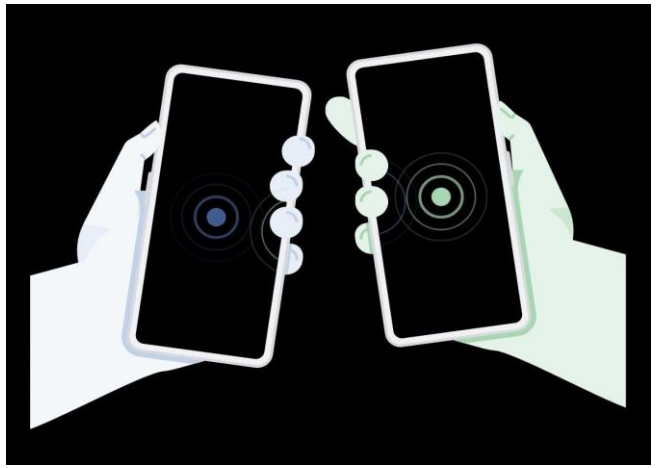

Phish Phingers
Project by Saph, Callum and Luca

Project Proposal & Timeline

Project Summary - Making Credit Card Theft Fun (and not illegal)

Phish Phingers is a long-term social challenge game for a group of friends to try and steal each others credit card information.

Players link their devices, one player is the 'Host' and target of all the 'Hunters'. For the duration of play (set by the players) the Hunters must sneak access to the Host's phone and 'Catch' them while the Host must stay safe. Although an app powers the experience, the core gameplay is intentionally non-screen-based (pervasive media), reinforcing the idea that digital threats often operate in plain sight.



Aims and Objectives

- To gamify NFC skimming as a method of promoting awareness of digital vulnerabilities.
- To create a mobile application that prioritises real-world social interaction.
- To simulate real-life cybersecurity risks through playful, immersive design.
- To challenge players to consider how data breaches and social engineering can occur in everyday settings.

Reasonings

- Technically Interesting
- Entertaining
- Informative of Security
- Learning CySec concepts and product design

Educational Intent

- The game is intended not just for fun but as a subtle educational tool. Players become more aware of how easily data breaches can happen when they’re least expecting it. It turns friends into cyber threats and, hopefully, will demonstrate that social engineering and physical access can be key vulnerabilities.

Future Development

- While the current version runs through phones, we've also brainstormed a future iteration using physical NFC tokens and Raspberry Pi stations in a centralised public space which is ideal for exhibitions or installations.

Date	Event
15 Feb 2025	Initial idea meeting. Discussed shared cybersecurity interests and early concept involving physical NFC chips.
28 Feb 2025	Luca shares insights from Southampton CyberSoc; team pivots toward using smartphones for realism. Roles assigned.
8 Mar 2025	Concept finalised: social NFC game using smartphones. Luca begins branding. Saph chooses Kotlin; Callum storyboards animation ideas.
21 Mar 2025	Early prototype shown. Callum suggests GIF animation. Luca shares logo drafts. Cybersecurity research continues.
5 Apr 2025	NFC detection logic built. App animation previewed. Branding materials like social media mockups designed.
17 Apr 2025	Internal playtest. Host alert feature added. App loop refined. Marketing copy and ethical framing discussed.
2 May 2025	Final bug fixes. Full app flow reviewed. Callum begins video storyboard. Physical token ideas discussed for future version.
13 May 2025	Final rehearsals and documentation submitted. Group reflects on project evolution and potential post-release plans.


Group Meetings

Meeting 1: 15th February 2025
Attendees: Luca, Saph, Callum

- Summary: We gathered for our first meeting to brainstorm ideas for our group project. All three of us have a shared interest in cybersecurity and deceptive design, which sparked the initial conversation around awareness games. Luca proposed the idea of gamifying credit card skimming using NFC technology. We explored early concepts involving hidden NFC chips and physical skimming. At this point, the idea was very much focused on physical objects as props. We agreed to individually research similar projects and real-life NFC skimming case studies for our next session.

Meeting 2: 28th February 2025
Attendees: Luca, Saph, Callum

- Summary: Luca reported back on a discussion he had with a member of the University of Southampton’s CyberSecurity society, where he was shown firsthand how surprisingly easy NFC skimming can be with modern smartphones. This demonstration led us to question the physical chip concept and consider a more realistic scenario involving phones. We discussed the possibility of building a mobile app where players interact primarily through devices they already own. The team was enthusiastic about pivoting the concept in this direction. We also outlined roles more clearly: Luca as project manager, researcher and brand designer, Saph as the lead programmer and Callum as visual artist and animator.

Meeting 3: 8th March 2025
Attendees: Luca, Saph, Callum

- Summary: We officially locked in the Phish Phingers concept - a long-term social challenge game where friends attempt to ‘skim’ each other's devices over a set period. We fleshed out the core mechanics, including roles (Host and Hunters), scoring systems, and the rule that the Host must remain ‘uncaught’ for as long as possible. Saph suggested building the app using Kotlin in Android Studio for efficient NFC implementation. Luca began sketching and researching early branding/logo ideas and concepts, drawing on academic sources regarding branding and researching design aesthetics of other party games. Callum began prototyping ideas for the app’s animation and visual tone.

Meeting 4: 21st March 2025
Attendees: Luca, Saph, Callum

- Summary: Saph shared early screenshots from Kotlin development, including the first working prototype of a Hunter-to-Host tag. We tested initial NFC interaction and discussed how to simulate “catching” someone in a fun way. Callum pitched the idea of incorporating an animated GIF background that conveys tension and subtle humour, referencing little fishes (playing on the word “phish”). Luca presented draft logos and typography styles. We reviewed several cybersecurity articles and case studies together to ensure the game remains grounded in real concerns.

Meeting 5: 5th April 2025
Attendees: Luca, Saph, Callum

Summary: Development was progressing well. Saph demonstrated NFC interactions that could differentiate between the Host and Hunter devices. We started discussing the onboarding experience and how to make the rules of the game clear. Callum showed an early version of the app animation - a little underwater scene of fishes swimming around GIF that subtly evokes digital paranoia as the fishes are swimming in open water but also just builds on the aquatic branding we were going for. Luca finalised the logo and began work on other brand materials like social media mockups and presentation templates.

Meeting 6: 17th April 2025

Attendees: Luca, Saph, Callum

Summary: We playtested the app internally and began refining the game loop. Saph introduced a new feature where the Host's phone emits a faint alert vibration if the app detects a skim attempt — adding more tension. Luca worked on writing marketing copy and explanatory visuals. Callum added final touches to the app background, adjusting loop timing to prevent lag. We discussed ethical implications of gamifying digital theft, and how we could frame it clearly as an awareness tool.

Meeting 7: 2nd May 2025
Attendees: Luca, Saph, Callum

- Summary: With the app approaching completion, we started building our documentation and presentation. We created a detailed breakdown of roles, game mechanics, and user journey. Luca polished the branding and edited pitch visuals. Saph shared the full app flow and addressed final bugs in the NFC tagging sequence. Callum helped storyboard our final demo video and prepared the animation assets. We also considered future possibilities, such as developing a hub-based version using physical NFC chips and Raspberry Pi stations.

Meeting 8: 13th May 2025

Attendees: Luca, Saph, Callum

Summary: Final checks and rehearsals for our presentation. We reflected on how far the concept had evolved from a physical prop game to a mobile-based social experience rooted in real cybersecurity risks. The project showcases a unique intersection of playful design, education, and public awareness. Luca wrapped up documentation and submitted final assets. We agreed to keep iterating on the concept after submission, potentially releasing a public beta in the summer.

Similar Projects & Research

Similar Project Research: KC7 & SherLOCKED (1)

The development of *Phish Phingers* was informed by existing cybersecurity education games, particularly *KC7* and *SherLOCKED*, both of which exemplify how play can enhance awareness of complex digital threats. *KC7* positions the player as a cybersecurity analyst investigating real-world data breaches, encouraging engagement through narrative-driven roleplay and self-directed inquiry (Katzcy, 2023). This detective-style framework served to inspire *Phish Phingers*, which similarly assigns active roles (Host and Hunters) to simulate real-world skimming threats. The accessibility of *KC7* and its use of browser-based interactions inspired our decision to ensure *Phish Phingers* functions smoothly on everyday mobile devices, minimizing barriers to participation and reinforcing the idea that cybersecurity awareness should be embedded in familiar technology use.

Similar Project Research: KC7 & SherLOCKED (2)

Similarly, *SherLOCKED* (Brown et al., 2021) has proven effective in reinforcing academic cybersecurity concepts through puzzle mechanics and gamified learning. Its successful integration into undergraduate courses validated our belief that interactive games can meaningfully support digital literacy. While *SherLOCKED* focuses more on abstract principles like the CIA triad, it still emphasized the power of experiential learning which is something we sought to replicate through *Phish Phingers*' real-time, long-term social play. Both sources encouraged us to lean into educational game design that avoids being overly didactic, instead trusting players to internalize risks and strategies through carefully structured gameplay. These projects underscored the potential for *Phish Phingers* to not only entertain but actively change user behaviour around device safety and awareness of NFC vulnerabilities.

References

References

- Brown, D., Murphy, T., & Beck, J. (2021). *SherLOCKED: A Detective-Themed Serious Game for Cybersecurity Education*. arXiv. Available at: https://arxiv.org/abs/2107.04506
- Katzcy (2023). *KC7 Cyber Range*. KC7cyber.com. Available at: https://kc7cyber.com

Inspiration: BLEWhisperer

The *BLEWhisperer* paper by Gangwal et al. (2022) presents a compelling case for how wireless communication protocols like Bluetooth Low Energy (BLE) can be subverted for covert data transmission. The researchers show that it's possible to exfiltrate data via BLE advertisements without pairing, essentially, without the user ever knowing. This felt like a parallel to NFC skimming, where the user might not notice anything out of the ordinary until it's too late.

While BLE and NFC are technically distinct, both rely on close-proximity wireless interactions and both have been shown to have real-world vulnerabilities. For us, *BLEWhisperer* validated our decision to simulate NFC attacks in a casual setting as it underscored the fact that these aren't far-fetched scenarios. They can happen, and they're subtle.

Importantly, the study also highlighted a tension we reflected on in our own development: convenience versus risk. Features like BLE and NFC are built for seamless interaction, but that seamlessness can be exploited. By turning this tension into a game, *Phish Phingers* aims to spark user reflection about how much trust we place in proximity-based technologies – often without thinking.

Reference:

- Gangwal, A., Singh, S., Spolaor, R. and Srivastava, A., 2022. BLEWhisperer: Exploiting BLE Advertisements for Data Exfiltration. *arXiv preprint arXiv:2204.08042*. Available at: https://arxiv.org/abs/2204.08042

Similar Project Research: Subterfuege

Subterfuge is a mobile strategy game where players form secret alliances, betray each other, and compete over several days for underwater dominance. While it doesn't directly engage with cybersecurity themes, the game's long-form social deception structure heavily influenced our thinking in *Phish Phingers*. Like *Subterfuge*, our game unfolds over a set time period, requiring players to remain alert and engaged with the ongoing social dynamics.

What struck us most was *Subterfuge*'s ability to make every interaction feel meaningful and fraught, even when nothing appeared to be happening on-screen. This reinforced our decision to keep *Phish Phingers* mostly non-screen-based (aside from the fact that was part of the assignment brief!) as the suspense and tension are meant to exist in the real world, not in the app. We wanted players to feel the pressure of surveillance, proximity and trust in their day-to-day lives, in the same way *Subterfuge* makes players second-guess their friends through chat and timing.

Another valuable takeaway was how *Subterfuge*'s UI maintained a clean, minimalist look despite the game's underlying complexity. It reminded us that players appreciate clarity, especially in games that lean heavily on player-driven strategy. That insight helped shape *Phish Phingers*' visual design as we kept the branding bold and readable so the game's core message wouldn't get lost in visual noise.

Reference:

- Snappy Touch & Gameblyr, 2015. *Subterfuge* [mobile game]. Available at: https://subterfuge-game.com

Inspiration: Flipper Zero (1)

During the early ideation stage of *Phish Phingers*, one device repeatedly surfaced in both our research and casual discussions: the Flipper Zero. Marketed as a “multi-tool for geeks,” Flipper Zero is a pocket-sized hardware hacking device that interacts with RFID, NFC, Bluetooth and other wireless protocols. What stood out to us was how it gamifies cybersecurity knowledge, from decoding access cards to exploring signal spoofing, in a way that feels playful, mischievous and oddly social and not just its technical versatility which was also something we found particularly interesting.

Inspiration: Flipper Zero (2)

Flipper Zero's presentation struck a balance between hacktivism and accessibility that we wanted to replicate with our own project as it takes powerful, often abstract concepts in digital security and wraps them in a much less intimidating Tamagotchi-like interface. This design choice (to turn learning into *play)* inspired our goal when developing *Phish Phingers* as we wanted to apply the same ethos to mobile NFC skimming. Rather than using physical access cards or dongles like Flipper Zero often does, we decided to lean into how smartphones now function as wallets, making our game more realistic for today's users. Although, our initial ideation for the project did involve physical cards and it is something we are looking into for future developments of the project.

The core mechanic of *Phish Phingers* (sneaking your phone near someone else's to simulate an NFC "skim") was inspired by watching videos of Flipper Zero users cloning RFID tags or reading hotel key cards with just a wave. Seeing how easy and invisible the process was helped shape our understanding of the stakes. We realised the average person has no idea how vulnerable they are in public spaces and much like Flipper Zero makes hacking approachable, we wanted *Phish Phingers* to make these vulnerabilities visible, not through fear, but through friendly deception and shared play.

Inspiration: Flipper Zero (3)

However, we also took caution from the ethical grey areas surrounding Flipper Zero as its popularity in online forums raised questions about responsible use, especially when devices intended for learning are repurposed for malicious pranks (Izquierdo, 2025). In designing *Phish Phingers*, we made clear efforts to position the experience as an awareness tool and a piece of fun, not a blueprint for actual exploitation. The point isn't to train people to skim, of course, it's to let them *feel* what it's like to be skimmed, so they can better understand and defend against it.

In summary, Flipper Zero didn't serve so much to inspire the technical inspiration behind *Phish Phingers,* more so, it shaped our philosophy: we wanted to make cybersecurity social, surprising and a little bit silly just like Flipper Zero.

References

- Flipper Devices Inc. (n.d.) *Flipper Zero - Portable Multi-tool for Geeks*. Available at: https://flipperzero.one/
- Izquierdo R. (2025) AS USA. *Terrifying new Flipper Zero device alarms cybersecurity experts: Could hackers unlock your car and steal your bank info?* Available at: https://en.as.com/latest_news/terrifying-new-flipper-zero-device-alarms-cybersecurity-experts-could-hackers-unlock-your-car-and-steal-your-bank-info-n/

Similar Project Research: Wormhole (1)

In developing *Phish Phingers*, our group drew significant inspiration from *Wormhole: A Perpetual Pin Game* by Stellar Factory. This game transforms a simple enamel pin into a continuous, real-world game of tag. Players spin the pin to receive a mission such as pinning it on clothing, tying it to an accessory or placing it somewhere noticeable and then discreetly pass it on to an unsuspecting target. The game is designed for infinite play, requiring no screens or digital interfaces.

This minimalist, physical approach to gameplay inspired our idea for *Phish Phingers* as we also aimed to create a game that operates subtly in the background of daily life, much like *Wormhole*. By leveraging the NFC capabilities of smartphones, *Phish Phingers* allows players to simulate the act of digital pickpocketing, fostering awareness of cybersecurity risks in a playful context.

Similar Project Research: Wormhole (2)

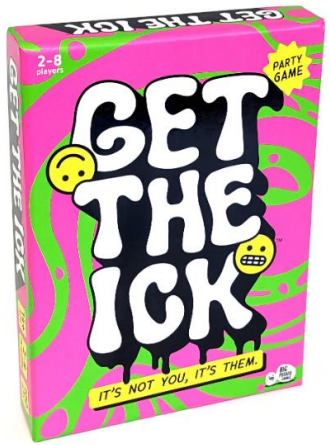
The social dynamics of *Wormhole* also influenced our design particularly the elements of surprise and stealth in the way that *Wormhole* is based around the thrill of secretly transferring the pin mirrors the covert interactions in *Phish Phingers*, where players must be vigilant and strategic. This emphasis on real-world interaction over digital engagement encourages players to be more mindful of their surroundings and the potential vulnerabilities in everyday technology use.

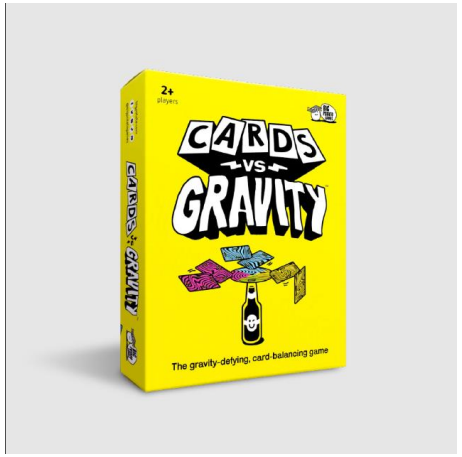
References

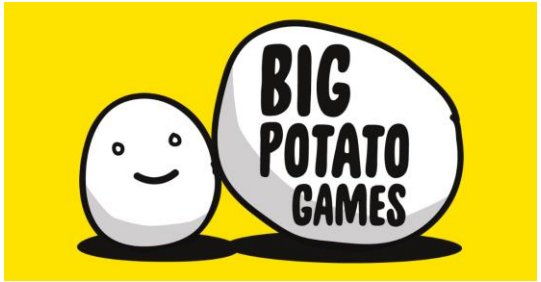
- Stellar Factory (n.d.) *Wormhole: A Perpetual Pin Game*. Available at: https://stellarfactory.com/products/wormhole

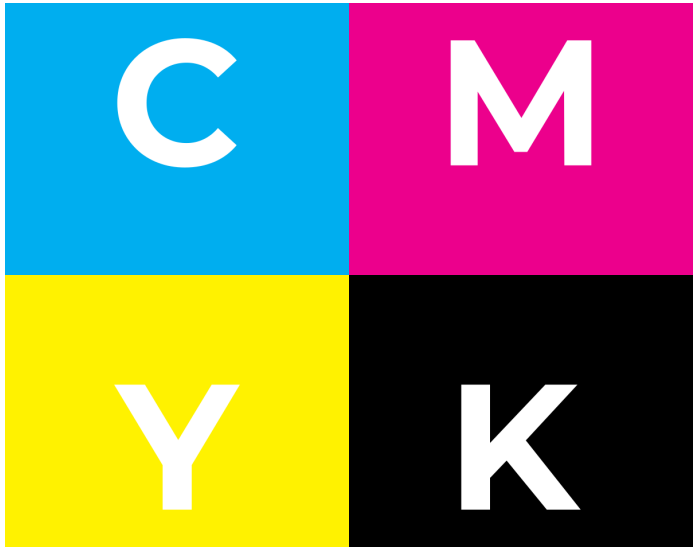
Logo/Branding

Branding Moodboard







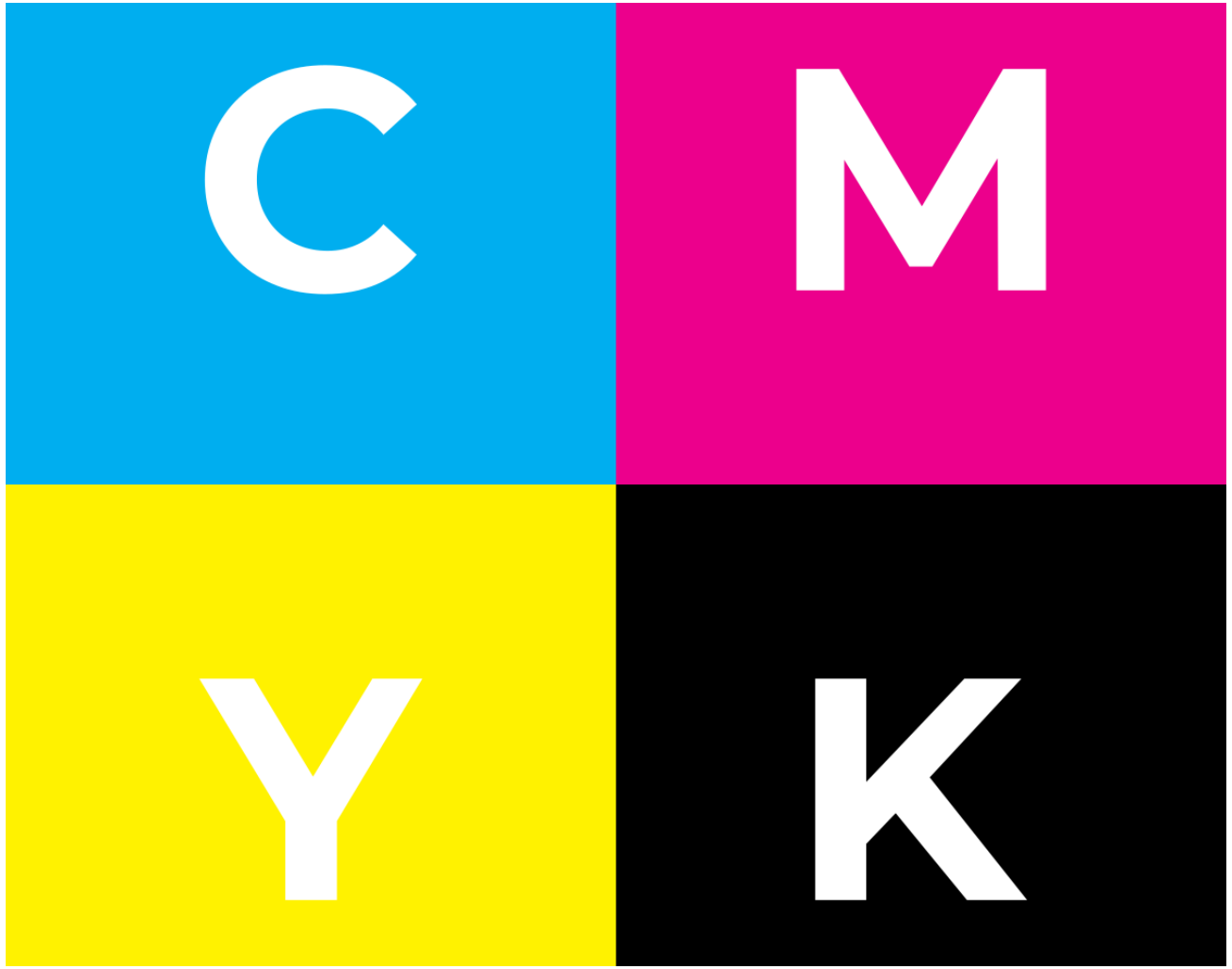


Inspirations

For the graphic design of our project, we drew inspiration from the bold, playful aesthetics of popular party games like *Get the Ick*, *Cards vs Gravity*, and *Colour Brain which* use bright colours, quirky typography, and eye-catching layouts to create a sense of excitement which is something we aimed to replicate in our own branding. Since our game explores themes like cyber security and credit card theft, it was important for us to strike a balance between informative content and an inviting, non-intimidating atmosphere. By channelling the visual language of these vibrant party games, we were able to craft a logo and visual identity that supports our goal of gamifying serious topics in a way that is fun, engaging, and approachable for a wide audience.

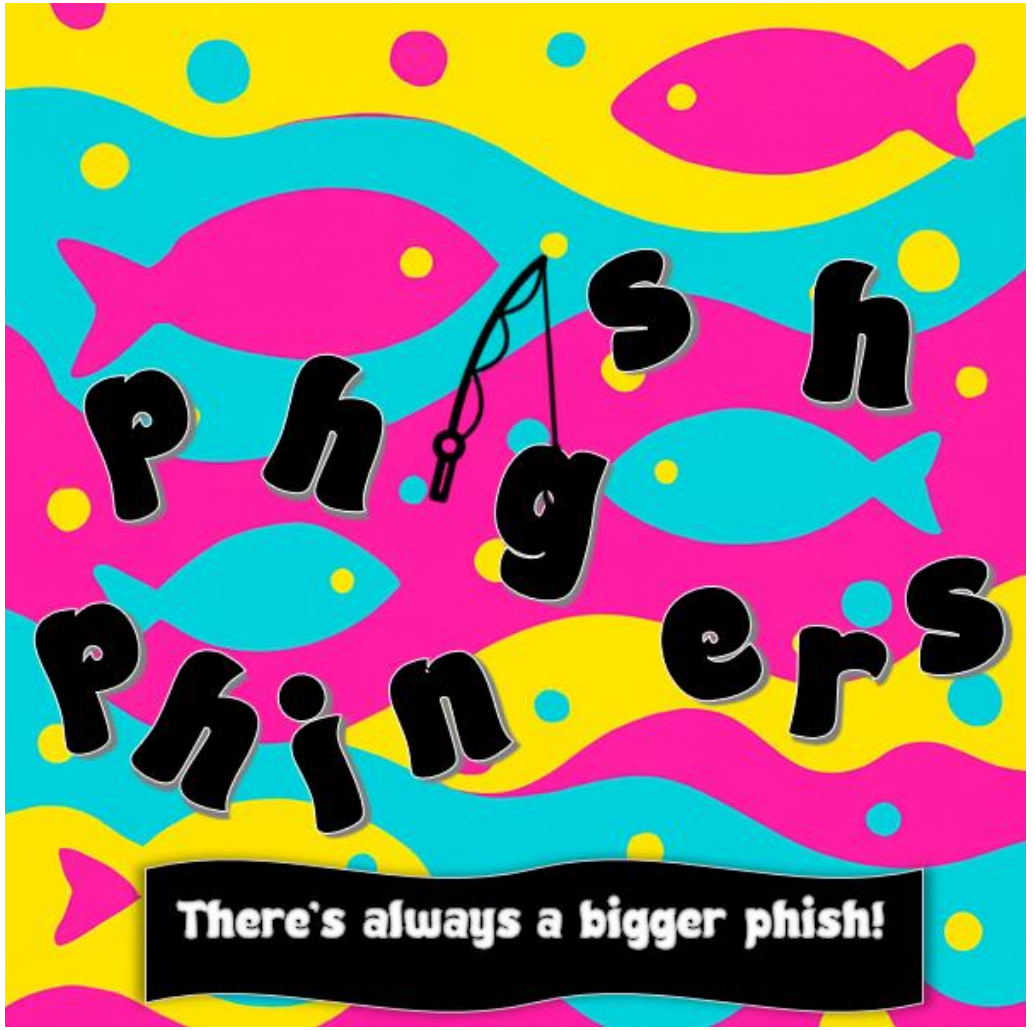
Colour Scheme

As such, we decided to employ a similar vibrant colour scheme to that used by the aforementioned branding inspirations. By utilising bright cyans, yellows, magenta with a contrast of black and white typography, our logo aims to successfully achieve the evocation of the sense of fun that we aim to elucidate in audiences. Our primary aim is to raise cyber security awareness and not make it seem intimidating.

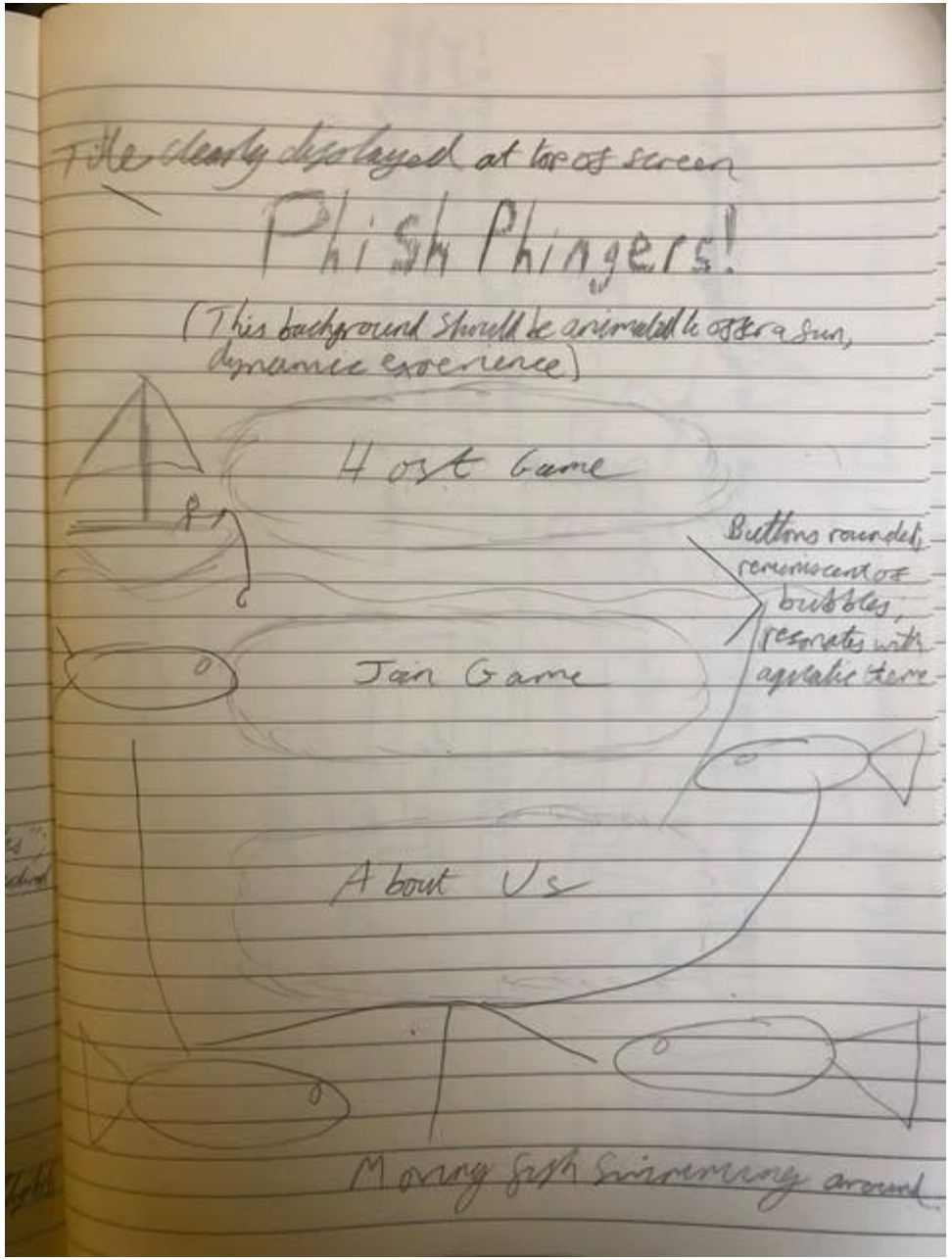


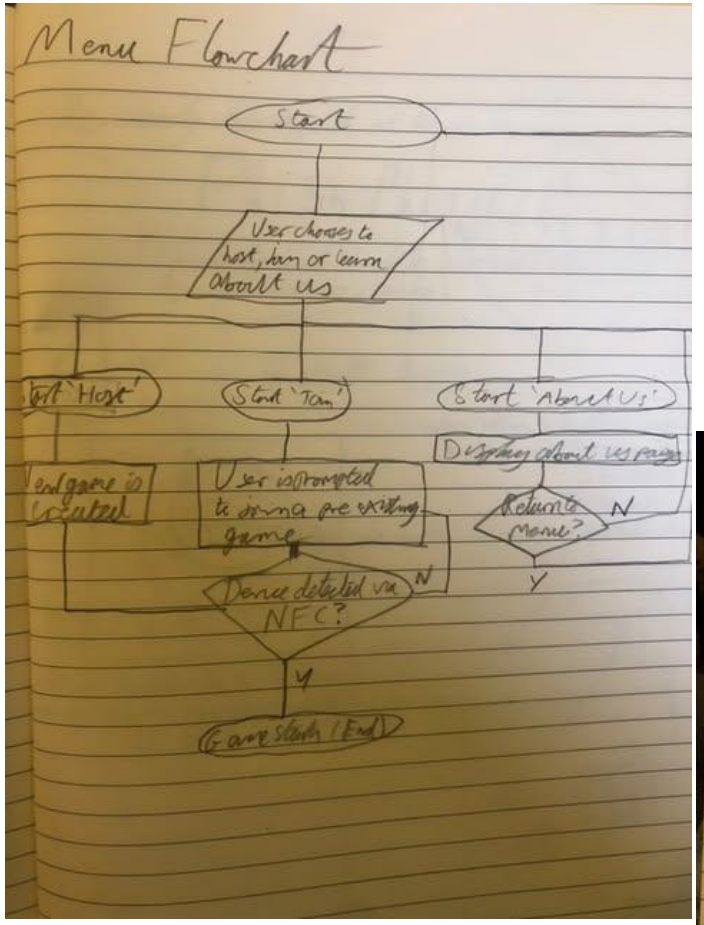
Our logo

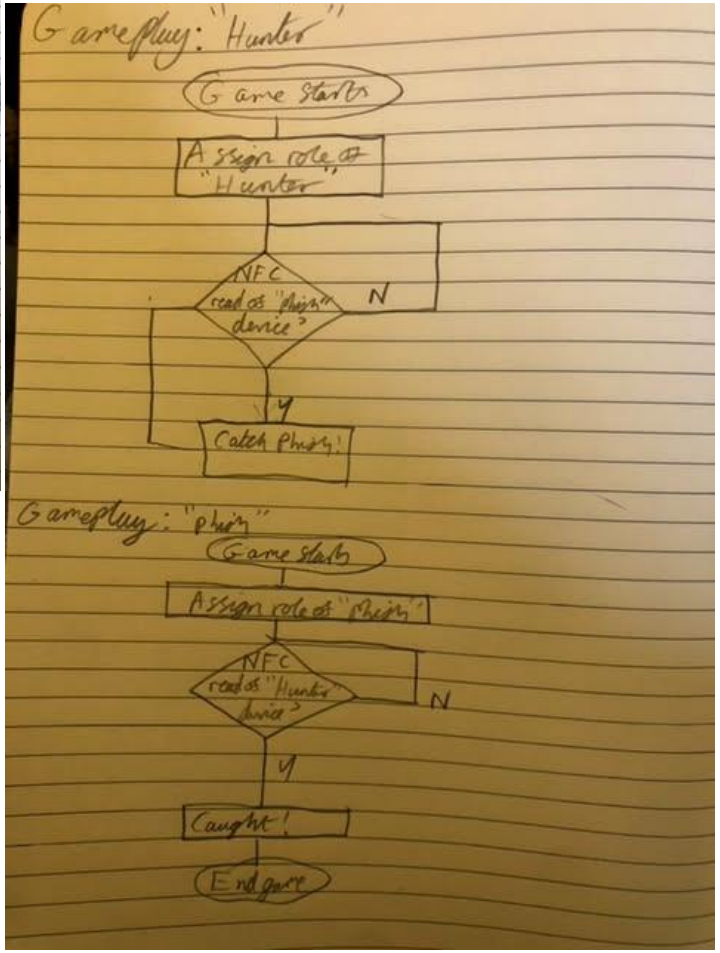
In the end, this was the design we ended up creating. We came up with a fun name that bridges the ludic nature of our game and our aim to raise cyber security awareness via a pun – making use of the term “phish” associated with scams. As such, our branding includes fish and water imagery which, in turn, almost makes the word “phish” seem less threatening by underscoring its association with the vibrantly coloured, somewhat cute looking fish in our branding.



Flowcharts


Title clearly displayed at top of screen
Phish Phingers!
(This background should be animated to offer a fun, dynamic experience)
Host Game
Join Game
About Us
Buttons rounded, reminiscent of bubbles, responds with agitable tone
Moving fish swimming around


Menu Flowchart
Start
User chooses to host, join or learn about us
Start 'Host'
Start 'Join'
Start 'About Us'
End game is created
User is prompted to join a pre existing game
Displays about us page
Returns to menu?
N
Y
Device detected via NFC?
N
Y
Game starts (End)


Gameplay: "Hunter"
Game starts
Assign role of "Hunter"
NFC read as "phish" device"
N
Y
Catch Phish!
Gameplay: "phish"
Game starts
Assign role of "Phish"
NFC read as "Hunter" device"
N
Y
Caught!
End game

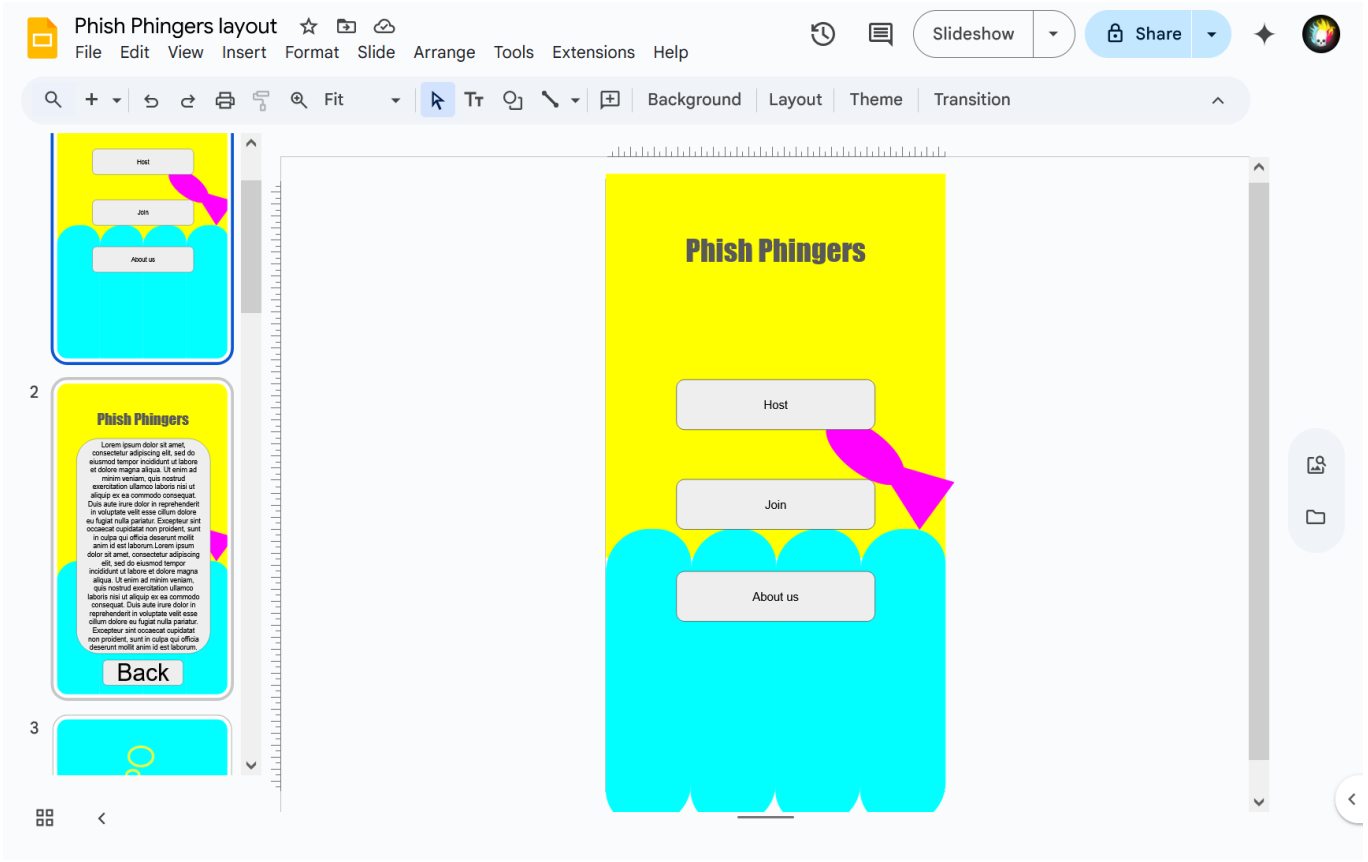

Design/Layout

Prototyping Layout

To establish a solid foundation for our app's design, we began with a simple PowerPoint prototype. This allowed us to quickly click through screens and test the overall user flow of the app.

We identified the key pages needed for our initial prototype:

- **Main Menu**
- **Hosting**
- **Game Mode selection**
- **Joining**
- **About Us**
- **In-Play**



Prototyping Design

Our game features two opposing teams: the Hunters and the Hunted. To reflect this dynamic, we incorporated subtle visual cues into the design:

- Joining a game takes the player *beneath the sea*, visually transforming them into the hunted — a fish.
- Hosting a game lifts the player *into the clouds*, symbolizing a hunter's perspective looking down from above.

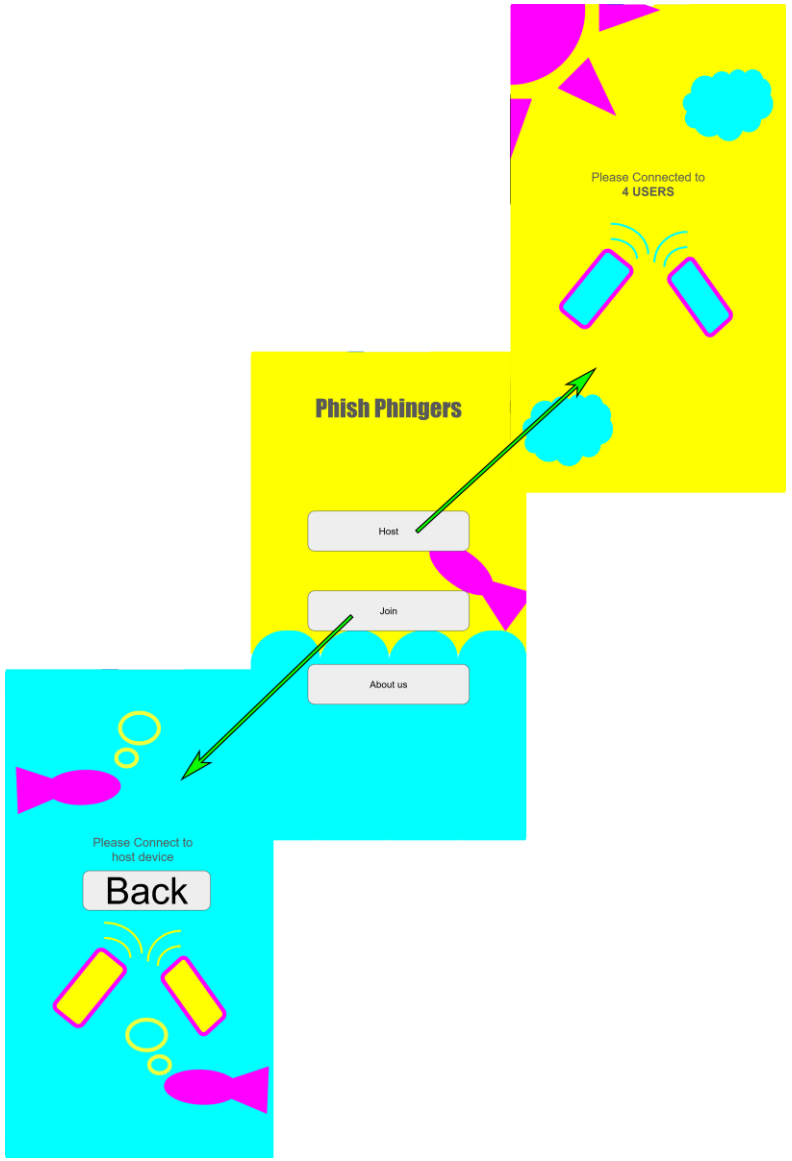
These transitions add a subtle layer of immersion, helping players feel more connected to their role in the game.

We also found that static screens can give the impression that the app has frozen. To keep the experience feeling alive and responsive:

- Bubbles were added to underwater scenes
- Moving clouds animate the sky

These ambient animations help maintain user engagement and prevent confusion or frustration.

[The prototype]



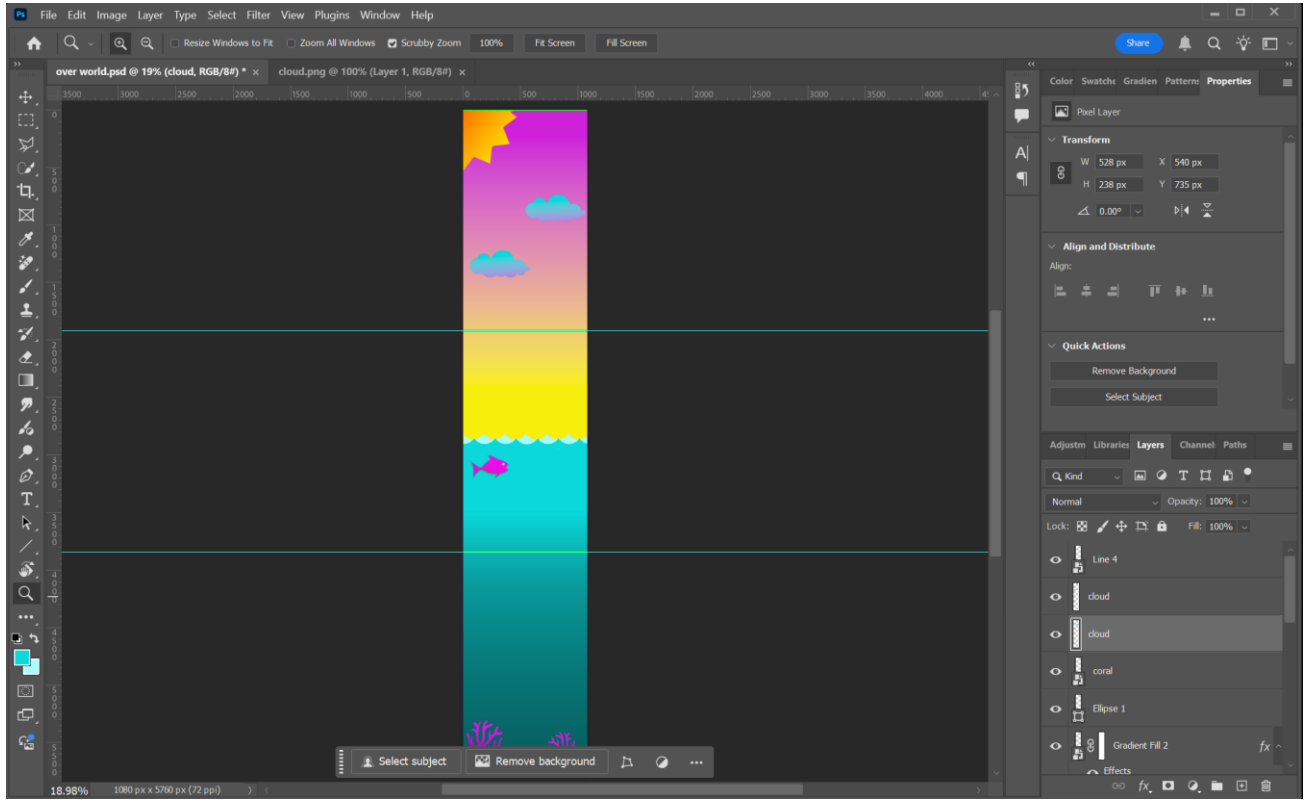
Assets

Once we finalized the design concept, I moved into Photoshop to create all the visual assets needed for the app's background.

To accommodate scrolling in the app, I started by using a standard phone resolution and tripled the height, allowing users to scroll up and down for a more dynamic experience.

The essential assets I designed included:

- **Background**
- **Waves**
- **Fish**
- **Coral**
- **Clouds**
- **Sun**
- **Bubbles**

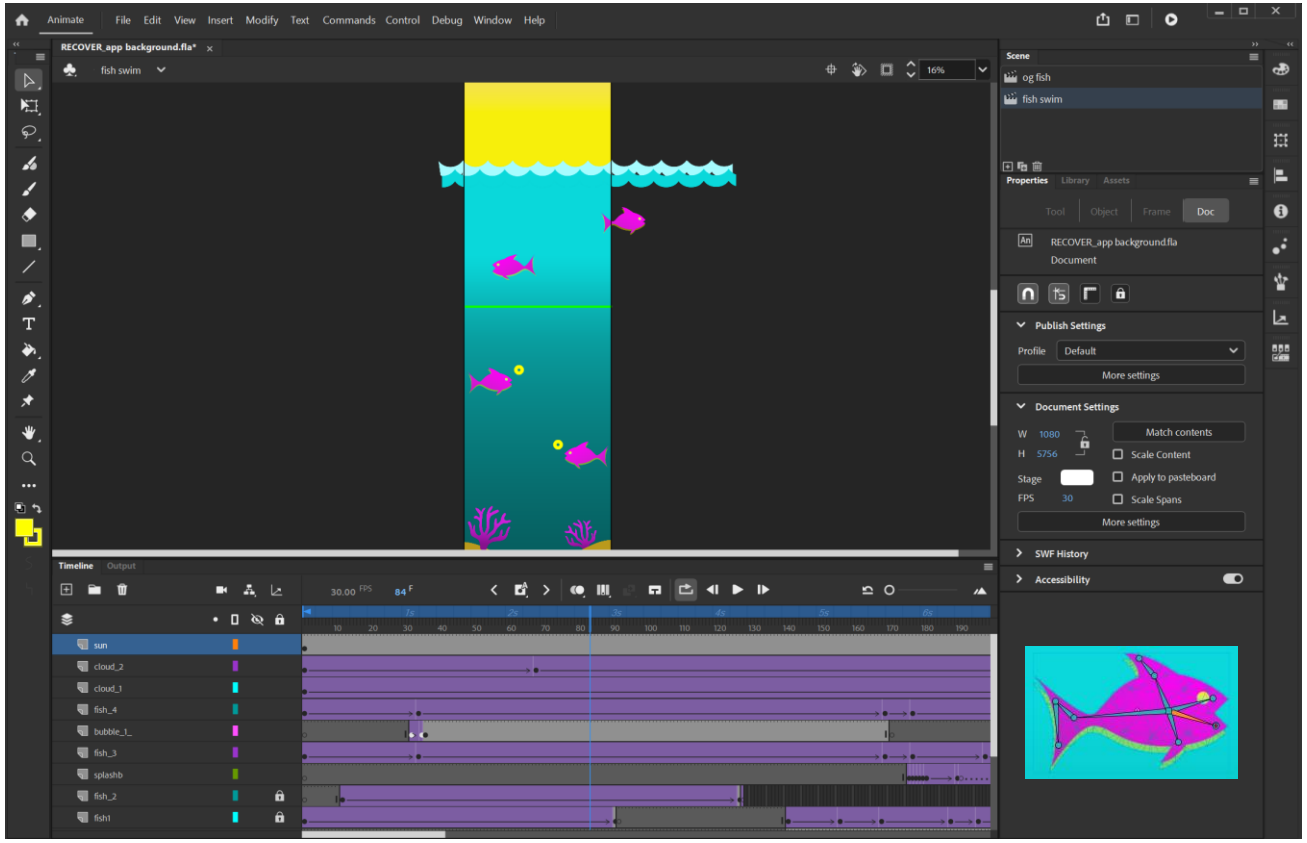


Animation

After designing the assets, I moved into **Adobe Animate** to bring them to life. This involved:

- Converting each asset into **symbols**, allowing for efficient animation and reuse.
- Adding **bones and rigs** to certain elements like fins and mouths to create natural, fluid movements.

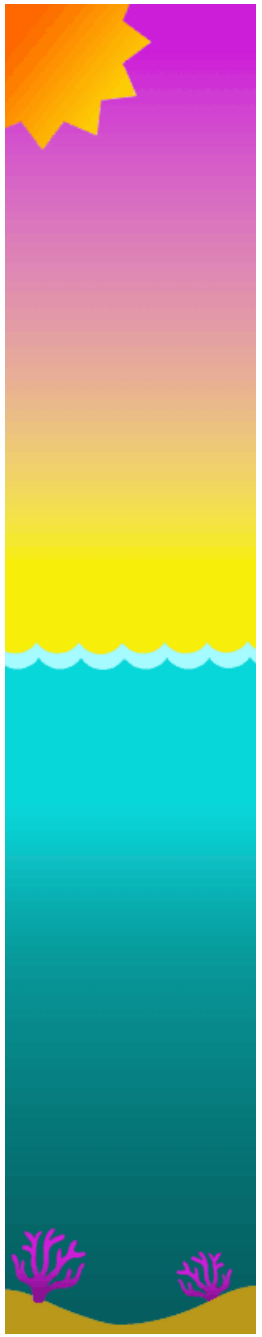
These animations added character to the app and helped reinforce a sense of liveliness and immersion throughout the user experience.



Final Animation

Once all the elements were combined — the design, assets, and animations — we created the final animated scene for the app.

This brings the user interface to life, blending functionality with visual storytelling to create a more immersive experience.

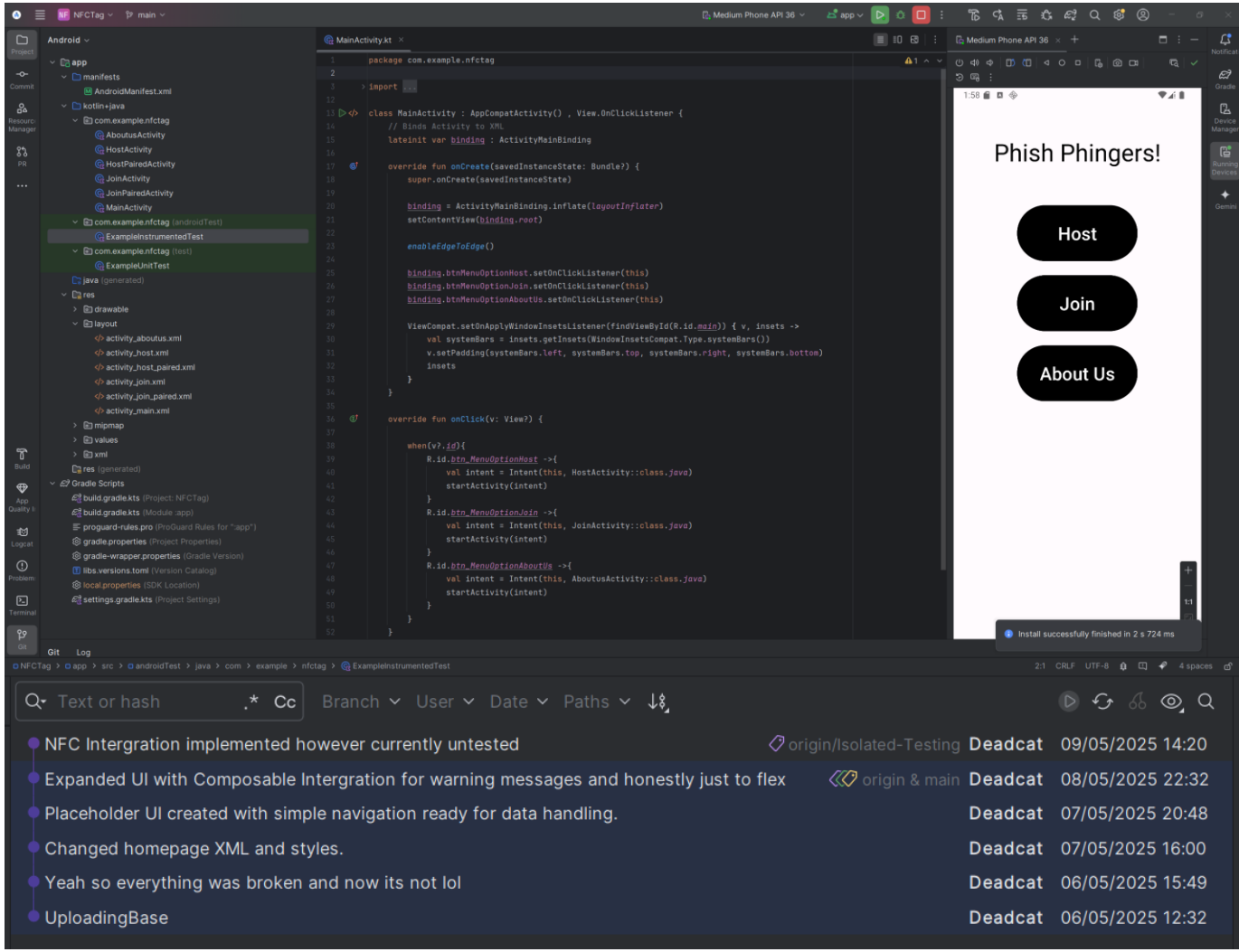


Technical

Application + UI

The App is written in Kotlin (agony I hate this language) using Android Studio. Version control handled through Git.

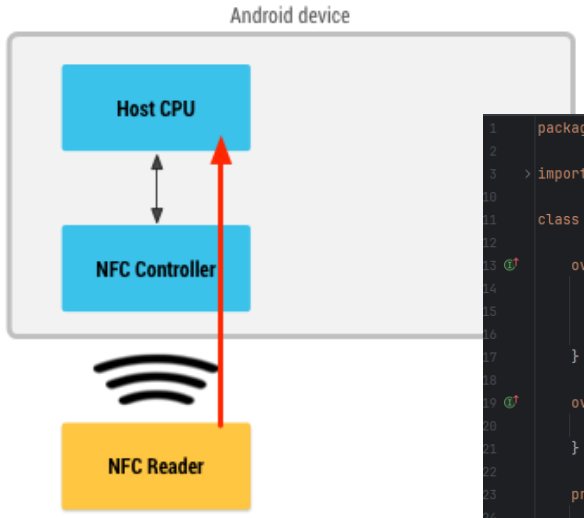
The base UI uses a Views format with a ViewModel Binding. Retrospectively this could have been a MVVM based application however I was still learning the language and chose to prioritise the extra technical requirements over upgrading the entire project.

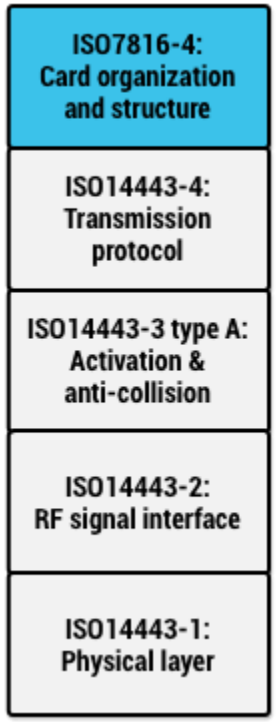


NFC Card Emulation

NFC Can be handled in multiple ways however for our needs where the target device will not have the application open we require Host-based Card Emulation (HCE) where the hosting device is treated as an NFC tag rather than an Read/Write mode which is used to interact with static tags*. HCE systems only use the ISO NFC Protocol

HCE systems require an Application-ID (AID) which routes the NFC signal to the desired application, this is how Contactless payment machines can open wallet apps.






```kotlin
package com.example.nfctag

import ...

class MyHostApduService : HostApduService() {

    override fun processCommandApdu(commandApdu: ByteArray, extras: Bundle?): ByteArray {
        triggerNotification("NFC Scan Detected", "You were scanned by a device")
        val deviceId = DeviceIdManager.getOrCreateDeviceId(this)
        return deviceId.toByteArray(Charsets.UTF_8)
    }

    override fun onDeactivated(reason: Int) {
        TODO("Not yet implemented")
    }

    private fun triggerNotification(title: String, message: String) {
        val channelId = "nfc_detected_channel"
        val manager = getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val channel = NotificationChannel(channelId, "NFC Events", NotificationManager.IMPORTANCE_HIGH)
            manager.createNotificationChannel(channel)
        }

        val notification = NotificationCompat.Builder(this, channelId)
            .setContentTitle(title)
            .setContentText(message)
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            .build()

        manager.notify(1001, notification)
    }
}
```

The MyHostApduService handles all HCE communications using the processCommandApdu function to handle the transmitted data. In this case record the ID and Send a notification. (WIP)

NFC Reading

The 'Hunter' device reads the HCE signal from the 'Host' device to register a 'Catch'. This is handled by the NfcReaderService.

During setup this is also used by the 'Host' to register each playing device to the game. (MASSIVE WIP)

```kotlin
package com.example.nfctag

import ...

class NfcReaderService : Service() {

    private lateinit var nfcAdapter: NfcAdapter
    private lateinit var pendingIntent: PendingIntent
    private lateinit var intentFilter: Array<IntentFilter>
    private lateinit var techList: Array<Array<String>>


    override fun onCreate() {
        super.onCreate()

        nfcAdapter = NfcAdapter.getDefaultAdapter(this)

        val intent = Intent(this, HostActivity::class.java).apply {
            addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP)
        }

        pendingIntent = PendingIntent.getService(
            this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_MUTABLE
        )

        intentFilter = arrayOf(IntentFilter(NfcAdapter.ACTION_TAG_DISCOVERED))
        techList = arrayOf(arrayOf(IsoDep::class.java.name))
    }


    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        startForeground(1, createNotification())

        Log.d("NfcReaderService", "Service started")

        if (intent?.action == NfcAdapter.ACTION_TAG_DISCOVERED) {
            handleNfcIntent(intent)
        }

        return START_STICKY
    }
```

```kotlin
private fun createNotification(): Notification {
    val channelId = "nfc_reader_channel"
    val manager = getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        val channel = NotificationChannel(channelId, "NFC Reader", NotificationManager.IMPORTANCE_LOW)
        manager.createNotificationChannel(channel)
    }

    return NotificationCompat.Builder(this, channelId)
        .setContentTitle("NFC Reader Active")
        .setContentText("Waiting for nearby devices...")
        .build()
}

private fun handleNfcIntent(intent: Intent) {
    val tag: Tag? = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
        intent.getParcelableExtra(NfcAdapter.EXTRA_TAG, Tag::class.java)
    } else {
        @Suppress("DEPRECATION")
        intent.getParcelableExtra(NfcAdapter.EXTRA_TAG)
    }
    val isoDep = IsoDep.get(tag) ?: return

    try {
        isoDep.connect()
        val response = isoDep.transceive(byteArrayOf(
            0x00.toByte(), 0xA4.toByte(), 0x04.toByte(), 0x00.toByte(),
            0x07.toByte(), 0xF0.toByte(), 0x01.toByte(), 0x02.toByte(),
            0x03.toByte(), 0x04.toByte(), 0x05.toByte(), 0x06.toByte()
        ))
        val deviceId = String(response, Charsets.UTF_8)
        Log.d("NFC_SERVICE", "Received Device ID: $deviceId")
        savePeerDeviceId(deviceId)
    } catch (e: Exception) {
        e.printStackTrace()
    } finally {
        isoDep.close()
    }
}
```

```kotlin
}

private fun savePeerDeviceId(id: String) {
    val prefs = getSharedPreferences("known_devices", Context.MODE_PRIVATE)
    if (!prefs.contains(id)) {
        prefs.edit().putBoolean(id, true).apply()
        Log.d("NFC", "New device detected and stored: $id")
    } else {
        Log.d("NFC", "Known device reconnected: $id")
        //Caught!
    }
}

override fun onBind(intent: Intent?): IBinder? = null
}
```