# Design Doc

A top down stealth game, that gets the player in touch with their inner B-movie hacker to take out security cameras and open doors.

There are two screens and gameplay flavours, one for hacking and one for stealth.

## Stealth

A top down camera perspective. Security cameras and guards pose as threats for the player. Entering their sight line is a game over.

Guards follow a patrol route. The player should move slower than the guards to force them to learn the patrols and carefully operate around them.

Security cameras are attached to walls and rotate. The player can hack them by clicking on them. Once hacked, a camera can be turned on or off. There should be parts of the level that require you to hack a camera.

Doors will block the players progress, forcing them to hack it. (Alternatively, put a security camera in a corridor that can’t be snuck around.)

Whilst hacking, the player cannot move. This forces them to find a good hiding spot away from guards. An ideal level layout should encourage the player to hack a security camera so they have a spot to hide from a guard, whilst getting close enough to hack a door.

## Hacking

The hacking should be an equally vital part of the game. The hacking should encourage the player to type on their keyboard as quickly as possible. The speed of their typing correlates to the speed of the hacking.

An image of a keyboard appears on the screen and certain keys flash. If the player presses the key in the window that it flashes on the screen, they gain hack progress. The timing window for the flash should be short enough that relying on reaction time alone isn’t enough to hit all the keys. The player will want to type as quickly as possible in the hopes of getting lucky. The flash should be long enough that some keys can be taken with reaction speed if the players fingers are near the corresponding key. (This gives the system a bit of skill expression)

To make the system more interesting. Pop ups will occasionally appear and slow your progress. The player must click on them with their mouse to get rid of them. This will impact their ability to hit the keys. The player will have to decide when to grab their mouse and delete the pop ups. This will give an ebb and flow to the hacking system as the player moves between typing (reaction speed challenge) and clicking (mouse movement accuracy challenge.)

The pop ups should further sell the idea of being a B-Movie hacker by using language like “Mainframe Firewall”, and “Router Point Data.”

## Dialogue

The game also has space for dialogue to give some personality to the project.

# Level Design

Diagram, schematic

Description automatically generated

Circles are the objective the player needs to reach. Lines are guard patrols.

The level is split into three wings the player can complete in any order. Each wing has a series of two challenges, with a mix of challenge styles. The player will have to back track after each challenge and this is intentional. It may be tempting to skip hacking the cameras if you think you can sneak around them, however you’ll have to back-track through the same area whilst dodging the cameras again. Either way the player will have to avoid the guards, so the gameplay is still engaging.

## First room

The player starts in a room with a security camera blocking their way out. This would have been a door but I didn’t finish. The player is taught about the hacking mechanic which they need to proceed.

The first room should have led to more of a hub style room, rather than a corridor with three passages. Its not very visually interesting which makes navigating harder and backtracking less exciting.

## Left Wing

The left wing contains 2 objectives. First the player needs to sneak around a guard. The security camera can be hacked from a distance OR snuck around with precise timing. The second objective can also be completed by dodging the cameras or hacking them. Its significantly easier.

The left wing is the easiest to do, because people will hopefully be bias towards doing the left track first (we read left to right. It’s a cultural thing.)

## North Wing

The player first has to sneak around a guard patrol, which leads to a second room. There is a bit of extra space beneath this room so the player can scan the camera over the whole room. There’s a guard and two security cameras. The player can sneak around them but its not easy. De-activating the cameras first will help.

## East Wing

The most difficult path. Multiple guard patrols overlap, requiring the player to plan their movement carefully. The first objective is locked behind a security camera. The player needs to hack this camera but the only safe place to do it requires them to sneak behind a guard and enter a cubbyhole. The second section has a large quantity of security cameras. The player could hack as many of them as they want in order to clear enough of a path they think they can sneak through.

# Considerations / Thoughts

## Pop Ups

It would be nice if the pop ups could spawn in bursts (like in waves) rather than one at a time between random intervals. This would support the idea that pop ups are meant to be dealt with in a group rather than one at a time. However, it might also remove the players agency of making the choice of when to click on the pop ups and when to focus on typing as fast as they can.

## Character Controller

Two options:

WASD / Arrow key movement controls

Or the player uses the mouse to move about.

Either way, the movement script needs to disable whilst the player is hacking. (The player needs a way to exit the hacking minigame early too)

Mouse Controls:

+ The player will want to click on hackable objects, if theyre playing with just one hand, the mouse lets them do both.

+ Can move in more than 8 directions (minor benefit)

- I can't do pathfinding, so trying to move around corners may feel unintuitive and uncomfortable

Keyboard (WASD / Arrows):

+ A lot easier to create.

+ Allows the player to move their body whilst positioning their mouse to the object they want to hack (minor benefit)

+ Sudden movements are easier to execute (e.g. the player might need to rush behind a corner because they didn't notice a guard coming, but they need to spend time moving their mouse to where they want to go. with WASD, its as simple as clicking the right button.)

+ Allows the mouse to possibly control the camera whilst not taking away the players ability to move. (Arguably a disadvantage if I want to make looking around come at the cost of mobility)

- Player can only move in upto 8 different directions (their inputs are slightly less precise. This isn't too important for my game though which shouldn't depend on such precise movement inputs)