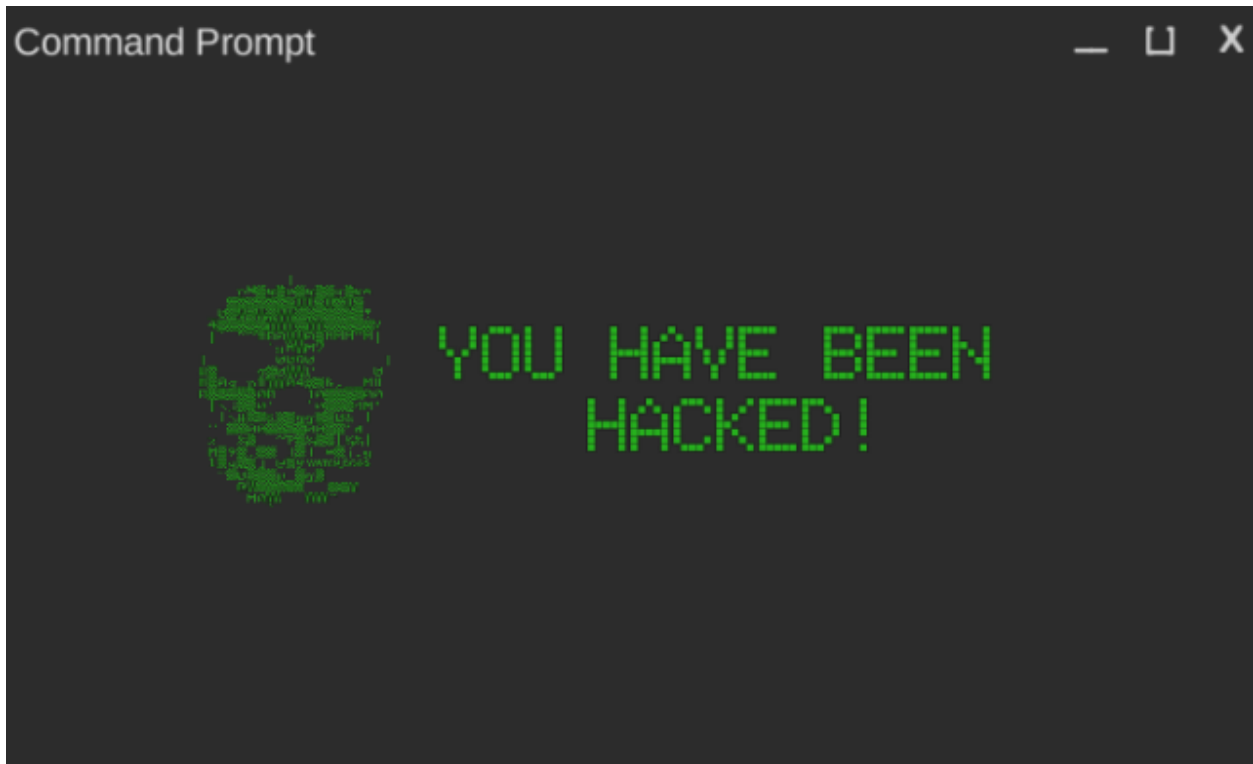


23:59!



Group Vuk 3

Imaan Abusamra - 1839590

Ammaarah Noormohamed - 1408224

Tenisha Moodley - 2105982

Intent

The game is a 2-dimensional (2D), click and type, interactive puzzle-solving game in which players play as a student whose computer has been hacked trying to submit their university assignment before the deadline arrives. The hacker toys with the student, presenting various obstacles and issues that need to be overcome before they are able to submit the assignment. Working within the constraints of a desktop setting, these obstacles are presented in the form of logical puzzles that use PC conventions such as having the laptop revert to a different language and requiring the player to find the settings menu and changing the language back(see **figure 1**).

Furthermore, smaller puzzles, such as finding out how to communicate back to the hacker through the Note-Pad application, are also implemented throughout the game in order to reinforce the puzzle-solving, interactive and explorative elements of the game. Puzzles will revolve around the perception and exploration for hints or answers found within images and texts found on the player's desktop, either in 'applications' or just on the desktop itself.

Designing a conventional computer system simulation experience which puts emphasis on the simulation experience drawn from the tension of error messages, pop up windows and other conventional computer system problems(see **figure 2**). However, because this is more than just a simulation, as it has a bit of a narrative and a two week development limit, the use of less realistic errors such as sped up timers are used to increase the feeling of fear, tension and anxiety of having their simulated desktop be infected with a hacker or virus while the deadline of an assignment is due.

Process

The initial prototype differed significantly from the current iteration. It made use of puzzles that broke typical PC conventions. As an interactive desktop experience, it makes more sense to work within the constraints of those conventions, thus the current iteration is closer to a desktop simulation than before. The added simulation factor also allows us to use those expected conventions and integrate them into the puzzle-solving

aspect, as conventions play a part in helping the player figure out what to do and how to solve various problems.

The puzzles are presented as various problems set by the hacker that the player encounters in their attempt to submit the assignment on time. For the majority, each puzzle is inspired by real desktop problems experienced by the designers. However, many of these puzzles are also linked to a riddle set by the hacker, requiring a player to interact with, and search on their desktop for the solution. Only once this solution is presented to the hacker will the player be allowed to continue to solve the computer issue in a more conventional way.

All the puzzles presented force the player to explore and further interact with various aspects of the desktop setting, further aiding immersion and ensuring that interactive experience is achieved.

The first puzzle is just a simple issue of the player needing to figure out how to communicate with the hacker. By eventually typing into the notepad application, the rest of the game is triggered and this also allows the player to realise that this notepad is the channel of communication, something which is also needed for future puzzle problems(see **figure 3**).

The second puzzle starts with the hacker hypothetically changing the player's desktop colour settings. The entire screen has now lost all colour(see **figure 4 and 5**). The player can try and fix this in the settings tab(see **figure 6**), but this will only work if the player first solves the first riddle presented by the hacker(see **figure 7**). The solution for this riddle will be found in one of the recent files in the Wurd application, which the hacker will hint at to give the player a direction to start searching and interact more with the desktop.

The third puzzle begins with the hacker hypothetically adjusting the player's desktop language settings(see **figure 4 and 5**). Everything is now displayed in Korean, even the

message sent by the hacker. Based on the previous puzzle the player needs to make the connection that this message is a clue. Interacting with this, the player will realise that the message can be copied(see **figure 8**). It will then need to be pasted into the Google translate window, which can be found through interacting with the desktop. Translating the clue will allow the player to find this riddle's solution, which is in the Strypify window, meaning the player will only find it if they interact with this window.

The final puzzle is easier to figure out and solve than the previous two, serving as a sort of denouement to the entire encounter. Solving this puzzle will uncorrupt the submission file and let the player finally submit their assignment, if they still have enough time.

Leaning further into these desktop conventions helps a player figure out ways to solve the problems encountered without needing any additional instructions, allowing the puzzle-solving theme to stay intact. However, we did not entirely simulate a general desktop experience. Minor changes are made, such as changing the names of well known applications, to create a slight subversion of those expectations(see **figure 9**). This allows us to represent a simplified version rather than an actual desktop replica without entirely breaking conventions set. The game now intends to be a realistic, although simplified simulation of a computer system. The communication between the hacker and player is an essential game progression element as well as a developer of the puzzle-solving constraint of our game as previously, mentioned, however our initial intention behind the communication between the hacker and the player was to be an integral part of the interactive game system further than just the progression of the game so that the game could focus on a more interactive narrative game instead of a simulation.

There are various additional interactive experiences in this game that don't necessarily drive the core puzzle experience. However, one of our main intentions is to convey an interactive experience and these additions build that immersion and the feeling of interacting with some sort of desktop. This includes aspects such as the Strypify playlist, which allows the player to play any of the songs from the list like they would on their

own desktop music application, sort of setting their own soundtrack for the game(see **figure 10**). One of the song names holds the answer to the riddle, but playing the songs themselves doesn't help in the puzzle solving experiences, just the interactive one.

The timer has been a part of the game from the first iteration. It aids in creating the tension-filled atmosphere we are trying to achieve, thematically fitting the experience of, both being hacked, as well as rushing to submit an assignment on time. Initially this timer was attached to the submission page, as a limit for completing an online test, remaining once the test window was closed. However, it made more sense in the desktop setting to rather have it replace the normal clock display on a computer screen, rather counting up towards a submission deadline time. The timer thus also serves as one of our end conditions, meaning the player loses if they cannot overcome all obstacles and submit their assignment in time. The adjusted time frame is slightly wider than initially intended. While still wanting to create that thematic tension filled experience, it is also important to allow the player time to engage with various interactive experiences in the game.

Other methods were also implemented, in an attempt to enforce the tension and anxiety-filled atmosphere that we intended to create. These include the use of the computer error message window that appears when the player attempts to upload their assignment (see **figure 11**), as well as the sudden change of the in-game desktop (mentioned previously in **figure 5**). Both these implementations act as a sudden occurrence within the game and are intended to create tension through the player's association with these issues in real life.

Reflection

Simulation requires participation from the viewer, whether it is physically interactive or not. So while researching various ways to go about implementing the hacker based interactive puzzle-solving constraints, we came across the idea of simulation as interaction instead of merely requiring the player to interact with the narrative and the

puzzles of the game. We were then introduced to the question of whether our game intends to be a simulation or a simulation lite? Our initial idea focused more on written narrative.

As previously mentioned, we abandoned previous ideas of more conventional styled puzzles revolving around twitch skill shooter styled puzzles as well as find the missing objects styled puzzles, and decided that if we were to flourish and develop the puzzles within the constraints of a narrative based interactive game then keeping the puzzle themes and mechanics within those constraints was necessary to not breaking the game flow we intended of a simulation. The realisation of how taxing building new knowledge for puzzles and for narrative for an immersive experience game such as this, is what drove us to eventually work with conventions of computer systems, basically to make a simple although realistic simulation instead of a simulation lite.

We started to explore puzzles according to the conventional computer system puzzles that people face everyday while working or interacting with their computer systems. We started to think about how we can follow computer working mechanics that communicate our intentions behind the game through the familiarity of a working computer system, hence the simulation of one being the core mechanic of the game and therefore also easy to understand and explore. This led us to start deliberating how to copy files, renaming files and working with the narrative as a simulation rather than the twitch skill and written narrative elements that we originally planned to work with.

How does something load? How do we install? How are things moved around? These questions started the iterative process of our revised idea of a simulation, puzzle-solving explorative game rather than the interactive narrative based puzzle-solving game idea. By wandering about what specifically would happen if the hacker forces the player to do things that they understand such as changing the language or the internet or the colour of their computer. Due to the familiarity of troubleshooting a computer system, the player would think to go to their computer settings as we intend to with the design, display and development of the simulation to correct the system problems. This is how

we progress the game narrative, as the player will only be able to correct their settings, and therefore the puzzles, after they explore the desktop, unpack what the hacker expects of them and then respond in the way we want either by solving the hackers riddles.

Changing the core puzzle interactions halfway through the assignment timeframe was needed to better meet our intentions without breaking computer conventions too much and giving players unrealistic expectations. However, further simplifying those conventions as earlier mentioned is also an important necessity to allow us to submit something workable that accurately conveys the game idea within the game design assignment time constraints. While a number of technical issues are still present in our current prototype, due to time constraints, more focus was put into making sure the game can flow from beginning to end, as well as into creating that hacked interactive experience that we are aiming for, as it is important to convey that experience in a workable prototype. Future iterations can work to fix these minor issues.

As an interactive experience, where the player has freedom to explore various in-game 'applications', multiple functions had to work alongside each other. This includes aspects such as the timer beginning alongside the hacked sequence, various changes happening in different windows when hacked, as well as just the various windows themselves and interaction with each one. This presented a number of issues and additional fine tuning. In order not to overscope, considering the time constraints of the project, these issues played a part in choosing which conventional desktop expectations to simplify. Further work on this prototype in the future could potentially allow us to better explore these issues and create something that better meets computer conventions.

Creating multiple problems at once for the player to overcome, helps to enhance that sense of tension and panic the game intends to convey. However, through playtesting, it was noted that players would often get confused or go off track, even when the puzzles

were enacted in a linear fashion because of the lack of subtle logic with the puzzles, so numerous play tests were committed before deciding which puzzle mechanics and elements to develop and how from the given results(see **figure 12**). The introduction of the sticky note tab with its automatic to-do list entries, while somewhat breaking that immersion of the player needing to interact with things themselves, helps the player to see where they stand in the game, serving as current objective reminders. Email, and calendar notifications also further serve as hints and reminders of what players would need to do in the game.

The time and timer mechanics were better adjusted from previous iterations to better fit the conventional layout of a computer time and then further develops into a more timer like mechanic that acts as a tension reinforcer to make the player feel like they have to act fast to establish how to solve the problems being forced into their lives at such a crucial time such as an online exam (see **figure 13**). The timer was therefore kept, despite breaking convention, because the design team felt it necessary to keep that sense of tension and panic, and it fits in thematically into the game.

Using typing and clicking to navigate gives players a means to explore within the expected conventions of a desktop setting. Going in this direction also opens a whole possibility of encounters and additional puzzles that can be explored by the designers in possible future iterations of the game. The ones presented in this prototype work to convey the possibility of a game like this, and create the intended experience.

Appendix

Intention Of the Developers	Intention Of the Game
Create tension and a sense of anxiety for the player. (Done through suddenly getting hacked, as well as assignment timer and hacker penalties).	Immerse the player in the experience of a computer system.
Simplistic representation of the game system, easy to understand mechanics and the communication through a familiar computer system which encourages player logic and immediate understanding of the game system without having to explicitly state it.	Interactable game system that encourages exploration through the game mechanics and constraints.
Developed around emphasis on hacker and player communication through the puzzle-solving mechanics.	
Explore a variety of puzzles: word based through riddles, as well as completable tasks.	

Figure 1 Lists showing the intention of the game, from the developers and from the game



Figure 2 Screenshot of the simulation notification pop-up of an email notification on the top right corner

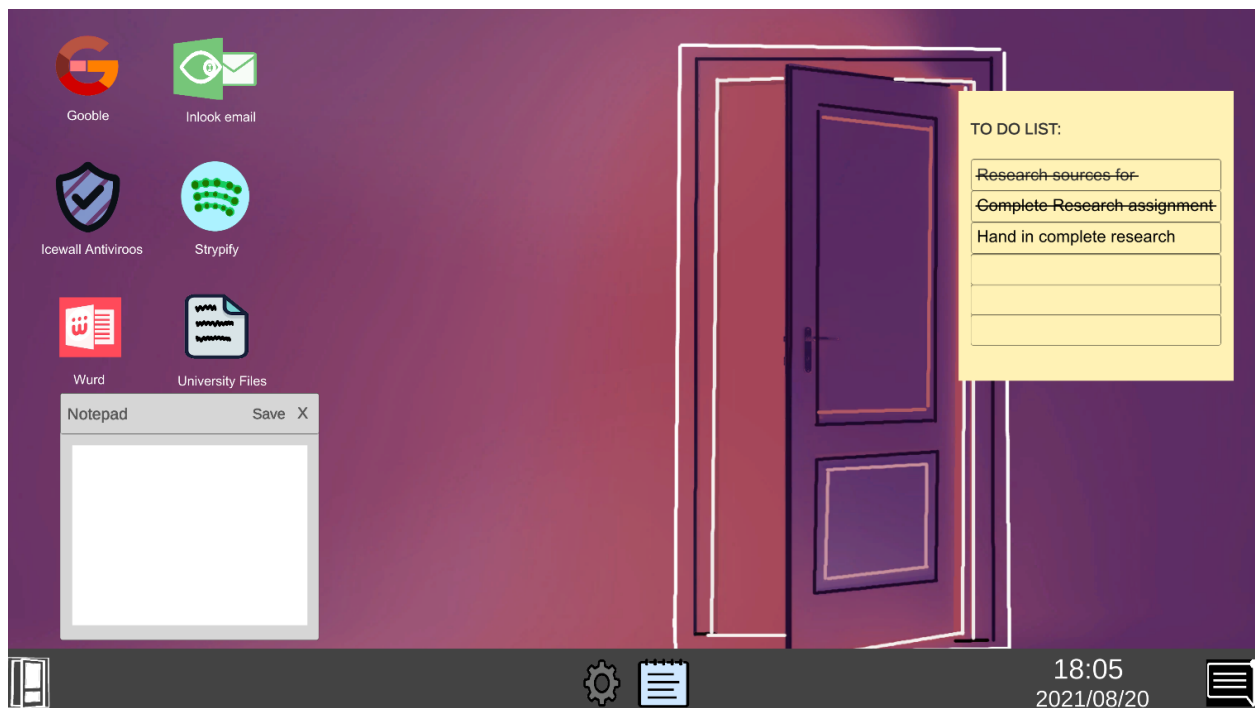


Figure 3 Screenshot of the NotePad application, where the player is allowed to interact with the hacker shown in the bottom left corner



Figure 4 Screenshots of the Non-Hacked Desktop in colour and the game system notification in English at the bottom left



Figure 5 Screenshots of the Hacked Desktop and in black and white and the game system notification in a Korean at the bottom right

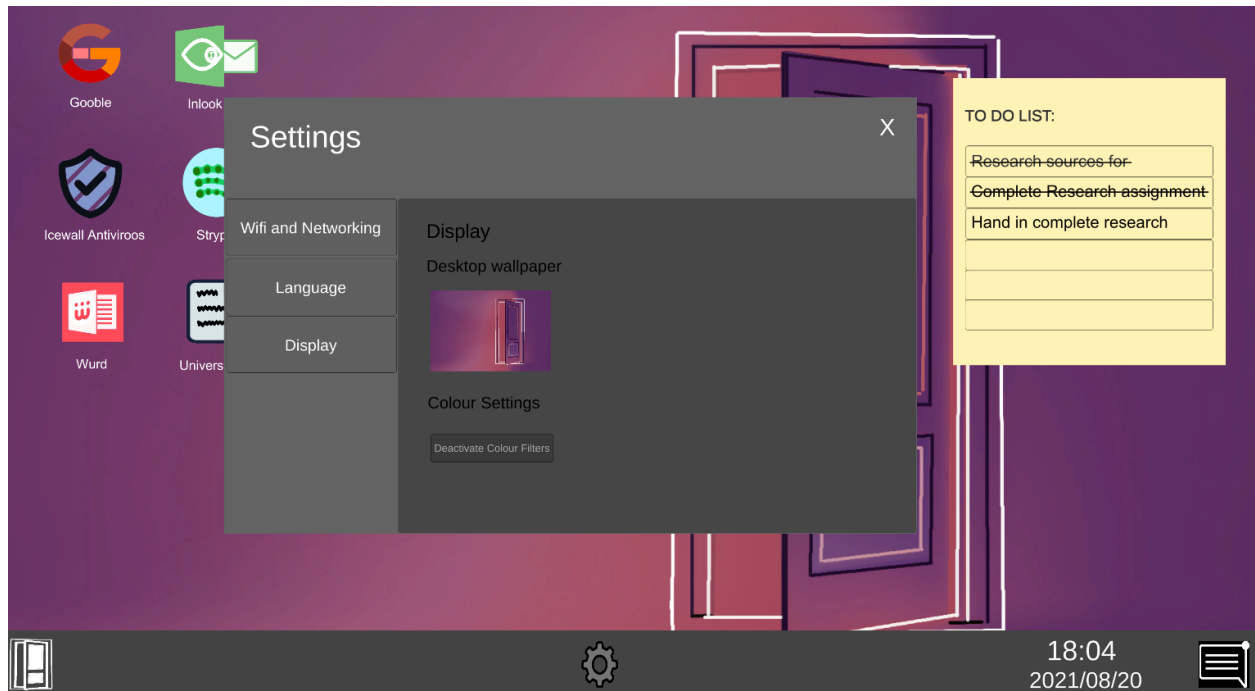


Figure 6 Screenshots of the player engaging with simulation realistic desktop experiences, such as the Settings Menu

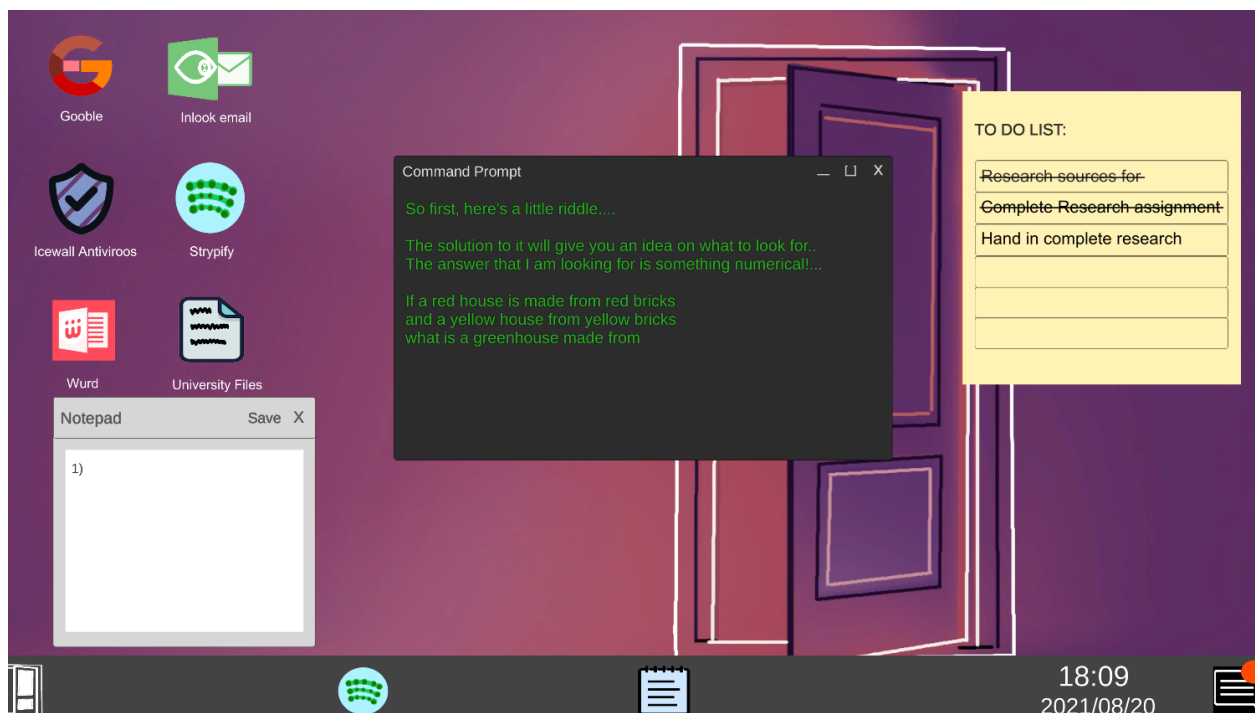


Figure 7 Screenshot of the Hacker Terminal where the hacker communicates riddles to the player but the player cannot communicate back

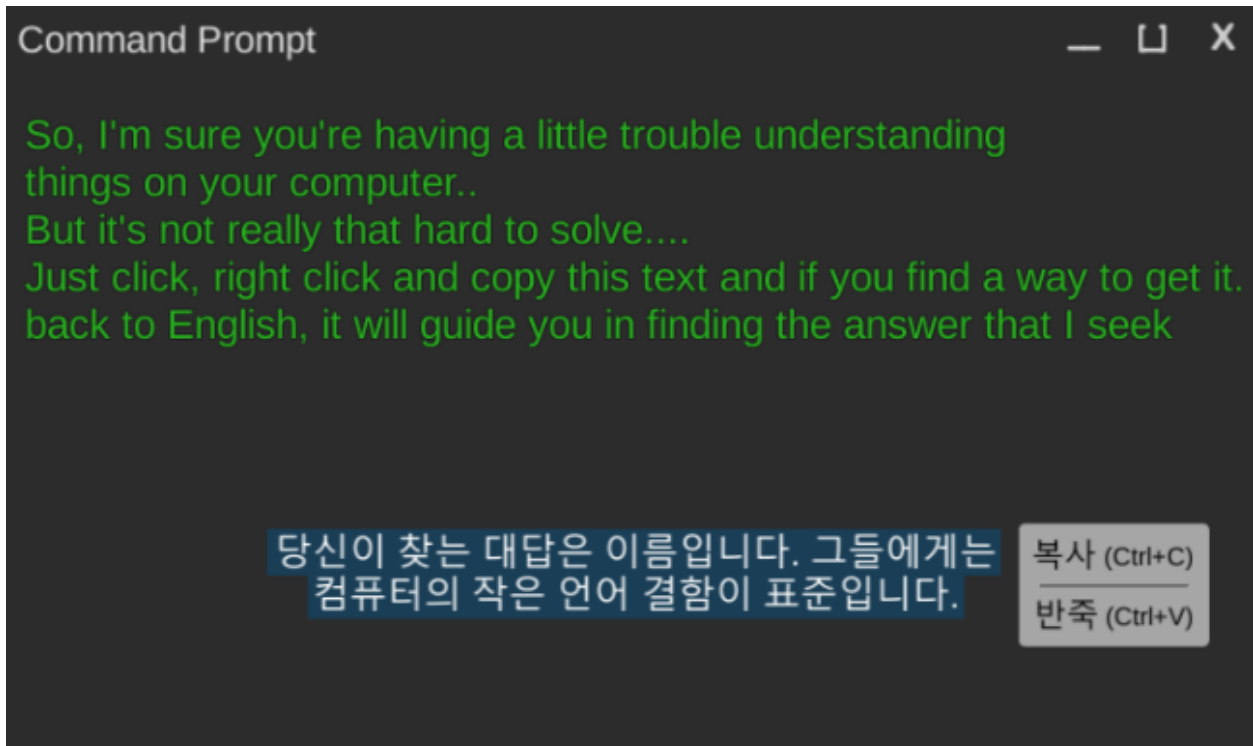


Figure 8 Screenshot of the message (displayed in Korean because by this stage the player has already been hacked and their computer system is now in Korean thanks to the hacker) that can be copied(Ctrl+C) and pasted(Ctrl+V) with right clicking over the message



Figure 9 Screenshots of the Current desktop iconsInternet Explorer - Gooble, Email application - Inlook Email, Antivirus - Icewall Antiviroos, Music Application - Strypify, Written Document Application - Wurd, System Folder Location - University Files, Settings, Notepad Application

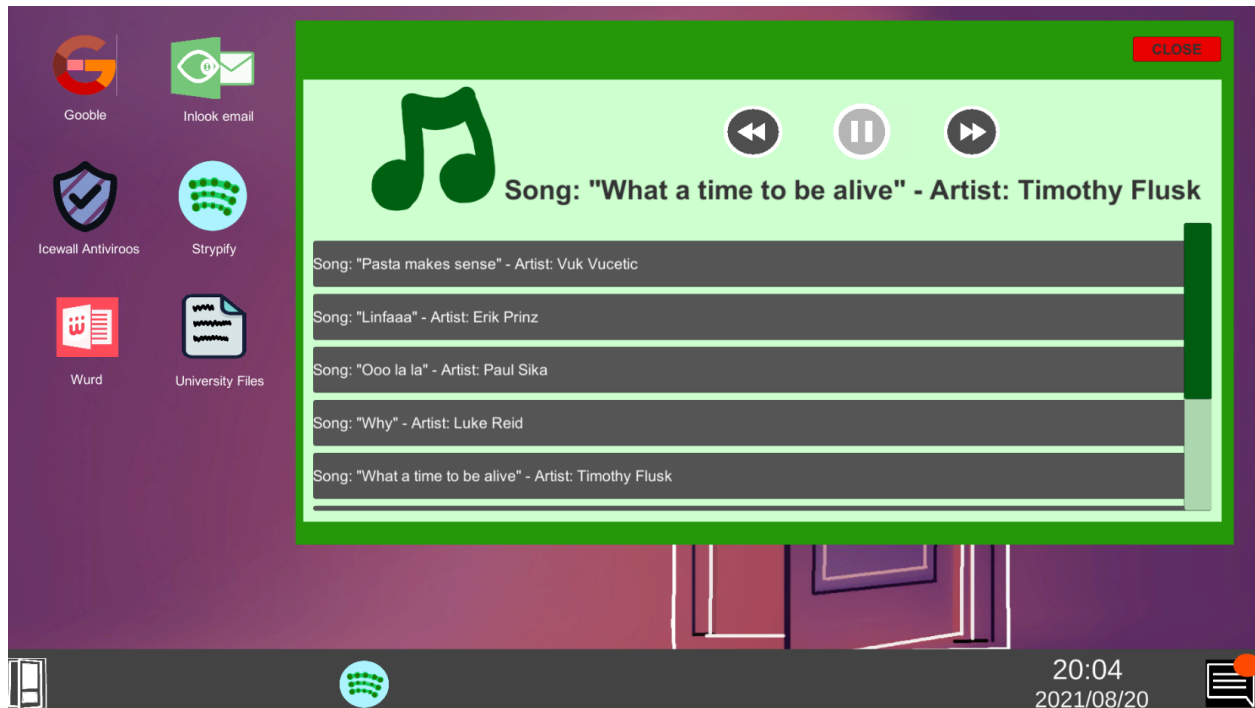


Figure 10 Screenshot of the Strypify song playlist

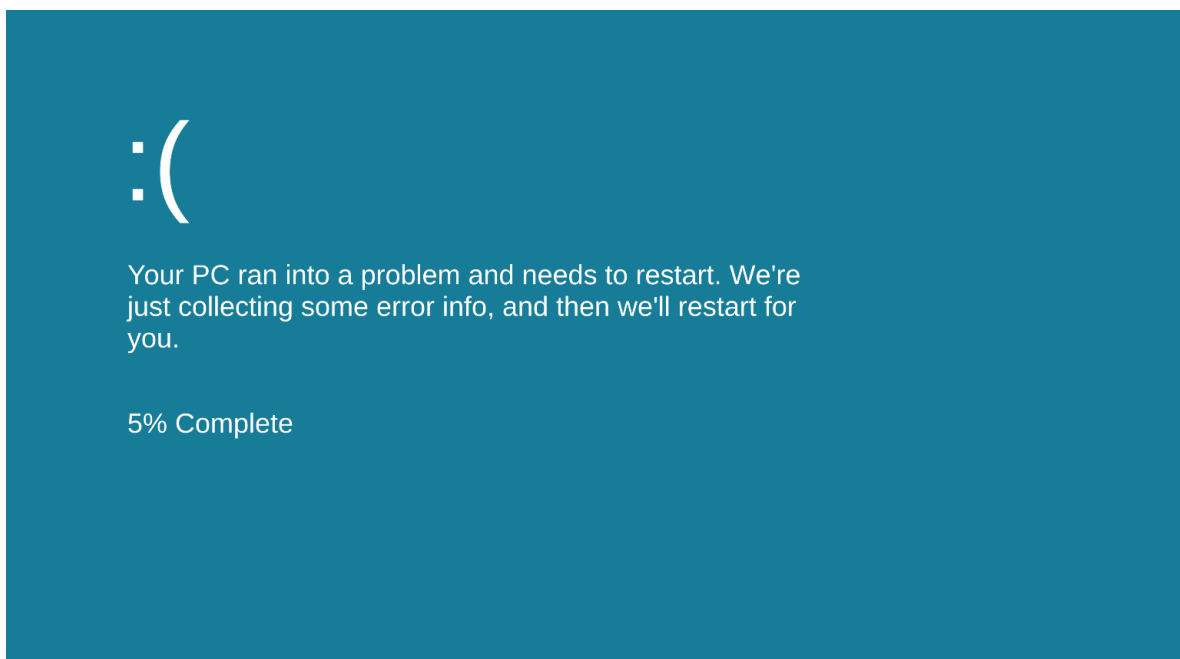


Figure 11 Screenshot of error window pop up

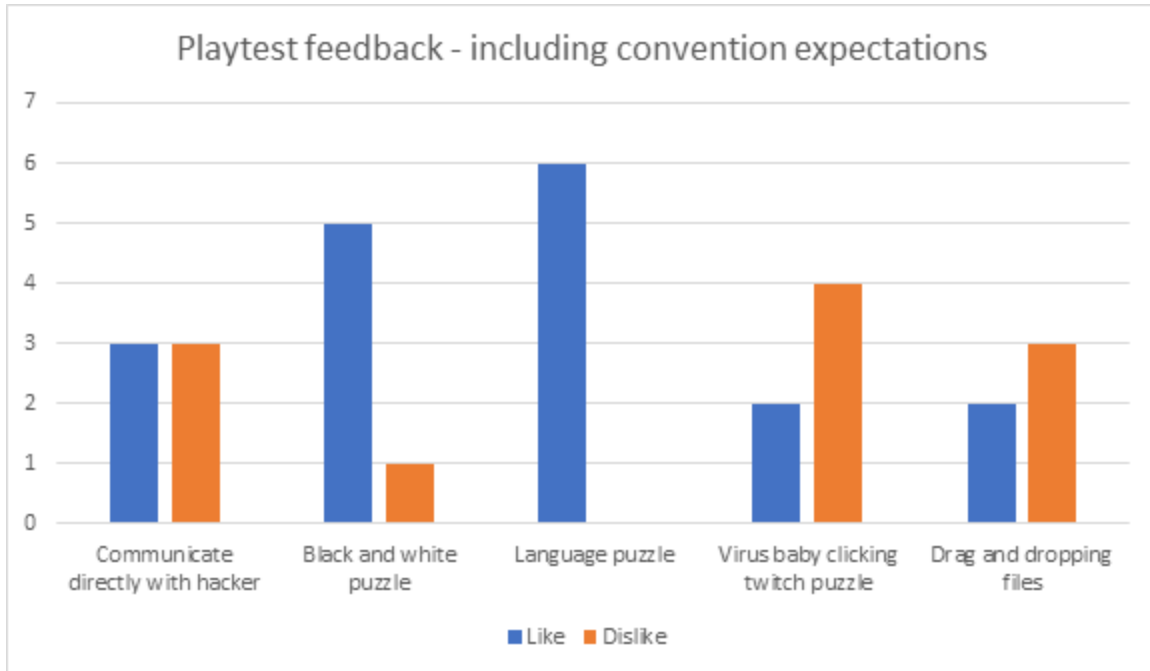


Figure 12 Bar Graphs showing the results of play testing/surveys concerning certain decisions within the game such as which mechanics and puzzles fit conventions and our intentions of the game

The Puzzle	The Solution
1) Hacker asks a riddle and expects an answer	Finding the NotePad to communicate with the hacker
2)Player is to answer the first riddle	Typing in 1511
3)Player is required to answer second question	Typing in Kim Woosung
4)Player is required to answer third question	Typing in Horton
5)The hacking ends and the player is required to end the game	Submitting your assignment

Figure 13 Table showing the solutions to all of the core puzzles