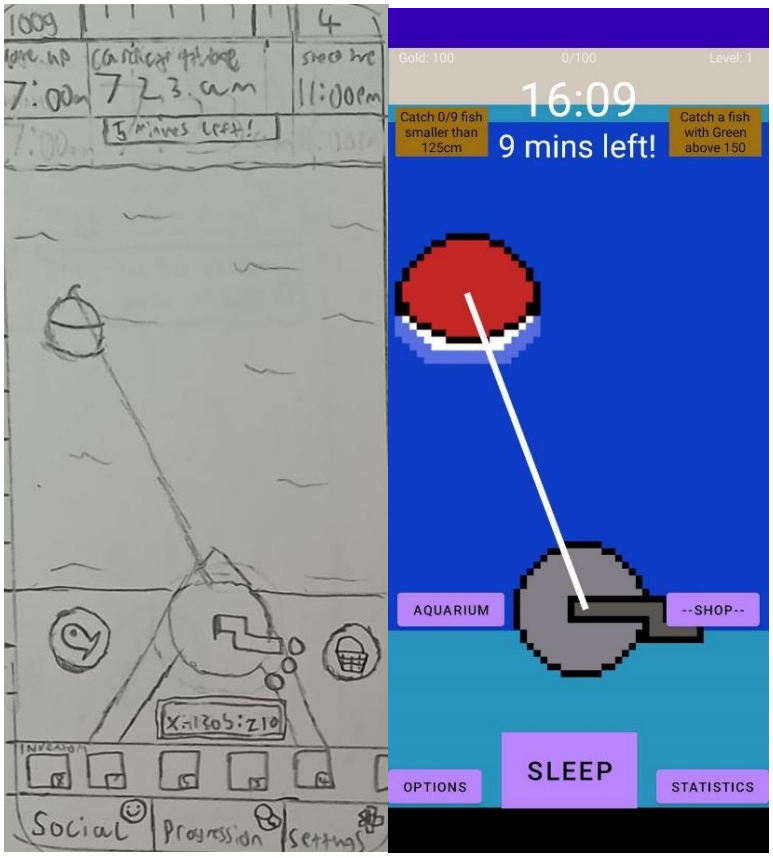


Figure 31 – Evolution of the fishing screen

Figure 32 – Evolution of the aquarium screen

Progression was renamed to the statistics screen (Figure 33), it originally held details of a user's quests, but these were moved to the fishing screen to give users immediate access to them. The whole tab now holds visual representations of the Record objects with the bars filling the page.

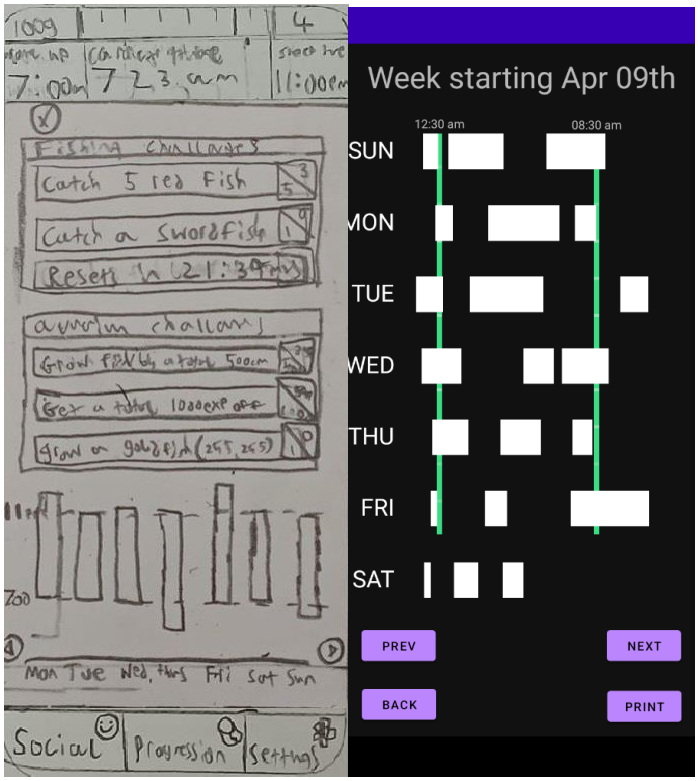


Figure 33 – Evolution of the progression screen

The shop tab (Figure 34) was largely the same except for the removal of the size algae section (combined into the colour food type) and the special section (Which was never implemented).

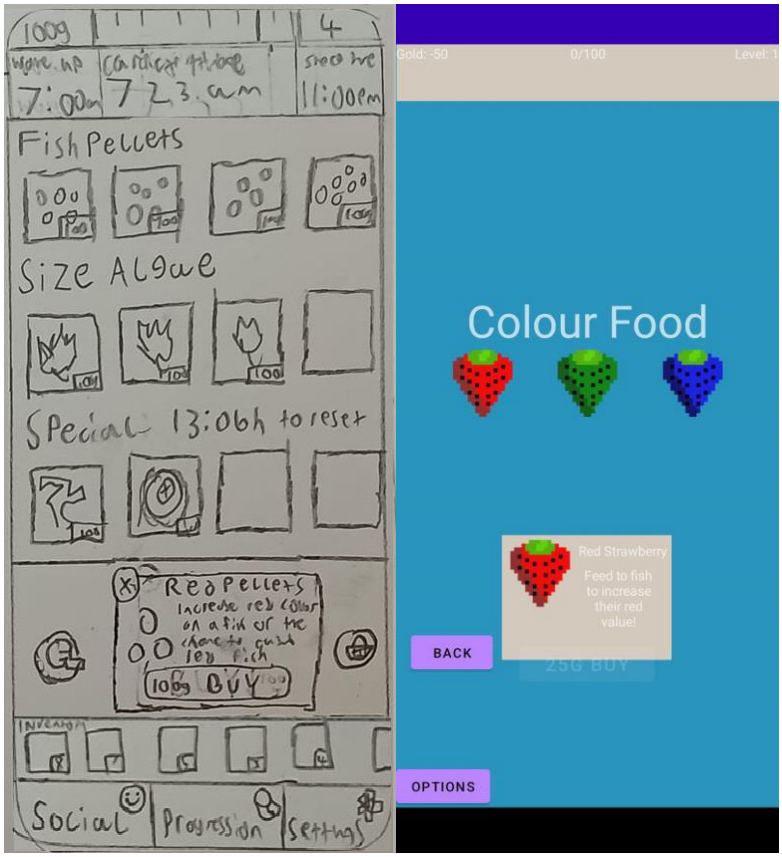


Figure 34 – Evolution of the shop screen

The options screen is primarily used to set the wake-up and sleep time, other planned features did not have enough time to be implemented (Figure 35).

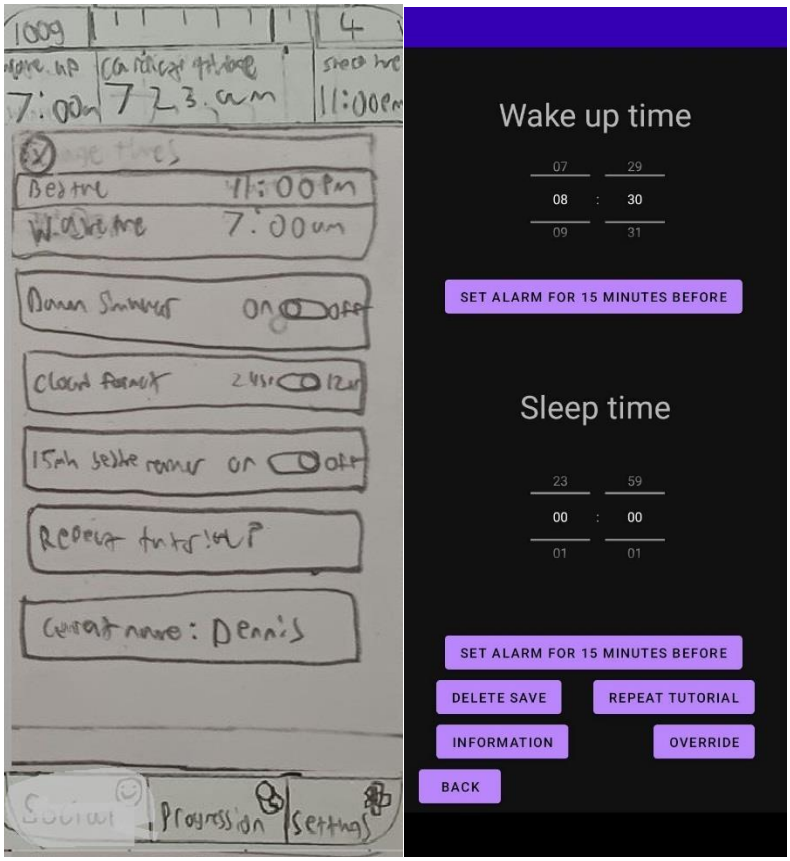


Figure 35 – Evolution of the settings screen

The friends list was never added due to the removal of online features so there is nothing to compare it to, the space where its button would have gone was replaced by the sleep button.

Initially, there was to be a testing period over the Christmas break where I would use the app myself to find bugs and test features, a period of distraction led to this not transpiring. At the start of February, the app could not consistently track the user's sleep, there was a bug in time calculations which caused crashing along with this the Service would not be initialised in some runs. Although fixed, these bugs could have been noticed earlier with the aforementioned testing period.

The program accurately tracking sleep was vital for its functionality and the experiment, therefore to assure the process contained no further bugs, a series of black box tests were created. JUnit was used to confirm that wakeUpTime and sleepTime were both being set to appropriate values such as wakeUp being later than sleep and the two always being within 24 hours of each other. The statistics page was tested by supplying large numbers of dummy data and seeing if the system worked appropriately. Gameplay was tested to assure every quest was completable and that users could level up.

Formal testing table displayed in the appendix.

With the app now stable, an internal release was pushed to the Google Play store using the Google Play Console on March 8th. This allowed testing but it was restricted to those who had been given explicit approval limiting its reach. With releases, Google provides a pre-launch report powered by Robo Test which uses an AI to methodically test every UI element in the app and report problems (Firebase, 2023). Although several crashes and accessibility issues were detected, the decision was made to push the app to open testing to allow more players further time to use it before the submission deadline. After completing several required surveys on how data was handled along with creating a privacy policy, the app was released to open testing as Fish And Sleep on March 14th (Figure 36).

Link to the Privacy Policy in the appendix

Further Store Listing Details

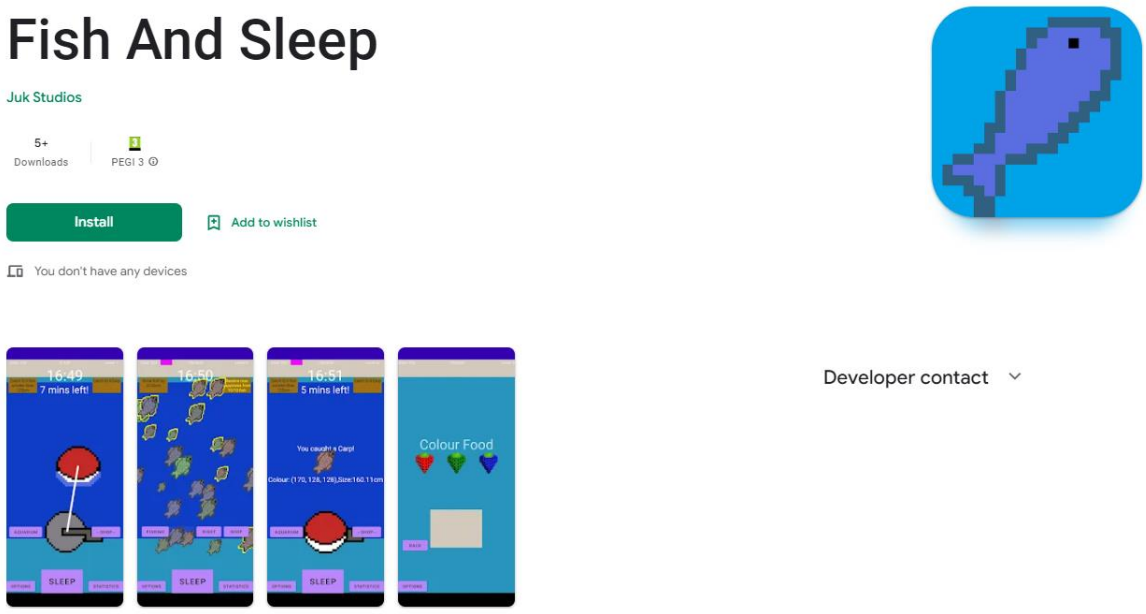


Figure 36 - Google play store listing for Fish and Sleep. See the appendix for more information on the Store Listing.

Project Evaluation

To obtain information on the project's success in helping depressed people keep to a regular sleeping schedule, an experiment was performed. The process was flawed leading to unusable results.

3.1 Experiment

Fish And Sleep was used for investigation into gamification's ability to encourage those with mood disorders to wake up and sleep at consistent times, this was essential for evaluating the success of the app. It was hypothesised that gamification can reinforce the habits of waking and sleeping at similar times for people regardless of mood disorders. Therefore, depressed people will have a more consistent sleep schedule using the gamified sleep assistance app, Fish and Sleep, than without it. Consistent here means, waking and sleeping close to a desired time each day and waking in the night less frequently.

Participants tracking their sleep could be unreliable, instead, the times the user's phone was active, which was already gathered by the app, would be used as an interpretation of sleep pattern. This is assumptive, a user can be awake and not using their phone however, time constraints made using this simpler method a requirement. An experiment mode was implemented, this locked gamification features (fishing minigame and the aquarium) for four nights. Records gained before and after gamification was introduced would be compared and differences studied. These would be extracted from participants' phones using a print button in the statistics menu which translated all Record objects into a text file (Figure 37).

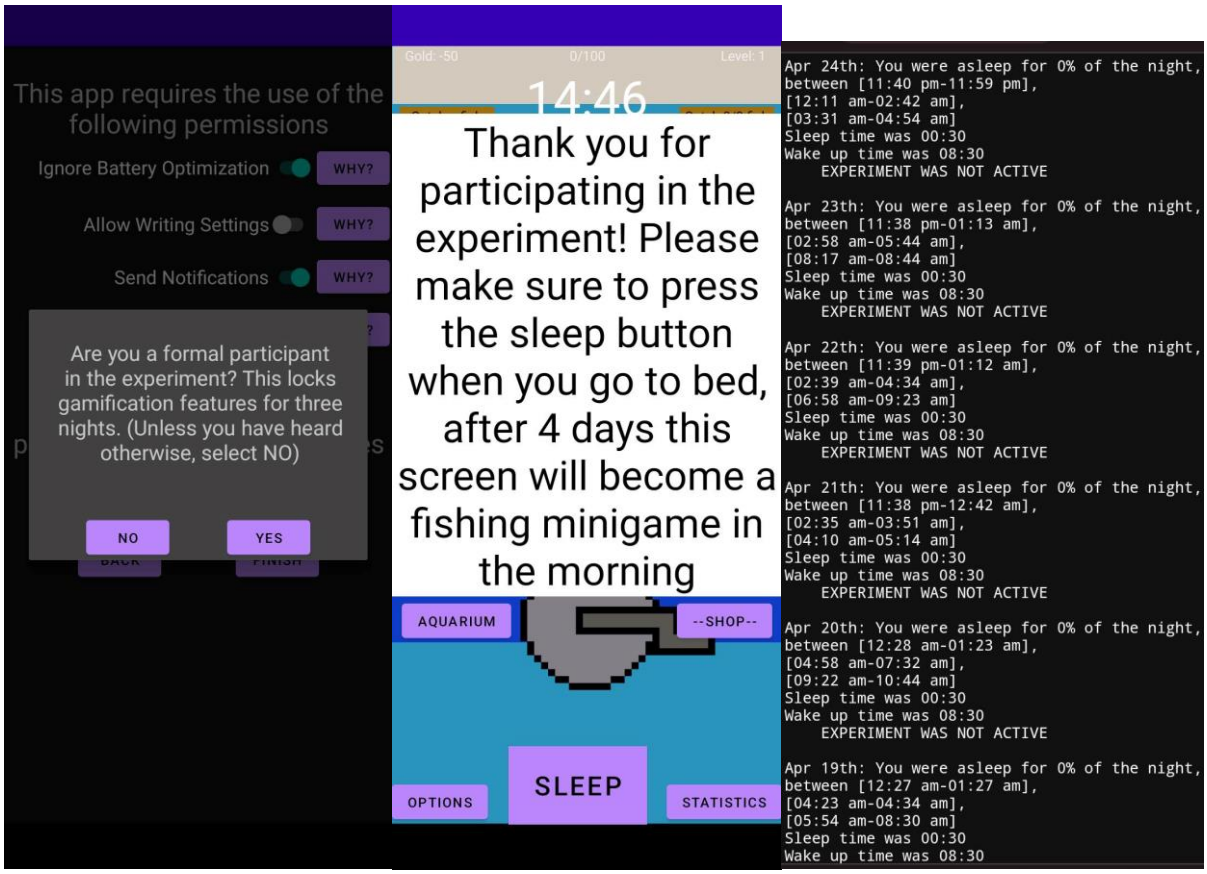


Figure 37 – From left to right; The popup asking if the user is a member of the formal experiment, the custom message that appears when the fishing game is not active and an example text file.

A pre-experiment survey was created, discussing sleep patterns, and a post-experiment survey, reiterating the questions, these would be compared to further support any results from the Records. The second survey also asks questions about their experience using Fish And Sleep, the app Alarmy can help wake people consistently but in an unpleasant way. It was important to evaluate if the solution was enjoyable as helping depressed people wake up should not further any emotional turmoil they feel. Both surveys were kept short and provided information on the purpose of each question to keep engagement high.

The full Surveys are included in the appendix.

The decision was made to not explicitly use participants with mood disorders for the experiment. It would have been difficult to find large numbers in this niche group and causing privacy concerns as sensitive questions would need to be asked. The experiment should therefore be seen as an evaluation of Fish and Sleep's effect on people generally.

3.2 Results

3.2.1 Experiment Results

Several issues occurred including, a small sample size, little time using the app, and general confusion which compromised results (Explored further in the discussion). The predominant issue was a fault causing the app to only stop tracking timings when first opened after the wake-up time. If a participant did not open Fish and Sleep in the morning, the Service would run throughout the day and cause unrepresentative results. Despite this known problem, the first timing from the Record was taken as when the user went to sleep and the last as when they woke up. Looking at the results clearly show this to be incorrect, but assuming the wake-up times myself would be biased so the stricter method was enforced. The percentage awake was calculated between sleep and wake time and was unaffected by the glitch.

Using gamification compared to without it, participants fell asleep **46 minutes** closer to their desired sleep time and stayed asleep **8.89% longer**, however, they would wake **15 hours and 12 minutes later** than desired.

Falling and staying asleep became more consistent with gamification introduced supporting the hypothesis but the ability to wake up drastically decreased. A faux conclusion could be "Fish and Sleep improves consistency in going and staying asleep however, it severely hurts their consistency in waking up". This experiment had three participants with only 5 non-gamified nights and 4 gamified nights observed compounded with bug issues and general confusion.

Due to the minute size of the results they fail to either support or oppose the hypothesis and this experiment holds no indication of gamification's ability to enforce consistent sleep schedules.

The full spreadsheet of Experiment Results is included in the appendix.

3.2.2 Survey Results

The experiment was flawed but the survey results were unaffected by the issues giving some information on Fish and Sleep's effects, the findings are displayed below (Figure 38).

Participant No.	Results
001	Went to sleep an hour earlier and went from rarely checking their phone to never.
002	Started sleeping an hour later, waking a half-hour later, and their sleep schedule differed from 1-2 times a week to 3-4 times.
005	Showed no change using the app.

Figure 38 - Result's from comparing the pre-experiment and post-experiment sleep pattern questions.

Participant 001 supports the hypothesis suggesting Fish and Sleep can support consistent sleep schedules and reduce overnight phone usage. However, Participant 002's sleep schedule became less consistent using Fish and Sleep and Participant 005 had no change at all. Therefore, these results fail to support the initial hypothesis.

The full Survey Results are included in the appendix.

The following survey was given to all users of Fish and Sleep and includes results from those not part of the formal experiment. A series of questions asked people their opinion on Fish and Sleep's effectiveness, for waking up, 50% found the app helpful (Figure 39).

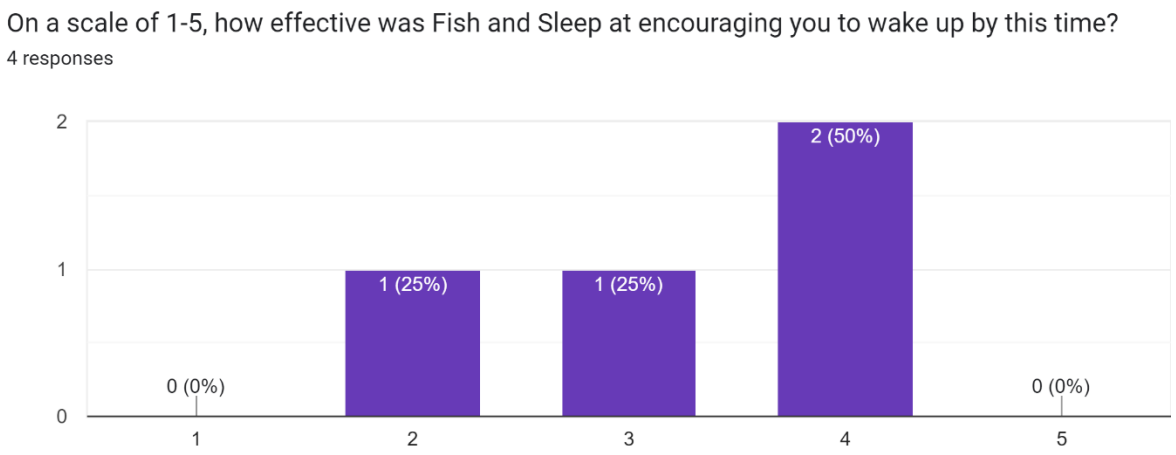


Figure 39 - Bar chart from the post-experiment survey results ('this time' refers to the user's wake-up time), the scale is from 'Not very helpful' to 'Very helpful'.

Further responses indicate that Fish and Sleep is more effective at encouraging people to sleep than wake which was echoed by the experiment. The results below show half of the respondents find Fish and Sleep effective at encouraging them not to use their phone past their sleep time (Figure 40).

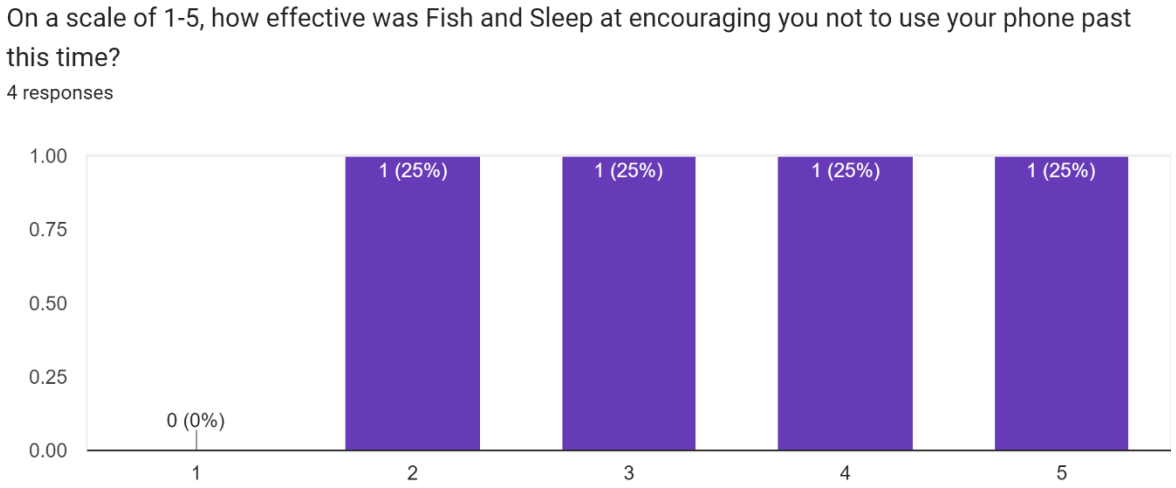


Figure 40 - Bar chart from the post-experiment survey results ('this time' refers to the user's sleep time), the scale is from 'Not very helpful' to 'Very helpful'.

These results do support the hypothesis, however, a larger sample size would be needed to give a reliable conclusion.

The other questions of the post-experiment survey address users' feelings towards the app along with their suggestions to improve it (Addressed in the

discussion). Most participants did not find the app confusing, the time spent on the tutorial may have contributed to this along with consistent UI design, this area can be considered a success (Figure 41).

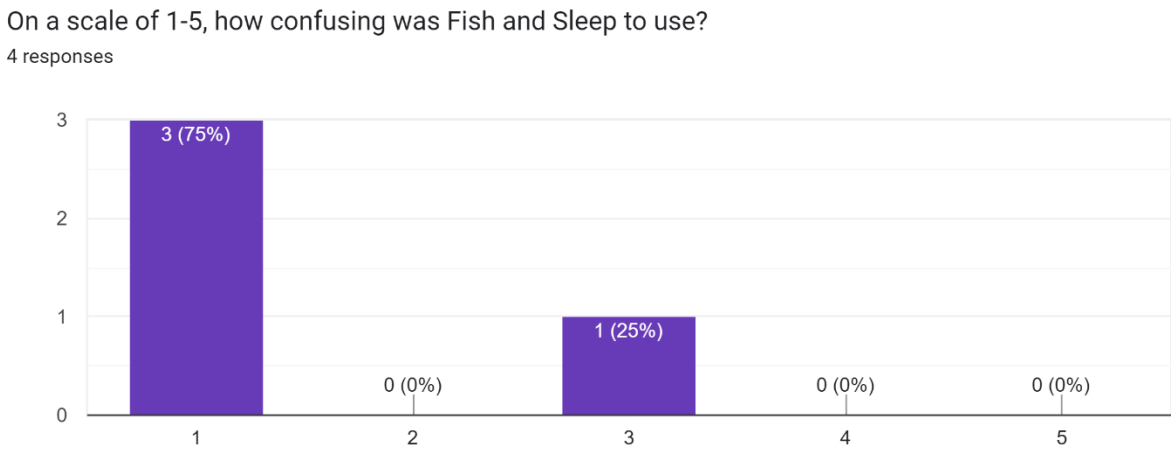


Figure 41 - Bar chart from the post-experiment survey results, the scale is from 'Not very confusing' to 'Very confusing'.

Participants responded that they enjoyed the fishing game with one writing it was "a nice way to start my day". Creating a comfortable environment that people wanted to wake up to was a goal of the project and has shown success (Figure 42).

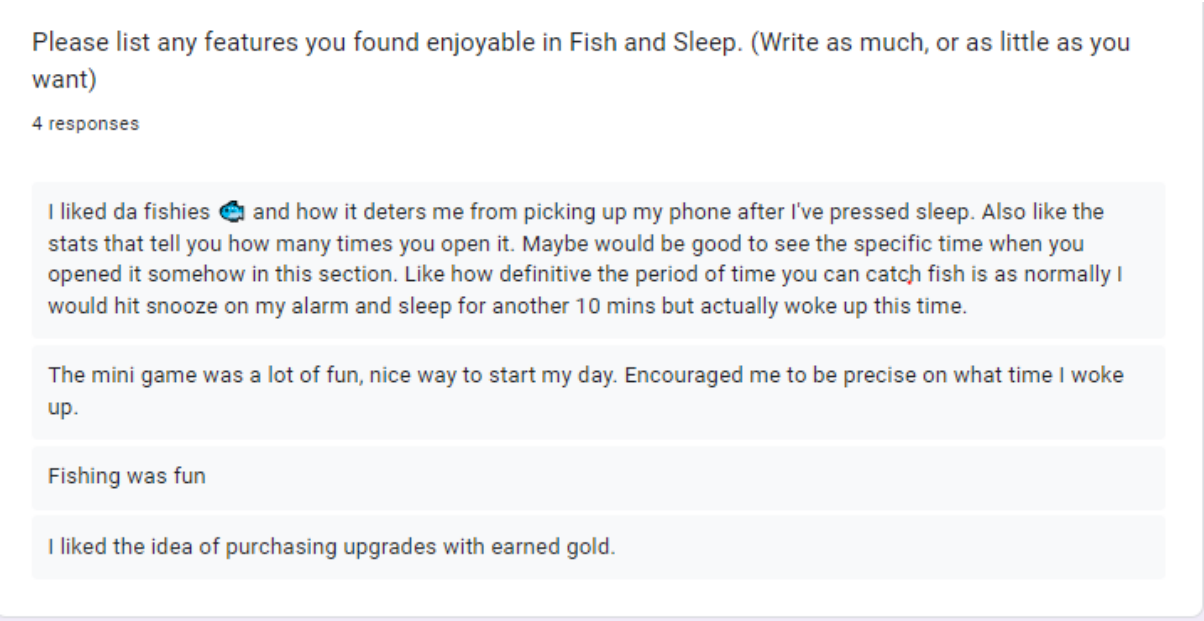


Figure 42 - 4 different responses to a question in the post-experiment survey

A common complaint was with the period users were allowed to fish for, some users wanted a shorter time while others wanted it longer. Original plans made

in the scenarios allowed the fishing period to be customised and this may have improved people's experience of the app (Figure 43).

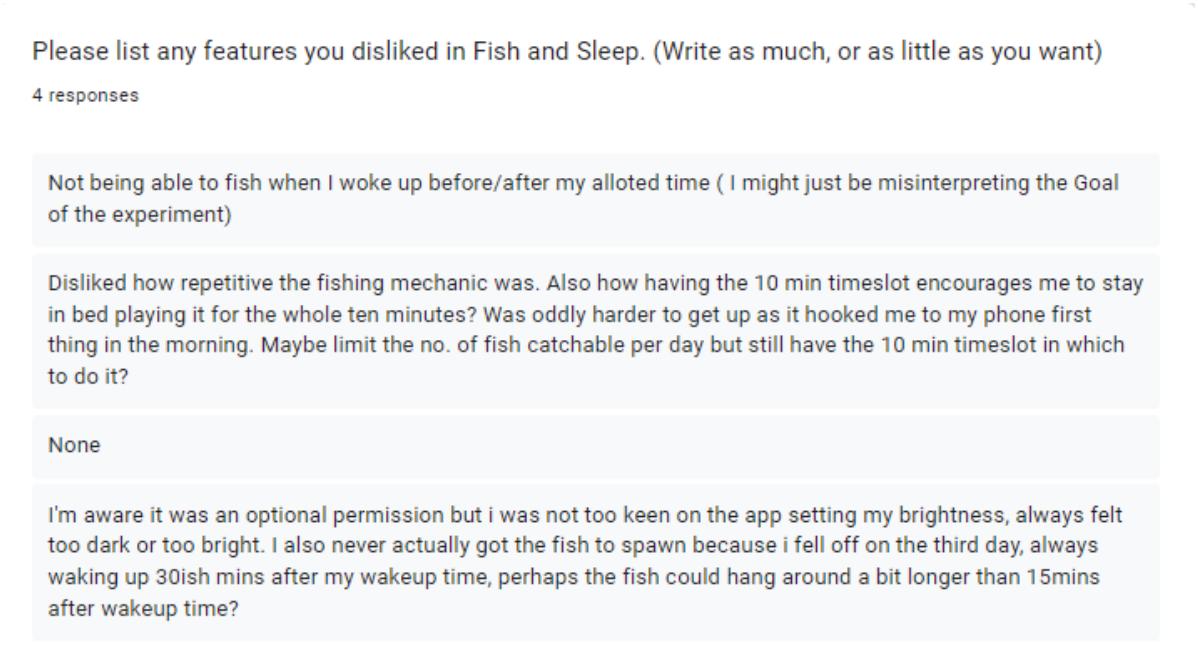


Figure 43 - 4 different responses to a question in the post-experiment survey

3.2.3 Summary of Results

The experiment was unsuccessful at supporting Fish and Sleep's ability to enforce consistent sleep schedules. Some data was rendered useless due to the time-tracking bug and this problem was exacerbated by a small sample size and short testing window. Although no support for the hypothesis was found, the surveys showed evidence that people generally enjoy gamification. This emphasises findings in a paper written on Sleepy Bird that found the average feeling towards the gamified alarm system to be happy and delighted (Dias, 2018). Fish and Sleep is a finished product and some responses to it are positive meaning the project was a partial success. However, a larger and stricter experiment would need to be performed to find evidence of its ability to enforce consistent sleep schedules.

Discussion

Creating Fish and Sleep was a great opportunity to explore app development and use computer science to solve a problem however, poor management led to issues in the evaluation and refinement of the project.

4.1 Achievements

The research was particularly successful; I had assumed that depression and sleep were connected and saw gamification as a possible solution, but this needed to be demonstrated through linking sources effectively. Sleep was established as vital for mental and physical health along with being something depressed people struggle with. Existing solutions (CBT-I and sleeping pills) were found to hold flaws; therefore, the unexplored field of gamification should be trialled targeting those with depression. The basis for the project was well-founded and a strong case was made for gamified sleeping apps and their potential to help people with mood disorders. The Pokémon Company plan to release a gamified sleeping app in the Summer of 2023 showing interest from large video game companies in this area (Anon., 2023).

The process of creating an engaging application that helped people sleep was worthwhile leading to a clear specifications list. Research into why users play using the user types hexad along with exploring methods of healthy sleep was efficiently translated into requirements. This allowed focusing on essential elements first, allowing the completed app to, at minimum, be engaging and useful. I found the application of Computer Science to help depressed people sleep gratifying, being able to use learnt skills to address a problem. Although the finished product was only partially successful, the requirements were well established and more development time could lead to further success.

Much was learnt in the experience of creating and releasing a mobile app. Learning to develop using Android Studio was very beneficial along with tackling the unique challenges of mobile devices such as differing screen sizes, lack of physical input and the possibility of being closed at any time. The process of deployment also gave me experience in areas such as the creation of privacy policies and the differences between internal, closed and open testing. My skills and confidence in mobile development have improved greatly during this project.

4.2 Problems

4.2.1 Management

Project management was generally quite poor, although the app was finished key features were missing and evaluating the finished app was given little time. Originally, the planning was to use a Gantt chart (Figure 44) which would have allowed future goals to be adjusted if a previous one was met however, this

would not be created.

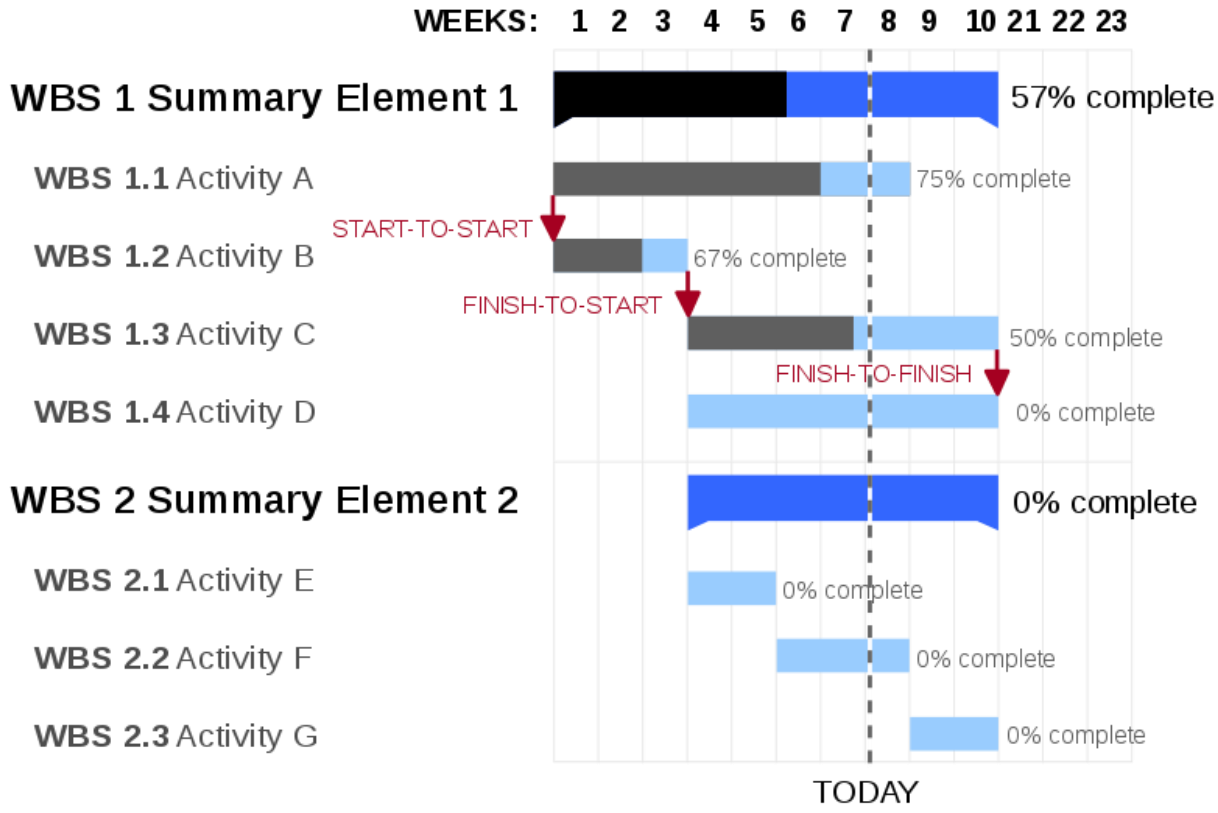


Figure 44 - An example of a Gantt Chart (This image is in the public domain and was taken from https://commons.wikimedia.org/wiki/File:GanttChartAnatomy.svg)

Instead, several individual plans were created whenever needed, specifying immediate tasks and assigning them a week to be completed. This method was flexible, which was useful as I did not have much experience developing an Android application and some features proved harder or easier to implement than expected. However, a view of the larger project was lost and more attention was placed on programming than other areas.

The first plan (Figure 45) created on the 9th of October is the only one to provide a detailed overview of every aspect of the project with their start and finish date. This provided clear goals to work towards and overall awareness of elements outside of programming, however, reaching certain goals would take far less time than expected. Creating the alarm section of the app (Point 4) only took a week to complete leading to much of the game being finished before Christmas. The rigidness of this plan meant development goals did not adjust to the newfound time leading to a large gap of unproductiveness over the break. The creation of a more flexible plan, such as a Gantt chart, may have alleviated these problems allowing goals to be shifted per changing circumstances.

1. All Research – Start 09/10/22 – End 27/10/22 – Includes reading sources, investigating relevant apps, writing the literature review, writing the app review.
2. Project Architecture – Start 27/10/22 – End 10/11/22 – Includes creating screen diagrams, reworking my scenarios, creating object architecture, performing game design.
3. Prototyping – Start 03/11/22 – End 17/11/22 – This will overlap with my project architecture allowing feedback design. Includes finding APIs to run and creating a working phone app I can develop on.
4. Creating Alarm – Start 17/11/22 – End 15/12/22 – Create a working alarm app and some basic skeleton of my app.
5. Creating Game – Start 20/01/23 – End 17/02/23 – Creating the actual playable systems. There is a break here for Christmas though I may start earlier depending on the progress of my alarm.
6. Finish Implementing Features – Start 17/02/23 – End 10/03/23 – Add anything else I have time to add that I've deemed important e.g. social aspects, customization. This will be the final state of my app.
7. Final Project Report – Start 10/03/23 – End 30/03/23 – Finish any final paperwork, this leaves me finishing the project 2 weeks before the final hand-in date giving me lots of time to catch up if something has gone wrong.

Figure 45 – The first created plan.

A more detailed version of the first three sections (research, architecture and prototyping) was also created. This plan (Figure 46) failed to be very helpful, it was again very rigid and most of these tasks were finished before expected.

No.	Position	date	Milestone
1	1	21/10/2022	All Research
2	1	04/11/2022	Project Architecture
3	1	04/11/2022	Prototyping
4	1	09/12/2022	Actual App making

To add more Milestones, add a new row above this one.

No.	Start Date	End Date	Task
1	03/10/2022	21/10/2022	Read through all sources
2	10/10/2022	21/10/2022	Investigate relevant apps
3	17/10/2022	21/10/2022	Write Litreature Review
4	17/10/2022	21/10/2022	Write App Review
5	28/10/2022	04/11/2022	Create screen diagrams
6	28/10/2022	04/11/2022	Create Objects Architecture
7	21/10/2022	28/10/2022	Find APIs
8	21/10/2022	28/10/2022	Create app I can run on a phone

To add more Tasks, add a new row above this one.

Figure 46 - Detailed plan of the first three sections

A diary was kept which would be updated whenever progress was made on the project allowing a constant reviewal of set goals (this diary is contained within

the git repository). Several plans were created in it, one was made after prototyping was complete on the 9th of November (Figure 47), and the dates indicate when the part should be finished. Buffer weeks ending on the 4th and 22nd of December were also included allowing time to finish a feature if it took longer than expected. These tasks were vague and more detail could have been provided to each to make development clearer, reviewing the goals should also have been more formal with weekly reviews as opposed to occasional checking. Nonetheless, this plan was effective as all features of the game were implemented by the 22nd.

13th Nov – Finish all Design

20th Nov – Working Fishing game (Without motion controls)

27th Nov - Working Aquarium

4th Dec – Optimise and fix bugs in Fishing game and Aquarium

11th Dec – Add time locked features for fishing and aquarium

18th Dec – Add Quests and Experience and Shop

22nd Dec – Optimise new features again

Figure 47 - Tasks to complete before Christmas.

The second plan (Figure 48) was a list of programming goals for completion after the Christmas break, however, this was ineffectual for several reasons. The description of the goals was unclear, and I was unable to recall what was meant by some of the statements after two months. They lacked estimations for how long each would take to complete, this could have shown sooner the lack of time needed to include online features. There was also a lack of priority for the goals, the bottom element (may be problems with timelock) was an app-breaking bug which should have had complete focus when development restarted.

I will implement in the new year, they are;

- Add tutorial
- Add stats screen
- See fish in the aquarium

Then online stuff

- Make the brightness go down or something when you sleep
- Make am pm work
- Make notifications lead to the app
- Still may be problems with timelock when dealing with midnight

Figure 48 - Tasks to complete after Christmas.

I had planned to test the app on myself over the break to find bugs and fine-tune values, however, this was not performed. Development restarted on February the 6th with a new plan (Figure 49) created to help complete the final elements of the project, its creation coincided with the decision to not implement online features. This plan's design closely followed the previous one with more detail added to each element giving a clearer status on the goal's progress. This plan again proved effective with a clear idea of tasks to complete each week and programming would be successfully finished by March 5th.

My plan at the moment, in finishing elements of the app is as follows;

Feb 12th – Add stats page, tutorial when loading for the first time, make brightness change

Feb 19th – Finish off any features not finished last week, make notifications work and fix bugs

Feb 26th – Polish game, add simple sound effects, some more graphics and make the app look professional

Mar 5th – Work started on User testing, App Completely finished

I can hopefully work around this plan to finish the app by the end of the month.

Figure 49 - Tasks to complete over February and March

The key issue in management was the lack of an overarching plan leading to poor focus outside of programming. Plans were created but the length of tasks

would be overestimated and underestimated due to inexperience in mobile development. The project was enveloped in implementing game features and little time was left for GUI design, creation of art assets and inclusion of online features. The experiment was the key area that suffered, little time could be spent finding participants and their duration using the app was short. As suggested a Gantt chart could have allowed the re-scheduling of future tasks when one proved easier or harder to implement. For future projects, undertaken in an unfamiliar domain a more flexible management schedule will be used so that unexpected completion times do not negatively affect the overall project.

4.2.2 Other Problems

Little time was spent creating and performing the experiment due to poor management. Testing had to begin as soon as possible leading to a short window to find candidates and only three volunteers being found. Finding participants with mood disorders in such little time was considered too difficult, the project had aimed to target those with depression but the evaluation of the project failed to include their input. Little data was collected as time spent using the application was short at only a week and a half, this was worsened as participants would often forget to press the sleep button which was needed to start tracking. This is a fault of requiring user input to start the Service as opposed to a lack of time but had the testing period been longer, missing days would have been less impactful making the issue less severe.

There were problems in the design of the experiment itself. As mentioned before, it was discovered upon collecting data that the app would not stop tracking when the power button was triggered until the app was opened after the wake-up time. For example, participant 002 pressed the sleep button but did not open the app until the day after the next causing the Service to run for over 24 hours. This had to be interpreted as them sleeping for the whole day as assuming their wake-up time using a selected result would be biased. This bug also prevented results from expressing if a user had woken up before their wake-up time which is a possibility that should have been considered. I do not believe this paper should be used to prove gamification’s ability to create consistent sleep schedules. If I was to perform the experiment again, I would work to more accurately express when a user was asleep along with performing a pilot study to discover issues beforehand.

Finally, I feel strongly that time should have been dedicated to a testing period, this was planned but never transpired. The bugs in the program (such as the one from the experiment) could have been spotted earlier and fixed before they became a problem. People also noted that the fishing game’s length felt

incorrect, the use of the app was required to grasp the optimal length of the experience which is why it should have been tested before a wide release. In future, I will be dedicated to using the project myself or organising a testing period, to identify problems that can be missed without proper use of the program.

4.3 Future work

Users provided feedback in the post-experiment survey as to what they felt could be added, I have used this feedback and my own thoughts to discuss what additions could be made to the project (Figure 50).

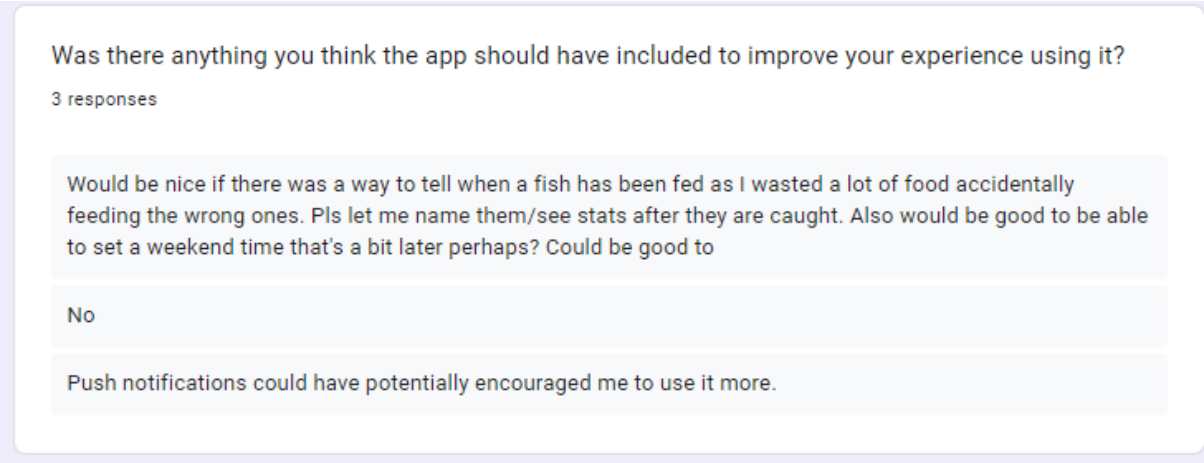


Figure 50 - responses to a question in the post-experiment survey

The key element missing which may have been beneficial is a lack of any online features. As described in the user requirements section, many people desire online interaction in their video games to keep them engaged. Looking at the top ten most played video games by player count, seven involve interaction with other players remotely in some way (Wikipedia, 2020). SleepTown includes collaborative features in its solution further proving that engagement can be increased by interaction with other players. Therefore, with more development time, the unused shared aquarium along with gifting and trading with other users should be implemented.

There were issues with users remaining in bed while using the app that could have been fixed with the initially planned motion controls. One user noted how the “timeslot encourages me to stay in bed playing it for the whole ten minutes? Was oddly harder to get up as it hooked me to my phone first thing in the morning”. Alarmy includes games that encourage people to wake up and stay awake by giving them an active task such as performing squats or taking a picture of a location. The focus of the project was on waking people up, but getting them out of bed is as important, therefore, implementing the planned

motion controls could improve the app’s ability to enforce a consistent sleep schedule.

I also believe in introducing more elements of customisability in the program to better suit individuals. As noted, users’ opinions differed on how long they believed the fishing period should be, there were also desires for wake-up times to change for certain days such as weekends. Although an extendable fishing period is abusable and a consistent sleep schedule is encouraged by experts (Anon., 2012), customisability holds benefits. Users would have input on what conditions make them sleep and wake most effectively allowing the app to be beneficial for a wider range of people. I believe that applications should serve as tools that a user can customise to their liking and not assume a designer's decision to always be correct.

Conclusion

A phone application using gamification to tackle sleep problems was successfully designed, implemented and released for general use. The final product rewards users for waking and sleeping at a specified time through collectable fish and in-game currency. It includes a statistics page which allows a comprehensive view of changing sleep schedules while using the app. Several planned features failed to be added (Motion controls, online) but this was a risk from the project's inception and a focus on essential elements led to a completed product.

Related works support gamification as capable of creating a consistent sleep schedule and the desire was to prove the same for Fish and Sleep. Poor management of areas outside of programming would give little time to creating and performing the experiment, combined with a fault in time calculation, much of the data was unusable. An evaluation was still performed on completed surveys which found users generally enjoyed the gamified system, but showed no indication that their sleep schedule became more consistent.

Overall, the project was able to produce a final design that matched the specification though there is no evidence to support its ability to help depressed people keep consistent sleep schedules. This paper holds value in its description of how to create a gamified solution to sleep problems along with highlighting it as particularly promising for those with mood disorders. Fish and Sleep was a promising attempt at applying this and I hope it encourages further exploration of this area.

Sources

References

Anon., 2012. *Tips for Getting a Good Night's Sleep.* [Online]
Available at: https://magazine.medlineplus.gov/pdf/MLP_Summer2012web.pdf
[Accessed 7 Apr 2023].

Anon., 2023. *Pokémon Sleep.* [Online]
Available at: https://www.pokemon.com/us/app/pokemon-sleep/

Cotton V, P. M., 2018. Gamification Use and Design in Popular Health and Fitness Mobile Applications. *American Journal of Health Promotion,* pp. 448-451.

Developers, A., 2019. *Activity.* [Online]
Available at: https://developer.android.com/reference/android/app/Activity

Developers, A., 2019. *Intent.* [Online]
Available at: https://developer.android.com/reference/android/content/Intent

Developers, A., 2019. *View.* [Online]
Available at: https://developer.android.com/reference/android/view/View

Developers, A., 2019. *View.* [Online]
Available at: https://developer.android.com/reference/android/view/View

Developers, A., n.d. *Save key-value data.* [Online]
Available at: https://developer.android.com/training/data-storage/shared-preferences

Dias, L. B. J. a. V. H., 2018. Gamification and serious games in depression care: A systematic mapping study.. *Telematics and Informatics, 35(1),* pp. 213-224.

Fang, H. T. S. S. J. a. S. A., 2019. Depression in sleep disturbance: A review on a bidirectional relationship, mechanisms and treatment.. *Journal of Cellular and Molecular Medicine,* pp. 2324-2332.

Firebase, 2023. *Firebase.* [Online]
Available at: https://firebase.google.com/docs/test-lab/android/robo-ux-test

Go, P., 2016. San Francisco: Niantic.

Hamari, J. K. J. a. S. H., 2014. Does Gamification Work? -- a Literature Review of Empirical Studies on Gamification.. *2014 47th Hawaii International Conference on System Sciences. ,* pp. 3025-3034.

HILDITCH, C. D. J. a. B. S., 2016. Time to wake up: reactive countermeasures to sleep inertia. *Industrial Health, 54(6),* pp. 528-541.

Ilhan, E. S. B. a. H. H., 2016. 2016. *Creating Awareness of Sleep-Wake Hours by Gamification. Persuasive Technology,* pp. 122-133.

Kanady, J. H. A., 2015. Development and Validation of the Sleep Inertia Questionnaire (SIQ) and Assessment of Sleep Inertia in Analogue and Clinical Depression. *Cogn Ther Res 39,* pp. 601-612.

Kaplan, K. T. D. a. H. A., 2018. Rise and shine: A treatment experiment testing a morning routine to decrease subjective sleep inertia in insomnia and bipolar disorder. *Behaviour Research and Therapy, 111,* pp. 106-112.

Krauchi, K. C. C. a. W.-J. A., 2004. Waking up properly: is there a role of thermoregulation in sleep inertia?. *Journal of Sleep Research, 13(2),* pp. 121-127.

Marczewski, A., 2015. Marczewski's Player and User Types Hexad. In: *Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design.* United Kingdom: Gamified Uk, pp. 65-80.

McGonigal, J. K. E. a. W. C. O., 2017. *SuperBetter.* Warszawa: Wydawnictwo Czarna Owca..

NHS, 2019. *Insomnia.* [Online]
Available at: https://www.nhs.uk/conditions/insomnia/

NHS, 2019. *Treatment - Clinical depression.* [Online]
Available at: https://www.nhs.uk/mental-health/conditions/clinical-depression/treatment/

NHS, 2022. *Cognitive Behavioural Therapy (CBT).* [Online]
Available at: https://www.nhs.uk/mental-health/talking-therapies-medicine-treatments/talking-therapies-and-counselling/cognitive-behavioural-therapy-cbt/overview/

Nutt, D. W. S. a. P. L., 2008. Sleep disorders as core symptoms of depression.. *Dialogues in Clinical Neuroscience, Volume 10, Issue 3,* pp. 329-336.

Oh, K. S. J. K. J. a. K. M., 2020. Analysis of a Wake-Up Task-Based Mobile Alarm App.. *Applied Sciences, [online] 10(11), p.3993. doi:10.3390/app10113993.*.

Out, T. S. T., 2012. Los Angeles: EA Mobile.

StatCounter, 2022. *Mobile Operating System Market Share Worldwide.* [Online]
Available at: https://gs.statcounter.com/os-market-share/mobile/worldwide

Tassi, P. a. M. A., 2000. Sleep inertia. *Sleep Medicine Reviews 4(4),* p. 341–353.

Trotti, L. M., 2017. Waking up is the hardest thing I do all day: Sleep inertia and sleep drunkenness. *Sleep Medicines Reviews*, pp. 76-84.

Walker, M., 2017. *Why We Sleep : The New Science of Sleep and Dreams.* UK: Penguin Books.

Walker, M., 2019. *How To Improve Your Sleep | Matthew Walker* [Interview] (29 Jan 2019).

Wikipedia, 2020. *List of most-played video games by player count.* [Online]
Available at: https://en.wikipedia.org/wiki/List_of_most-played_video_games_by_player_count

You, Y., 2020. *The gamification of sleep : a case study on user engagement with SleepTown in everyday(night) life.* [Online]
Available at: https://lup.lub.lu.se/student-papers/search/publication/9009216

Code Sources

For this project, many pieces of code were adapted from existing sources to learn how to structure and use the various modules and classes that android studio implements. The main sources were Stack Overflow, geeksforgeeks.org, tutorialspoint.com and Android developers. Comments have been left in the source code in appropriate places to indicate where adapted code is used and where it is from, note that these comments may have been missed if the code was reused elsewhere. The full list of code sources is found below.

Obtaining screen size: https://www.geeksforgeeks.org/how-to-get-screen-width-and-height-in-android/

How to use Runnables: https://stackoverflow.com/questions/62489406/how-to-continuously-move-object-in-android-studio

Dragging and Dropping Imageviews: https://www.tutorialspoint.com/android/android_drag_and_drop.htm

Creating an ImageView dynamically:

https://www.geeksforgeeks.org/how-to-insert-image-to-screen-at-the-touched-coordinates-in-android/

Getting the centre of an ImageView: https://stackoverflow.com/questions/24954976/set-coordinates-as-center-of-imageview-android-programmatically

LERP function: http://www.java2s.com/example/java-utility-method/lerp/lerp-double-a-double-b-double-amt-f21dc.html

Change screen brightness: https://www.youtube.com/watch?v=-UOmi1AsfxA&ab_channel=NoobDeveloper

Embedding links in text: https://www.youtube.com/watch?v=tL9fCKC2lNA&ab_channel=LearntoDroid

Using pickers: https://abhiandroid.com/ui/timepicker

Scrollable Text: https://stackoverflow.com/questions/1748977/making-textview-scrollable-on-android

Get the date ending: https://stackoverflow.com/questions/4011075/how-do-you-format-the-day-of-the-month-to-say-11th-21st-or-23rd-ordinal

Creating popups: https://www.youtube.com/watch?v=4GYKOzgQDWI&t=780s

Obtaining write settings permission:

https://stackoverflow.com/questions/32083410/cant-get-write-settings-permission

Using viewPagers: https://www.geeksforgeeks.org/viewpager2-in-android-with-example/

Saving and Loading JSON files: https://levelup.gitconnected.com/how-to-create-an-android-app-that-persistently-saves-custom-objects-5362f277cb29

Write to a file: https://stackoverflow.com/questions/14376807/read-write-string-from-to-a-file-in-android

Save to the Downloads Folder:
https://stackoverflow.com/questions/28183893/how-to-store-files-generated-from-app-in-downloads-folder-of-android

Fix for path separator glitch:
https://stackoverflow.com/questions/5963535/java-lang-illegalargumentexception-contains-a-path-separator

Create a Dictionary: https://stackoverflow.com/questions/5692940/how-can-i-create-a-new-directory-on-the-sd-card-programmatically

Creating a Global Variable:
https://stackoverflow.com/questions/1944656/android-global-variable

Code for Semaphores: https://www.geeksforgeeks.org/semaphores-in-kotlin/

Creating Notifications: https://stackoverflow.com/questions/34517520/how-to-give-notifications-on-android-on-specific-time

Setting the Size of a Drawable Programmatically:
https://stackoverflow.com/questions/4609456/set-drawable-size-programmatically

Broadcast receivers: https://www.geeksforgeeks.org/broadcast-receiver-in-android-with-example/

Creating Services: https://developer.android.com/guide/components/services

Running a Function in Another Activity:
https://stackoverflow.com/questions/22869928/android-broadcastreceiver-onreceive-update-textview-in-mainactivity

Creating a LineView between two points:

https://www.youtube.com/watch?v=FsXzushEbp4&ab_channel=LearnShareAnythingAnyone

Appendix

Gitlab Repository Structure

Inside the root folder is a PDF called Jacob Bradford – “Fish and Sleep Project Diary”, this is the diary mentioned in the management section that was kept continually updated during the project. There are also diagrams at the end of this diary but the majority of these are included later in the appendix.

Enter the folder “app/src” for the code, the details of the subfolders are listed below.

- app/src/test/java/jukstudios/main/fishingproject - Contains the Junit tests.

- app/src/test/main/java/jukstudios/main – Contains the xml file defining certain global properties of the app

- app/src/test/main/java/jukstudios/main/fishingproject – Contains all the classes of the program organised in a subfolder

    - The root folder contains classes accessed throughout the program

    - Activity classes - Contains code run by the different activity screens

    - Notifications – Contains classes needed to create notifications

    - Objects – Contains stored objects such as fish or items

    - Services – Contains classes used to track when a user is asleep

    - Visual Objects – Contains classes used to render visual elements

- app/src/test/main/java/jukstudios/main/res – The layout subfolder here contains the activity xml files and the values subfolder contains all the strings

The project contains no branches other than master.

How to run the source code

The code is available publicly on the google play store (the Store Listing is here). To run the source code, Android Studio must be downloaded, and the program loaded into it. You can either connect an Android phone and enable debugging on the device to run it or download a virtual system within Android Studio to emulate the app. The following link gives information on creating an environment to run the program.

https://developer.android.com/studio/run

Scenarios

Scenario 1

Sophie is a 20 Year old University Student currently on their summer holidays, she suffers from low moods due to depression and has downloaded the app to try and help her stay positive in the mornings.

She opens the app for the first time and goes through a set up process, she puts her name and date of birth in and selects a username. The app explains that it

will set an alarm for a certain time and then allow the user to play a fishing mini game over a certain time frame for a high score. It says that this will help to engage the user's brain during the start of the day and encourage them to wake up. Sophie selects to be woken up at 9:15am and for the game to start at 9:30am and to last 1 hour. The app asks her if she would like to set any reminders to show up every morning a long with a list of default ones she can select. She selects "Drink a glass of water", "Todays going to be okay" and sets a custom one of "Take your antidepressants". The app then takes her to the main screen however the current time is 8:03pm, the app displays a message encouraging her to come back tomorrow. Before closing the app she goes into friends list panel and adds two friends who she knows have the apps by typing in their user names that they have sent to her. She closes the app and a few hours later goes to sleep.

The next day she wakes up to the alarm sound at 9:15am, she sits up and see's two notifications sent to her phone confirming that her friend requests have been accepted. She opens the app and goes into the social panel again, she sees a news feed and a panel saying her friend Daniel has caught a swordfish, despite not really understanding this, she sends a thumbs up by clicking the small button in the corner of the panel. It is now 9:25am, Sophie enters the gear tab and selects the only equipment she currently has; some common bait and a common rod. It is now 9:30am, the app sends a notification to Sophie's phone saying to start fishing, the fishing rod at the bottom of the screen appears allowing her to interact with it and start fishing. She fishes for the full hour, over this time the app spreads out the three notifications she has set to show up popping up in the top right of the screen. At 10:30am the fishing rod disappears and Sophie's score is shown and the app displays text reminding her to come back tomorrow. Later in the day Sophie receives a notification indicating her friend Daniel has liked her score for the day.

Scenario 2

Bruce is a 27 year old Office Worker, he has difficulty waking himself up in the morning which he needs to do to get to work, he has been using the app for a month. He has the app to set to wake him up at 6:45am and have the game begin at 7:00am and last for 15 minutes, he has it set to not wake him up on Sunday's as he does not work on that day. He has his reminders set as "Drink a glass of water", "Brush your teeth" and "Make Breakfast".

Today is a Thursday and Bruce wakes up at 6:45am from the alarm. He opens the app and goes into the shop section, different types of gear are selected for him to buy and screen be swiped to select more. He opts to buy with his points

the 5 “Legendary baits”. He closes the shop and goes into the gear page, he equips the “Legendary bait” and the “Amazing Fishing Rod”. While in the vault screen he touches a tab and looks at his previous high score. It is now 7:00am and the fishing minigame starts. Bruce plays the game while walking around his house and doing other actions, usually when prompted by his reminders such as brushing his teeth. It is now 7:15am and Bruce has brushed his teeth, made breakfast and gotten a glass of water, the game shows him his score today and indicates it’s a new high score for him.

Later that day, Bruce opens up the app again to set a custom reminder that will only play the next day called “Bring in present for Suzan”.

MoSCoW requirements

Fishing game

1. The app **must** generate fish to catch
   1.1 Fish **must** have a size, colour, and type
   1.2 Fish **could** have a pattern, different coloured parts
2. The app **could** generate other things to catch
   2.1 Packages **could** be generated which contain items
   2.2 Decorations **could** be generated for placing in a fish tank
   2.3 Messages in a bottle **could** be generated that require a player to throw their reel into a specific part of the water
3. The player **must** be able to swipe their finger to launch their reel into the water.
   3.1 The location the float lands **must** be determined by how far, and in what direction the, the player swipes.
   3.2 The app **should** show a gps at the bottom of the screen showing the x and y coordinates of the fishing float
   3.3 The app **could** have a wind speed every day that effects how long the float stays in one spot
4. Holding down the reel button after the reel has been cast **must** pull the fishing float back towards the centre.
5. The user **should** be able to enable motion controls to use instead of tapping

Aquarium

1. Players **must** be able to see their fish in an aquarium
   1.1 Fish **must** move around the aquarium
2. Players **must** be able to feed their fish items from their inventory by dragging the item into the fish tank

2.1 Colour pellets **must** cause a fish to start gaining a certain colour in pigment
2.2 Size algae **must** cause a fish to grow or shrink
2.3 The amount these values increase by **must** be equal to the fishes happiness value

3. Every fish **must** have an individual happiness value
   3.1 A fish's happiness **must** be proportionate to the amount of time a user spent on their phone the previous night
   3.2 A fish's happiness **must** be more effected by the phone if it has a higher rarity value
4. Fish **must** be tappable every day to generate an amount of experience proportionate to their happiness and rarity
5. Fish **should** be tappable after providing experience to see their stats
   5.1 The stats **should** show the fish's colouration
   5.2 The stats **should** show the fish's size
   5.3 The stats **should** show the fish's current happiness
   5.4 The stats **should** show the fish's current hunger
   5.5 The stats **could** show the fish's name
   5.6 The stats **could** allow the user to rename the fish
6. The app **should** hold at least 10 aquariums
7. Each aquarium **should** hold a maximum of 25 fish
8. Aquariums **could** be decorated with collectibles that a player finds

Shop

1. There **must** be a currency that the player can earn and spend at the shop
   1.1 Players **must** earn money from levelling up
2. There **must** be buyable items that will be placed in a user's inventory
   2.1 There **must** be pellets that change a fish's colour
   2.2 There **must** be size algae that increases or decreases a fish's size
   2.3 There **could** be a section that shows limited deals a player can buy
   2.4 Items in this **could** be fruit to increase a fishes happiness, decorations, care packages, messages in a bottle.

Social

1. There **should** be an online component to the game that allows players to interact with each other
2. Players **should** be able to add other users to a friends-list
   2.1 Players **should** be able to remove other users from a friends-list
   2.2 Searching for a player that doesn't exist **should** notify the user saying "that user does not exist"

3. Players **should** be able to gift fish to their friends
4. Players **should** be able to create a shared aquarium with up to 5 friends that gives rare rewards if they all don't turn their phones on after their bedtimes
5. Players **should** be able to see the aquariums of people on their friends list.
6. Players **should** be able to see the level of their friends

Quests and experience

1. There **must** be an experience system that allows a user to level up
   1.1 Catching a fish/object **must** give an experience value adjacent to its difficulty to catch
   1.2 Tapping a fish in the aquarium **must** give an experience value adjacent to its happiness
2. There **should** be a quest system that gives users goals to work towards. Quests will stay around until completed where they will give the player experience points and switch to a new quest.
   2.1 There **should** be fishing quests that reward a player for fishing specific targets (catch 5 red fish, catch 12 fish, catch a swordfish)
   2.2 There **should** be long term goals that require growing fish (increase a swordfish to its maximum size, create a golden fish (255, 216, 218)
   2.3 There **could** be an achievement system rewarding players for reaching esoteric goals
3. There **should** be a screen to monitor your progress
   3.1 The player **should** be able to see their current quests
   3.2 The player **could** be able to see every achievement and which ones they've gotten.
4. The player **should** be able to track how long they were awake and asleep for everyday through a visual means such as a graph

Settings

1. The user **should** be able to switch between 12hr time and 24hr time
2. The user **must** be able to change their sleep and wake times after they are initially set
3. The user **should** be able to access a tutorial that lets them learn how to use the app
4. The user **should** be able to turn on or off the dawn simulator
5. The user **should** be able to enable or disable 15 minute bed time warnings
6. The user **should** be able to enable or disable permissions in here
7. The user **should** be able to change their username

Java Code Lab tutorial link

https://developer.android.com/codelabs/build-your-first-android-app#0

Formal testing

1.x is tests in global

2.x is tests in the statistics page

3.x is tests on gameplay mechanics

Test No.	Purpose	Method	Actual Result	Changes made
1.1	Wakeup time should always be greater than Sleeptime whenever the two are set.	JUnit test going through every combination of waketime hours and sleep time hours	Wakeup time and Sleeptime could be the same value, this would be easy to abuse.	If Waketime and Sleeptime are the same, then an extra hour is added to Sleep
1.2	Wakeup time should always be set to the next day when reset with resetDate if the currentTime is past the sleepTime	Junit test going through every combination of waketime hours	Waketime was always correct, in some instances, it would be two days after the current date but this would be in events when someone was sleeping from 1:00am to 2:00am.	None
1.3	SetWakeAndSleepTime should always set the sleeptime to be after the currenttime but not by more than 24 hours	Junit test going through every combination of wake and sleep hours along with different current times	The sleepTime was always past the currenttime but within 24 hours	None
2.1	Moving between pages on the stats page should work correctly even with large numbers of records	Generate many different nights and try to move between pages.	The app was able to increment and decrement the records list even when it was incredibly large however, moving to the final page would cause an index error causing the wrong	The way the program calculated deciding whether to go to the next week or not was broken. I have changed the algorithm to detect if the next

			graphs to be displayed on a page.	date spans a new week fixing the issue
2.2	The stats page should be able to show a massive variety of different sleep times	Generate many different example nights and test they are correctly shown on the bar graph	The app could successfully show many different times, however, the popup showing what times the user was asleep was unable to show all the times when a large number were part of the Record.	The size of the text in the popup has been shrunk to provide more space. Graph blocks are rounded to be a multiple of 15 minutes but I reduced this to 1 minute to gain a more accurate graph after finding it didn't affect performance
2.3	When time asleep is calculated it should show the time the user was asleep between the sleeptime and waketime	Generate many different example nights and calculate if they have the correct percentage awake.	The formula to generate the time awake was inaccurate simply taking the last time and taking away the first time. This causes problems as if a user slept the whole night but went to bed 30 minutes before their sleep time, the program would register this as not being asleep.	The right formula was being used in the global object, so I added a total waketime variable to the record which it can use instead of having to work out the total wake time again.
3.1	Fish should give experience when tapped on	Tap fish in the aquarium and see if the experience amount increases	The experience amount did not increase	This has been added now
3.2	The experience bar should increase in level when it reaches its max amount	Increase the experience bar to the max amount and see if it increases	The experience amount reached 100 out of 100 but did not increase.	The level has been set to increase when experience is

				greater than or equal to the cap
3.3	All Quests should be able to be completed	Test every quest to make sure it's completable.	The quest to catch a fish smaller than a certain size was impossible to complete as fish were always above one of the numbers.	This quest has had its value's changed.

Privacy Policy

https://www.termsfeed.com/live/d76141c1-3850-4820-96ef-a4b43ff37579

Store Listing

https://play.google.com/store/apps/details?id=jukstudios.main.fishingproject

The description of the app is the following;

Fish and Sleep is a gamified sleep assistance app that use's catching and raising fish to incentivise you to wake up and go to sleep.

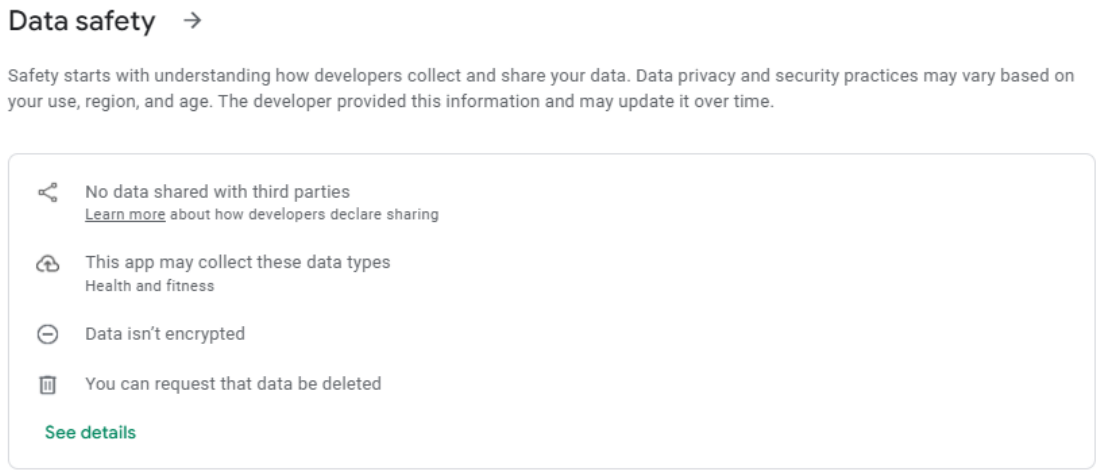
You can only catch fish for 10 minutes after you wake up so make sure you're up on time to catch them! Raise your fish in the aquarium, feed them different foods to grow their size and change their colour. Fish hate bright lights, using your phone past your bedtime will upset them so go to sleep to keep them happy! Fish and Sleep tracks the time's you were asleep giving you a comprehensive list of when your phone was active.

(Fish and Sleep can detect when your phone is turned on and off, it has no ability to record what you are using. Times your phone was off are saved but these are only for the user to see and are not shared with anyone else)

This project was made for my year 3 dissertation and it would be very helpful if you could fill out a short survey linked below on your experience using it!

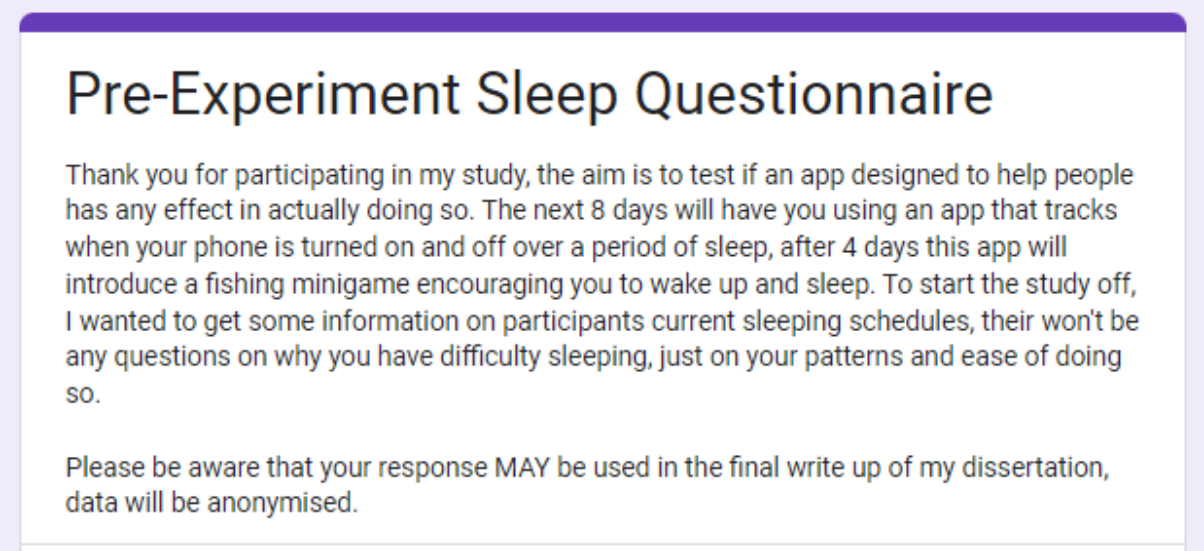
https://forms.gle/mqzHMwSvsFoWs2p7A

Data safety information is shown on the app page.



Surveys

Pre-experiment sleep Survey



Sleeping Patterns

Question's about your sleeping habits

What time do you normally start going to sleep by on a weekday? (As in getting into bed and closing your eyes) *Use 24 hour time.* *

Time

__ : __

What time do you normally wake up in the morning on a weekday? *Use 24 hour time.* *

Time

__ : __

How long does it normally take you to get out of bed after you've woken up? *

- ◯ I get out of bed immediately
- ◯ Up to 15 minutes
- ◯ Up to 30 minutes
- ◯ Up to an hour
- ◯ more than an hour

How often during a week, including weekends, does your sleep schedule differ from the times above? (e.g going to sleep or waking up and hour earlier or later) *

- ◯ Never
- ◯ 1-2
- ◯ 3-4
- ◯ more than 4 times

How long before trying to sleep do you stop using your phone? *

- ◯ More than an hour before
- ◯ Less than an hour before
- ◯ Immediately before

If you wake up in during the night, how often do you use a smartphone or other device? *

- ◯ Always
- ◯ Often
- ◯ Rarely
- ◯ Never

For the study, what time would you like to try and go to sleep by? (Use the same time as you will set as the app's sleep time) *

Time

__ : __

For the study, what time would you like to try and wake up by? (Use the same time as you will set as the app's wake up time) *

Time

__ : __

Please provide your participant ID (Should have been sent to you) (This will be used to I can link the post-experiment survey you'll be doing) *

Your answer

Post-experiment sleep Survey

Post-experiment sleep survey

Thank you for using fish and sleep and helping with my dissertation! You should've used fish and sleep for at least a day including the fishing minigame in the morning, this survey will just ask you some questions about your experience using it and whether it helped your sleep patterns at all.

Please be aware that your response MAY be used in the final write up of my dissertation, data will be anonymised.

Sleeping Patterns

Question's about your sleeping habits (If you were part of the experiment you would've seen these questions before, please fill them out again with nay changes from the past week)

What time do you normally start going to sleep by on a weekday? (As in getting into bed and closing your eyes) *Use 24 hour time.* *

Time

23 : 00

What time do you normally wake up in the morning on a weekday? *Use 24 hour time.* *

Time

:

How long does it normally take you to get out of bed after you've woken up? *

- ◯ I get out of bed immediately
- ◯ Up to 15 minutes
- ◯ Up to 30 minutes
- ◯ Up to an hour
- ◯ more than an hour

How often during a week, including weekends, does your sleep schedule differ from the times above? (e.g going to sleep or waking up and hour earlier or later) *

- ◯ Never
- ◯ 1-2
- ◯ 3-4
- ◯ more than 4 times

How long before trying to sleep do you stop using your phone? *

- ◯ More than an hour before
- ◯ Less than an hour before
- ◯ Immediately before

If you wake up in during the night, how often do you use a smartphone or other device? *

- ◯ Always
- ◯ Often
- ◯ Rarely
- ◯ Never

Please provide your participant ID (IF you are not part of the formal experiment then please skip this question)

Your answer

Fish and Sleep Review

Question's about your experience using fish and sleep

What time did you set as your wakeup time for Fish and Sleep? *Use 24 Hour time.* *

Time

__ : __

On a scale of 1-5, how effective was Fish and Sleep at encouraging you to wake up by this time? *

	1	2	3	4	5	
Not very helpful	◯	◯	◯	◯	◯	Very helpful

What time did you set as your sleep time for Fish and Sleep? *Use 24 Hour time.* *

Time

__ : __

On a scale of 1-5, how effective was Fish and Sleep at encouraging you not to use your phone past this time? *

	1	2	3	4	5	
Not very helpful	◯	◯	◯	◯	◯	Very helpful

On a scale of 1-5, how confusing was Fish and Sleep to use?

	1	2	3	4	5	
Not very confusing	○	○	○	○	○	Very confusing

Was there anything you think the app should have included to improve your experience using it?

Your answer

Please list any features you found enjoyable in Fish and Sleep. (Write as much, or as little as you want)

Your answer

Please list any features you disliked in Fish and Sleep. (Write as much, or as little as you want)

Your answer

Did you encounter any bugs or crashes using Fish and Sleep? If so please describe them

Your answer

Experiment Results

			Fishing Game not active					Fishing Game active				
Participant 001				Day 1	Day 2	Day 3	Average	Day 1	Day 2	Day 3		Average
Sleep Time	23:30:00		Sleep Time	02:29:00			02:29:00					
Wakeup Time	09:00:00		Sleep Diff	02:59:00			02:59:00					
			Wakeup Time	09:22:00			09:22:00					
			Wakeup Diff	00:22:00			00:22:00					
			Percent Asleep	78%			78.00%					
Participant 002												
Sleep Time	04:00:00		Sleep Time	03:17:00	03:44:00	04:32:00	03:51:00	03:41:00				03:41:00
Wakeup Time	12:00:00		Sleep Diff	00:43:00	00:16:00	00:32:00	00:30:20	00:19:00				00:19:00
			Wakeup Time	12:58:00	14:36:00	24:58:00	17:30:40	48:46:00				48:46:00
			Wakeup Diff	00:58:00	02:36:00	24:26:00	09:20:00	36:46:00				36:46:00
			Percent Asleep	95%	93%	89%	92.33%	100%				100%
Participant 005												
Sleep Time	01:00:00		Sleep Time	01:06:00			01:06:00	00:04:00	00:31:00	01:13:00		00:36:00
Wakeup Time	09:30:00		Sleep Diff	00:06:00			00:06:00	00:56:00	00:29:00	00:13:00		00:32:40
			Wakeup Time	09:30:00			09:30:00	9:31:00	9:22:00	9:17:00		09:23:20
			Wakeup Diff	00:00:00			00:00:00	00:01:00	00:08:00	00:13:00		00:07:20
			Percent Asleep	98%			98.00%	96%	98%	96%		96.67%
Totals	Sleep Diff						01:11:47					00:25:50
	Wakeup Diff						03:14:00					18:26:40
	Percent Asleep						89.44%					98.33%
Sleep with game compared to without			00:45:57									
Wake up with game compared to without			-15.12									
ercent asleep with game compared to withou			-8.89%									

Survey Results

Pre-experiment Results

Sleeping Patterns

What time do you normally start going to sleep by on a weekday? (As in getting into bed and closing your eyes) *Use 24 hour time.*

3 responses

01 : __ | 01:00 2

04 : __ | 04:00

What time do you normally wake up in the morning on a weekday? *Use 24 hour time.*

3 responses

09 : __ | 09:30

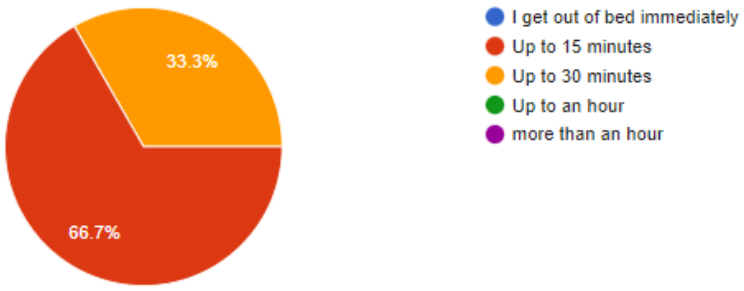
10 : __ | 10:35

12 : __ | 12:00

How long does it normally take you to get out of bed after you've woken up?

Copy

3 responses



How often during a week, including weekends, does your sleep schedule differ from the times above? (e.g going to sleep or waking up and hour earlier or later)

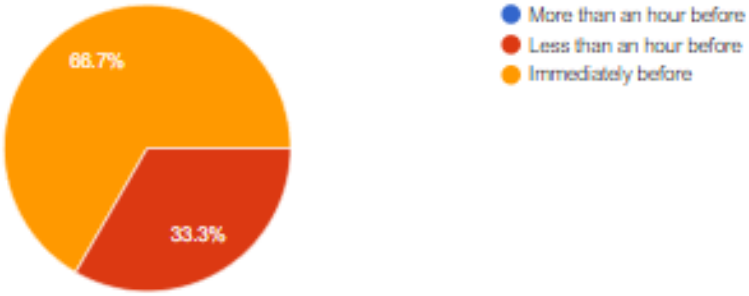
Copy

3 responses

How long before trying to sleep do you stop using your phone?

Copy

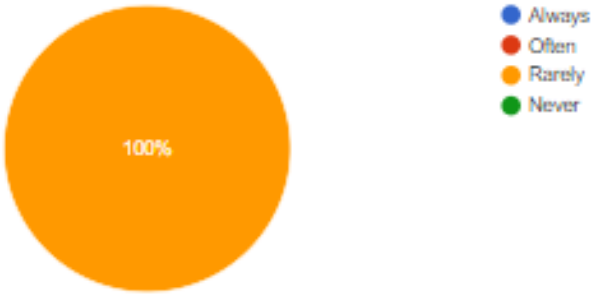
3 responses



If you wake up in during the night, how often do you use a smartphone or other device?

Copy

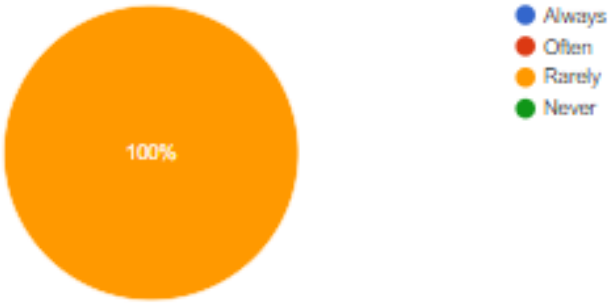
3 responses



If you wake up in during the night, how often do you use a smartphone or other device?

Copy

3 responses



For the study, what time would you like to try and go to sleep by? (Use the same time as you will set as the app's sleep time)

3 responses

01 : __ | 01:00 01:15

04 : __ | 04:00

For the study, what time would you like to try and wake up by? (Use the same time as you will set as the app's wake up time)

3 responses

00 : __ | 00:00

06 : __ | 06:00

09 : __ | 09:30

Please provide your participant ID (Should have been sent to you) (This will be used to I can link the post-experiment survey you'll be doing)

3 responses

002

001

005

Post-experiment Results

Sleeping Patterns

What time do you normally start going to sleep by on a weekday? (As in getting into bed and closing your eyes) *Use 24 hour time.*

4 responses

01 : __	01:00
05 : __	05:00
23 : __	23:30 2

What time do you normally wake up in the morning on a weekday? *Use 24 hour time.*

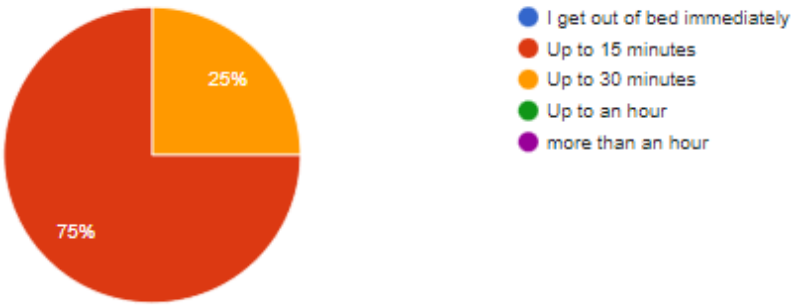
4 responses

08 : __	08:20
09 : __	09:30
10 : __	10:30
12 : __	12:30

How long does it normally take you to get out of bed after you've woken up?

Copy

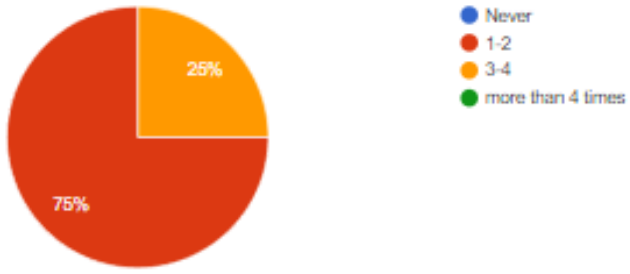
4 responses



How often during a week, including weekends, does your sleep schedule differ from the times above? (e.g going to sleep or waking up and hour earlier or later)

Copy

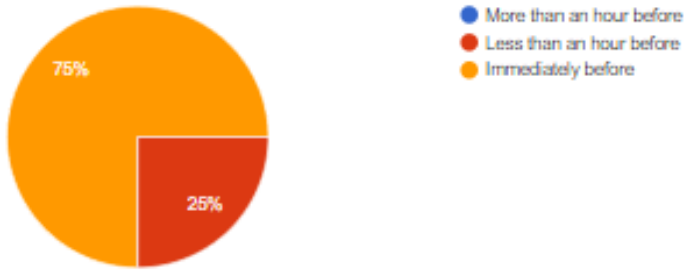
4 responses



How long before trying to sleep do you stop using your phone?

Copy

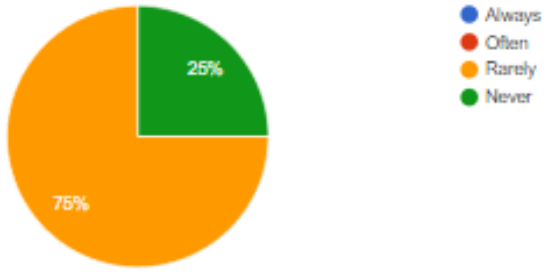
4 responses



If you wake up in during the night, how often do you use a smartphone or other device?

Copy

4 responses



Please provide your participant ID (IF you are not part of the formal experiment then please skip this question)

3 responses

5

001

002

Fish and Sleep Review

What time did you set as your wakeup time for Fish and Sleep? *Use 24 Hour time.*

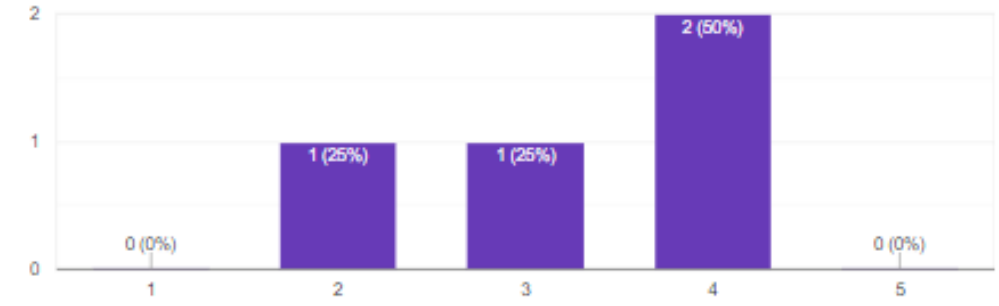
4 responses

08 : __	08:00 08:20
09 : __	09:30
12 : __	12:00

On a scale of 1-5, how effective was Fish and Sleep at encouraging you to wake up by this time?

Copy

4 responses



What time did you set as your sleep time for Fish and Sleep? *Use 24 Hour time.*

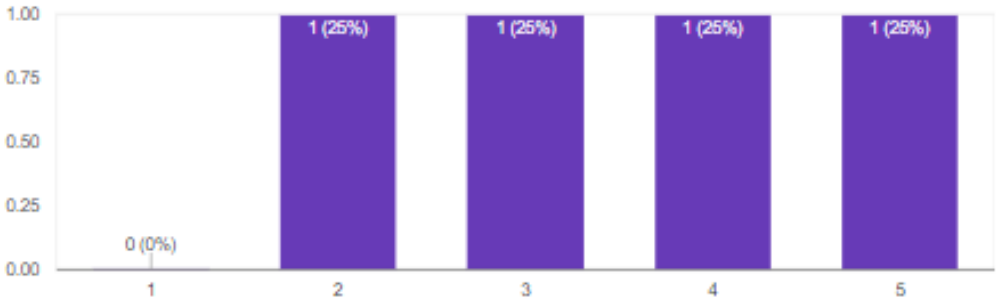
4 responses

01 : __	01:00
04 : __	04:00
11 : __	11:00
23 : __	23:40

On a scale of 1-5, how effective was Fish and Sleep at encouraging you not to use your phone past this time?

Copy

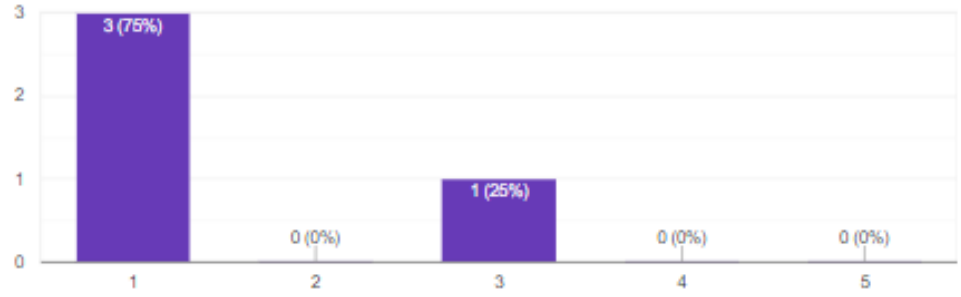
4 responses



On a scale of 1-5, how confusing was Fish and Sleep to use?

Copy

4 responses



Was there anything you think the app should have included to improve your experience using it?

3 responses

Would be nice if there was a way to tell when a fish has been fed as I wasted a lot of food accidentally feeding the wrong ones. Pls let me name them/see stats after they are caught. Also would be good to be able to set a weekend time that's a bit later perhaps? Could be good to

No

Push notifications could have potentially encouraged me to use it more.

Please list any features you found enjoyable in Fish and Sleep. (Write as much, or as little as you want)

4 responses

I liked da fishies 🐟 and how it deters me from picking up my phone after I've pressed sleep. Also like the stats that tell you how many times you open it. Maybe would be good to see the specific time when you opened it somehow in this section. Like how definitive the period of time you can catch fish is as normally I would hit snooze on my alarm and sleep for another 10 mins but actually woke up this time.

The mini game was a lot of fun, nice way to start my day. Encouraged me to be precise on what time I woke up.

Fishing was fun

I liked the idea of purchasing upgrades with earned gold.

Please list any features you disliked in Fish and Sleep. (Write as much, or as little as you want)

4 responses

Disliked how repetitive the fishing mechanic was. Also how having the 10 min timeslot encourages me to stay in bed playing it for the whole ten minutes? Was oddly harder to get up as it hooked me to my phone first thing in the morning. Maybe limit the no. of fish catchable per day but still have the 10 min timeslot in which to do it?

None

Not being able to fish when I woke up before/after my alloted time (I might just be misinterpreting the Goal of the experiment)

I'm aware it was an optional permission but i was not too keen on the app setting my brightness, always felt too dark or too bright. I also never actually got the fish to spawn because i fell off on the third day, always waking up 30ish mins after my wakeup time, perhaps the fish could hang around a bit longer than 15mins after wakeup time?

Did you encounter any bugs or crashes using Fish and Sleep? If so please describe them

4 responses

No crashes or anything buggy. Had some small issues with bounding/fish in the aquarium but could just wait for them to move back onto screen.

No

Ui not being inline with my phone

No the app functioned as intended.