Game
Programming
Laboratory

Factory Reset

Take back your life

TEASER

Nicolas Hafner - Producer, Programmer, Visual Artist, Designer
Amir Heidari - Programmer, Designer
Gengyan Li - Programmer, Designer
Nicole Manojlavic - Sound Designer, Writer, Designer
Pulkit Singh - Visual Artist, Designer

Contents

1

Formal Project Proposal

1.1. Game Description

1.1.1. Overview

Our game is a 2D stealth platformer, set in the dystopian future of 2099. You play as a member of an underground group of outlaws and your job is to smuggle valuable information. As the internet is under heavy surveillance by the mega corporations, the only feasible way to deliver this information is on foot.

Use parkour, hiding places, and gadgets to your advantage to avoid being detected by the private security roaming the streets. Should you get detected, you will be forced to try and lose your pursuers through fast-paced action platforming. Littered throughout the level are terminals and discarded storage drives that can be picked up to find new information and increase your score.

1.1.2. Background Story

The year is 2099. Corporations have grown to a massive scale and have over taken most of the governmental capabilities. The corporations control almost every aspect of most people's lives – their workplace, the food they buy, the homes they live in, their communication tools. All personal devices and even the internet have become entirely controlled, with backdoors built in. Attempting to encrypt and hide your communication results in an immediate penalty.

Despite all this, a group of underground outlaws attempt to live their lives free from the corporate shackles. However, in order to do this they need to avoid the ever-present surveillance of the mega corporations. Part of doing so requires knowing what they are planning next. Your job is to retrieve the information from your double-agents and deliver them to the next checkpoint. Without this information, the corporations are going to find your underground settlement and wipe it out.

1.1.3. Design Decisions

The game is primarily a stealth game where you have to avoid being detected by corporate drones and cameras as you make your way through the level. Cameras and Drones have obvious view cones that the player must evade to avoid detection. See Figure 1.1.

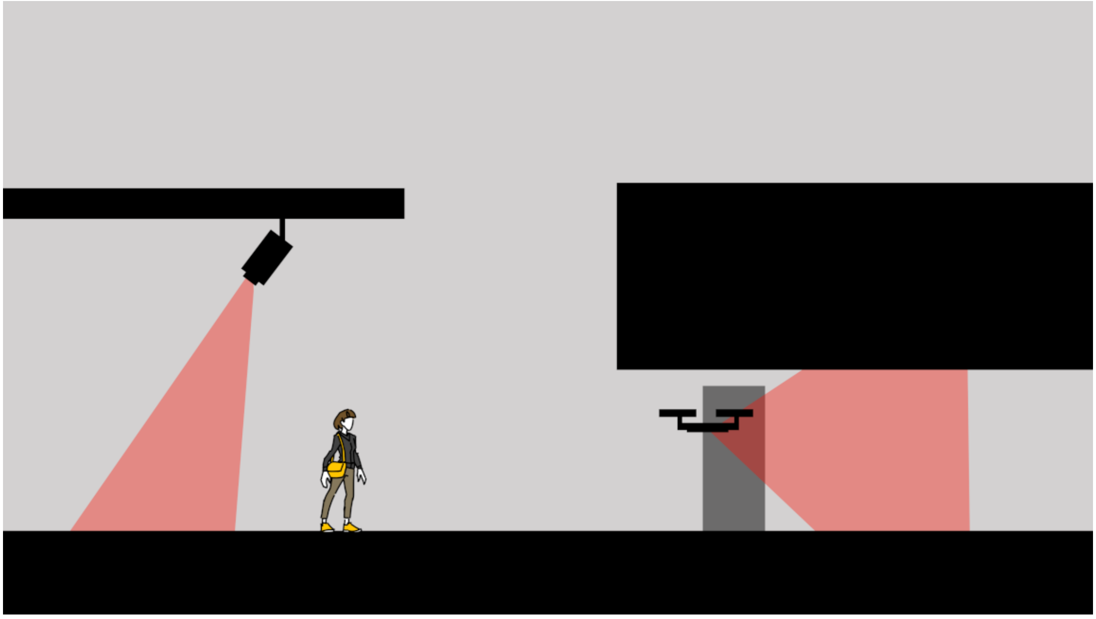


Figure 1.1.: *A basic sketch of detection areas.*

To avoid detection the player can use parkour or hiding places as shown in Figure 1.2.

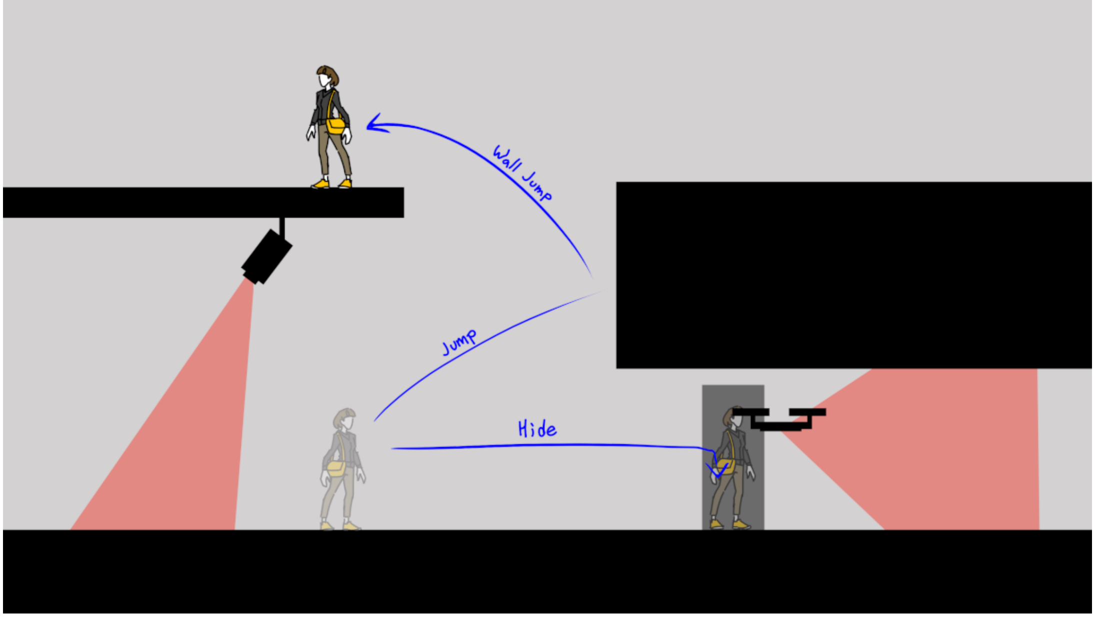


Figure 1.2.: *Using parkour or hiding places to evade detection.*

Throughout the level are terminals and memory drives that contain additional information. The player can pick these up for a bonus. See Figure 1.3.

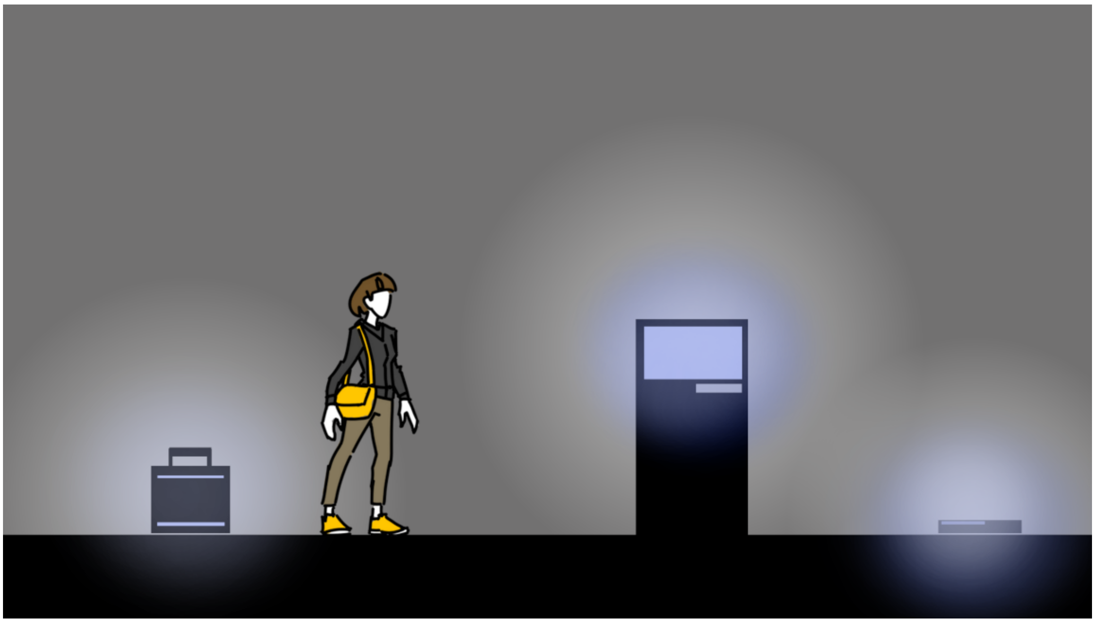


Figure 1.3.: *Items that provide additional information that can be collected.*

When the player is detected, a chase sequence is initiated where the drones start following you, as shown in Figure 1.4.

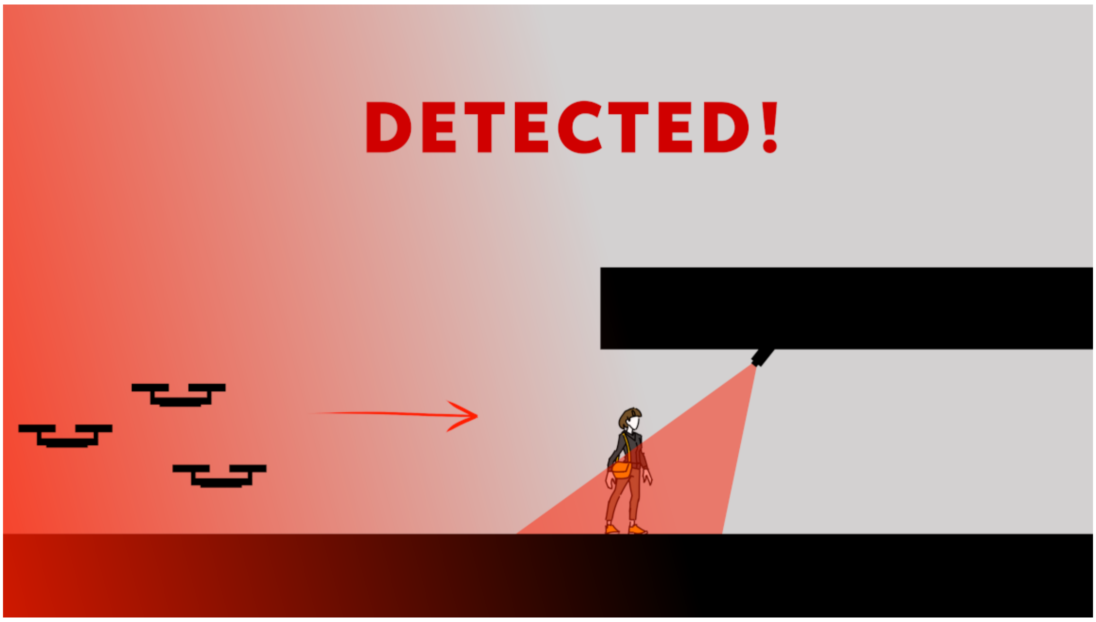


Figure 1.4.: *On detection, the player gets chased by drones.*

The player has to outrun the pursuers via precise platforming and reach the end, or find a hiding place and wait for the evasion counter to run out. See Figure 1.5.

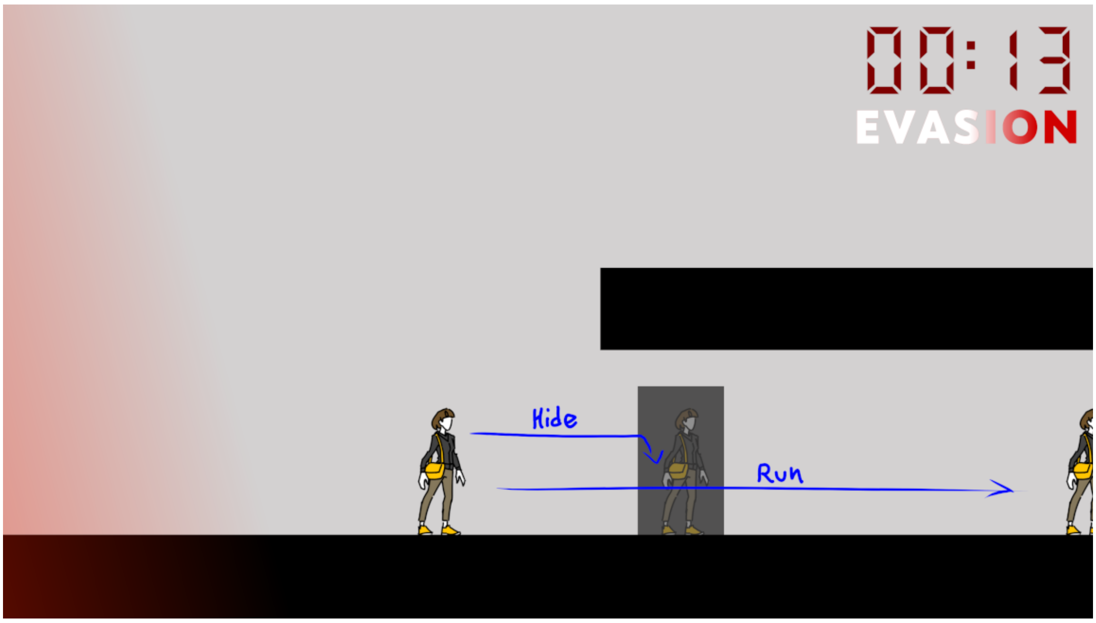


Figure 1.5.: *During the chase, the player can either gun for the end of the level, or try to hide.*

Immediately going for the end will give you a better time score, but, as you won't be able to pick up additional information during the chase, you'll have a much worse score overall.

Running around or interacting with certain items causes noise that can be heard by some enemies. This will prompt them to investigate the area.

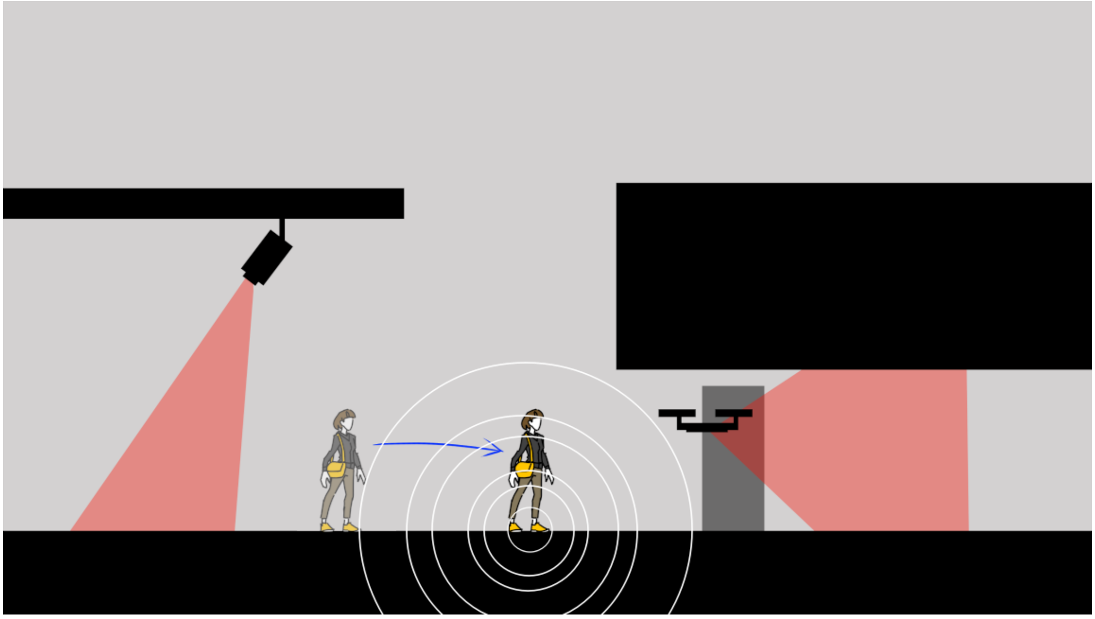


Figure 1.6.: *Running around will cause noise that can alert enemies.*

Our game takes inspiration from other titles such as Mark of the Ninja, Celeste, Mirror's Edge, Metal Gear Solid, and SpeedRunners.

1.2. "Big Idea" Bullseye

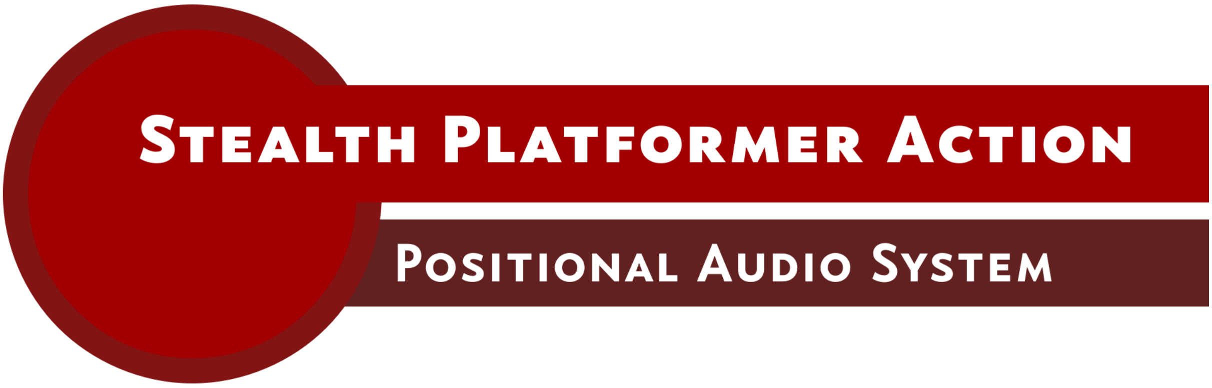


Figure 1.7.

1.3. Technical Achievement

To make the game environment feel more tangible and immersive, we will implement a positional sound system. To integrate this with the world further, entities within the environment will also be able to "hear" the sounds and react to them. This should couple well with the stealth focus of the game.

1.4. Development Schedule

1.4.1. Layered Task Breakdown

Functional Minimum

- A level composed out of blocks.
- Allowing the player to walk and jump.
- Regions that will cause the player to be detected.
- Basic sound effects.
- A "wall of death" to simulate chases.

Low Target

- Player animations for walking and jumping.
- Basic enemy AI.
- Allowing the player to collect information.
- A full level including graphics.
- A chase countdown to evade chases.
- Level saving and loading.
- Basic sound design.
- A score screen.

Desired Target

- Two or three fleshed-out levels.
- A basic story.
- Good sound design.
- More advanced chase mechanics and visuals.
- An editor that allows placing blocks and entities.

High Target

- Music.
- Checkpoints.
- Fully-fledged editor.

Extras

- Gadgets for environment changes and interaction.
- Crouch walking.
- Animations for enemies.
- A significant number of levels.
- Different player characters and NPCs.
- Slopes.

1.4.2. Task List

Task	Time Estimate	Assigned To
Chase Sequence	12h	Amir
Detection Regions	8h	Gengyan
Enemy Art	24h	Pulkit
Environment Art	100h	Pulkit
First Level	3h	Nicole
Formal Proposal	10h	Nicolas
Future Levels	∞	Everyone
Information Collection	3h	Gengyan
Interaction Feedback	12h	Amir
Level Editor	20h	Nicolas
Level Geometry & Collisions	20h	Gengyan
Level Saving & Loading	6h	Nicolas
Meetings & Discussion	35h	Everyone
Music	10h	Nicole
Particle Effects	15h	Amir
Patrolling Enemies	15h	Gengyan
Player Animations	20h	Nicolas
Player Movement	10h	Gengyan
Positional Sound System	20h	Nicolas
Protocol	20h	Nicolas
Sound Design	20h	Nicole
Sprite Animation System	12h	Amir
Story Writing	8h	Nicole
Testing	∞	Everyone
Tileset Rendering	8h	Amir
User Interface	15h	Nicolas
Wall of death	8h	Amir

1.4.3. Timeline

	Rough Proposal			Formal	Prototype	First Demo			Interim Report		Alpha		Playtest
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
Proposal		■	■	■									
Story Proposal				■	■								
Level Geometry					■	■							
Art Preproduction					■	■	■						
Player Animations					■	■	■						
Detection Regions						■							
Player Movement						■	■						
Basic Sounds						■	■						
Rendering						■	■	■	■				
Wall of Death							■						
Animation System							■	■					
First Level							■	■					
Positional Sound							■	■	■				
Environment Art							■	■	■	■			
Chase Sequence								■	■	■			
Enemy Art								■	■	■			
Level Editor								■	■	■	■	■	
Information Collection									■				
Patrolling Enemies									■	■	■		
Story Writing									■	■	■	■	
More Levels									■	■	■	■	■
Particle Effects										■	■		
Advanced Sounds										■	■		
Testing & QA										■	■	■	■
User Interface											■	■	
Music											■	■	■

1.5. Assessment

This game will mostly be aimed at hardcore players that can enjoy some difficulty. The combination between stealth action and fast platforming should provide for an exciting, tense experience. As the game is set in a dark, dystopian future, the look and feel of the game should reflect that. The platforming should feel good, and the stealth rewarding. More specifically this means that the level design should allow challenging movement without it feeling frustrating. Similarly, the stealth sections should be fair, and it should be obvious to the player when they would get detected.

2

Prototype

2.1. Prototype Setup

Since our game is a 2D platformer, modelling it on paper turned out quite difficult. We opted for a turn-based board game where the 2D level is represented on a coarse grid (see Figure 2.1) printed out on paper. The level is divided up into multiple "chunks", indicated by the chunk-transition regions. Only one chunk is ever visible at a time. The player controls a piece of paper that can be moved along the paper grid. Each turn, the player must pick one of the following actions:

- If the player is not wall-sliding:
  - Stand in place.
  - Walk one tile to the left or right.
  - The player can jump to the left or right by three tiles.
  - The player can jump diagonally to the left or right by two tiles up and two tiles to the side.
  - If the movement in the previous turn was not standing, they can run two or three tiles to the left or right.
- If the player is wall-sliding, they can jump diagonally away from the wall by two tiles up and two tiles to the side.
- If the player is on a tile with a hiding spot, they can decide to hide.
- If the player is hidden, they can decide to unhide.
- If the player is on a tile with a pickup, they can decide to pick it up.

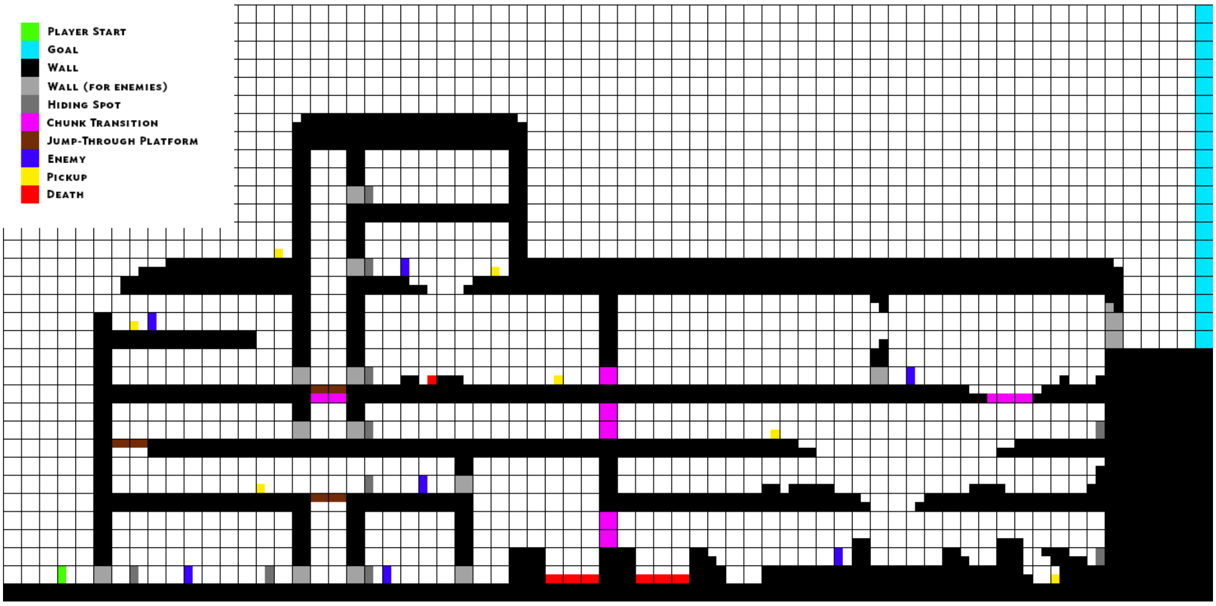


Figure 2.1.: *A sample level designed for the paper prototype.*

After the player has chosen an action, the following rules apply to simulate the physical interaction. Note that the order of the rules is significant.

- If there is a wall in the path of the movement direction, the player is stopped before the wall.
- If the player is wall-sliding, they are moved down by one tile.
- If the player is not directly above a wall or jump-through platform:
  - If there is an adjacent wall, they are wall-sliding.
  - If there is no adjacent wall, the player is dropped straight down to above nearest wall or jump-through platform.
- If the player is on a goal tile, they win.
- If the player is on a death tile they lose.
- If the player is on a chunk-transition tile, the chunk is changed to the one the player is moving towards.

The game not only models player movement, but also models enemy AI and the chase sequence to a limited extent. To model the AI, each enemy is modelled as a piece of paper. Another player moves these pieces every turn as the player moves, with the following rules:

- The enemy starts out facing right.
- The enemy is moved one tile in the direction they are facing if the next tile is not a wall, not a wall for enemies, and not directly above a wall.
- If the previous tile was the same tile as the current tile, the enemy changes the direction

it is facing.

Only enemies visible on the currently visible chunk are updated. Enemies have a "viewcone" (see Figure 2.2 that extends to the side they are facing. If the player collides with this viewcone, the chase mode is activated.

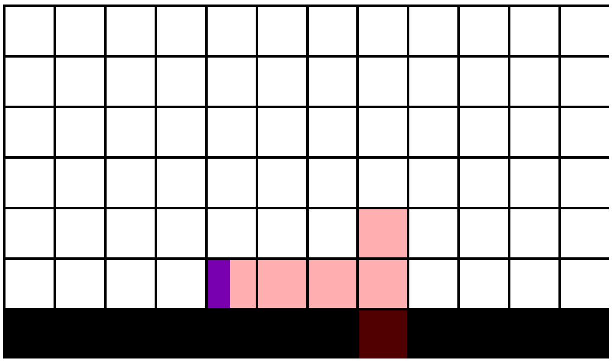


Figure 2.2.: *The viewcone of an enemy as it extends outwards.*

While in chase mode, a ruler is dragged across the paper from left to right, advancing by two tiles each turn. The following extra rules become active:

- If the player is not hidden and caught by the ruler, they lose.
- If the player is on the same tile as an enemy, they lose.
- If the ruler reaches the end of the paper, chase mode is deactivated.
- Enemies always face right.

The player's performance is measured in multiple ways:

- How often chase mode was activated. More activations is worse.
- How many turns passed. More turns is worse.
- How many pickups were gathered. More pickups is better.

We have deliberately omitted the sound system from this prototype, as we could not figure out a consistent way to model the way sound would travel over the environment without slowing down turn times too much or confusing players too much. We also felt that the inclusion of the sound system would overly complicate our already complex paper model.

2.2. Playing Experience

The game is rather slow to play and it is pretty easy to get caught in the enemies' vision. The movement itself isn't very fun, which isn't surprising as it barely resembles the feeling of a real platformer. Making decisions on where to go based on where the enemies are is slightly interesting however, so at least that part of the stealth mechanic works on paper.

Needing pieces that can be physically moved by the player also requires very large tile sizes, ideally even larger than what we ended up doing. This however means that paper bigger than A4 would need to be used, or the sections of the levels would have to be made even smaller. As it is at the moment, moving the pieces around is a bit cumbersome and impairs the vision of the level.

2.3. Findings and Conclusion

Creating the prototype we noticed multiple things. First, modelling a platformer on paper is very hard to do, and extremely hard to make actually fun. A lot of the enjoyment of a platformer comes from the continuous feedback and interaction with the environment. However, despite this it helped us realise a few things that are useful for the actual game too:

Large levels, while giving designers more options overall, also make it much harder to create an effective experience. Splitting the level up into chunks instead allows focusing on one or two goals for each chunk while still allowing the designer to create a cohesive whole. This division should also allow players to not feel overwhelmed if the level has many different sections.

While none of the sizing constraints of the paper prototype exist digitally, we will have to tune the size of the viewport in playtesting, as having too much or too little on screen could impact the stealth mechanics very heavily.

The primary engagement seems to come from exploration, rather than from a more puzzle perspective as is usual in other stealth titles. This might be a result of our paper model, but at the moment it does seem likely that we'll need to experiment with our design further to ensure players are also occasionally encouraged to think about a situation, rather than taking everything as a platforming challenge. The sound system that we omitted from this prototype should help with that.

3

Interim Report

3.1. Progress

We have managed to complete almost all of our core game components, including several extras. To elaborate, we have fully implemented:

- **Collision Detection** We have working AABB swept collision detection against arbitrarily sized entities, and against a rigid grid tile map, including appropriate collision response. There are both completely rigid walls, jump-through platforms, and walls that only affect patrolling enemy movement.
- **Sprite Animation Rendering** Our engine includes a sprite animation system that lets you specify animations in a horizontal sprite strip as a pair of start and end frame indexes, duration in real-time seconds, and options to loop and change animations on animation end.
- **Tilemap Rendering** The method we use for rendering tilemaps onto the screen allows us to create arbitrarily large levels without any performance hit. This is achieved by encoding tile information in a texture and using a pixel shader to render the appropriate tiles onto a fullscreen quad.
- **Player Movement** Player movement allows satisfying running, jumping, climbing, wall jumping, and hiding mechanics. The player and other entities interact with the "layer zero" tilemap, which uses specific colours to designate specific types of environment.
- **Realistic Viewcones** Every enemy type (static camera, pivoting camera, ground drone, aerial drone) has a viewcone that can dynamically change. The viewcones are also realistically occluded by surrounding terrain. See Figure 3.2.
- **Level Loading** We devised a simple level file format to save and load levels from. See

Figure 3.1.: The animation tilesheet for the player.

section A.1.

- **Level Editor** In order to speed up level production and collaboration, we developed an advanced level editor that allows producing complete levels. The level editor can be used directly from a browser and is thus platform independent. See Figure 3.3.
- **Enemy AI** Our enemy AI includes simple ground enemy patrols, and aerial drones patrolling a base region. On detection, aerial drones in the same chunk as the player will move towards the last detected region and start searching for the player in the vicinity. When an aerial drone sees the player themselves, they will start chasing the player directly. Upon touching an enemy, the player dies and the level resets.
- **Player Animations** We have designed a player character and hand-crafted a full set of pixel animations for every possible movement action and interaction. See Figure 3.1
- **Positional Sound Playback** We have also implemented a custom positional sound engine with various tunable parameters, such as panning range, attenuation factor, and attenuation approximation function.

We have also partially implemented:

- **Level Tileset** The current tileset is usable for simple levels, but for nicely looking and varied levels, we will need to add a lot more tiles and variety, especially in decorations and background/foreground elements.
- **User Interface** While we have a system in place to produce simple textual menus, we have yet to create a dialog system or options menu.

3.2. Challenges

The most difficult aspects so far have been precision issues with the collision detection and getting realistic viewcones that are correctly occluded by the environment. The amount of detail required to craft aesthetically pleasing levels also required the creation of a complex level editor, which took up a lot of time.

Otherwise we experienced several minor issues and limitations concerning the Monogame framework, especially related to rendering. These issues mostly arose due to a lack of proper documentation, and/or exposure of more direct hardware feature access. So far we have managed to solve or work around these issues, though they massively increased the amount of time taken to implement certain features, especially the tilemap rendering.

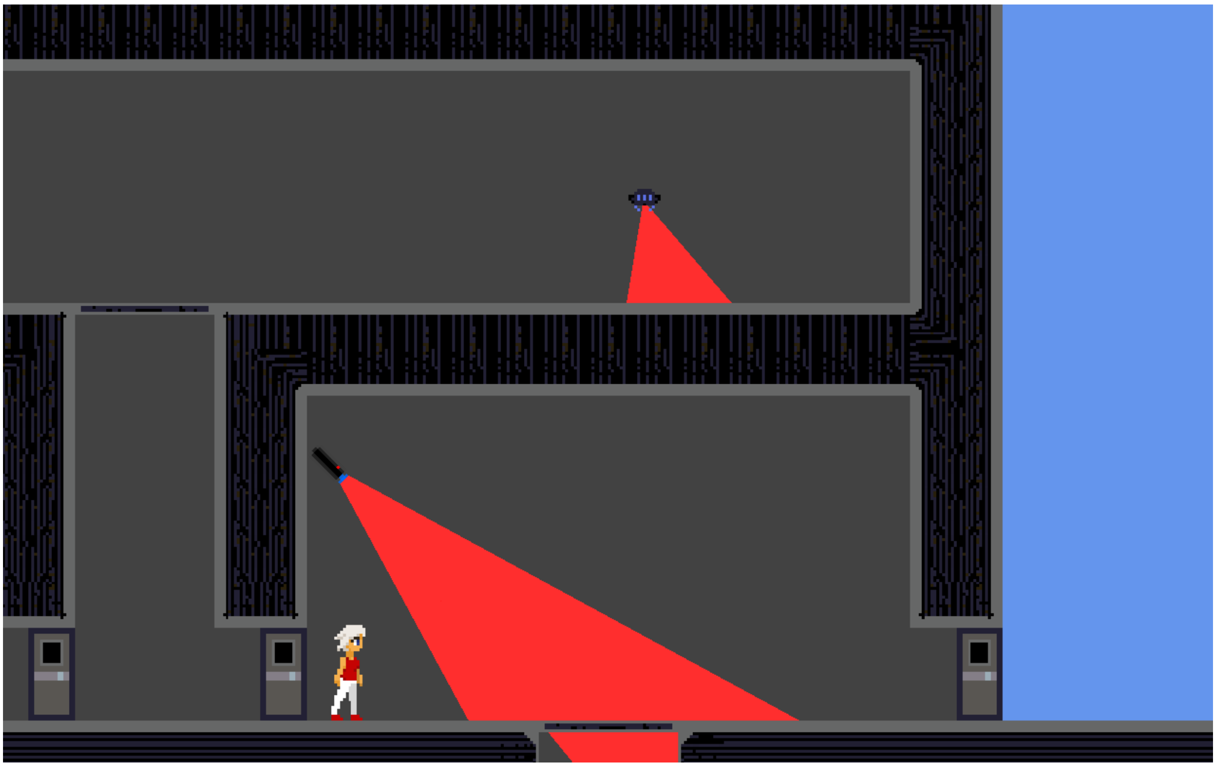


Figure 3.2.: Viewcones are realistically occluded by level geometry.

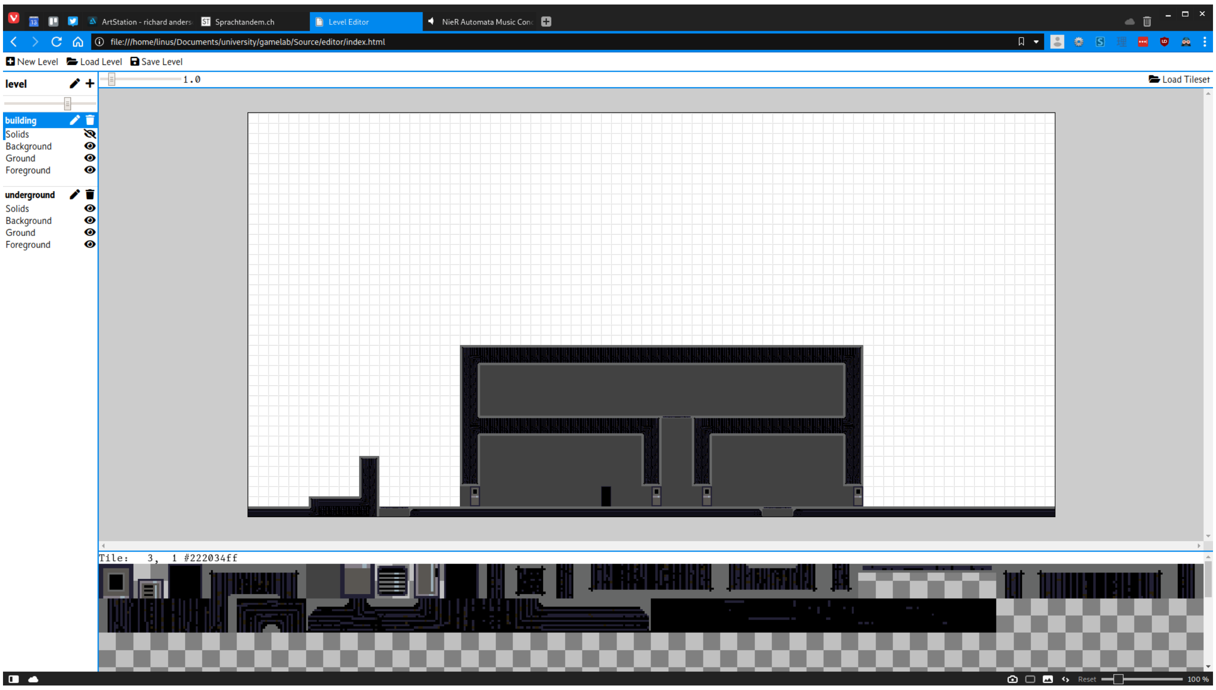


Figure 3.3.: The custom level editor allows convenient authoring of full levels, including story.

3.3. Future Work

For the following few weeks we should focus on the following aspects:

- **Positional Audio AI** We would like enemies to react to sounds that appear nearby. When hearing something suspicious, they should go and investigate the area. This requires the system to “know” how severe a sound is, and properly occlude sounds based on distance and the intermediary environment.
- **Sound Effects and Design** We will need specific, recognisable, and pleasing sounds for a variety of player and enemy interactions.
- **Level Tilesets** The current tileset is minimal and needs to be expanded to allow creating more detailed levels.
- **Story Dialogue** In order to deliver story, we need a dialogue system that is used for delivery.
- **Story Writing** The story also needs to be written of course.
- **Level Design** To complete the game, we’ll require at least one fully-fledged level, ideally multiple ones.
- **Fine-Tuning Parameters** Several parameters about the sound, AI, movement, and level design will need to be fine-tuned based on player experience and survey.

4

Alpha Release

4.1. Progress

As we had already completed most gameplay aspects by the interim report, we mostly worked on fixing bugs and implementing bonus features.

- **Enemy Hearing** Enemies are now able to hear sounds made by the player. The sound is attenuated over distance and dampened by walls. When an aerial drone hears a sound, it will go to the approximate source location of the sound to investigate.
- **User Interface** A full game UI is in place, including main menu, options screen, level loader, pause menu, and score screen. We did this using the UWP XAML framework.
- **Sideloading** Thanks to our compact level format we were able to add the ability to side-load levels from the internet. Meaning the game can easily download a level from a URL and play it.
- **More Environment Art** There are now also two additional environment tilesheets, one for an outside rooftops area, and one for an interior office.
- **Dialogue** We also implemented a dialogue system allowing us to deliver written story. Dialogue can be triggered at set points in a level, be played when a pickup is collected, or when the player decides to initiate a conversation themselves.
- **Custom Sounds** Custom sound effects have been added to accentuate the gameplay.
- **Music** Two custom music tracks were added that provide ambience.

4.2. Challenges

Aside from the typical programming challenges presented by bugs and the fixing thereof, we encountered two major issues that we were unable to resolve.

4.2.1. Chunk Transitions

When the player transitions between two chunks, the camera needs to transition as well. Typically the camera is clamped within the chunk bounds as to not show any untiled areas. However, during chunk transition, the camera might sweep over regions outside of the two chunks being transitioned between. We were unable to implement an apt solution to this problem. We tried fixing the camera to only move horizontally or vertically. However, this leads to awkward pauses and long transitions while the camera aligns with the other chunk before finally being able to move over. There are two solutions we are aware of, but which we did not attempt to implement due to time constraints: 1) Using a fading effect instead of a camera sweep when the game detects an unclean transition 2) extending the chunk bounds outside of what the camera shows when fully within a chunk. This would allow the level designers to fill in empty space that's only visible during transitions.

4.2.2. Monogame Limitations

The Monogame framework we are required to use has several limitations that we had to work around, especially regarding sound. The 3D sound engine in use does not expose ways to change the sound attenuation properties. There is no way to perform sound effects like reverb, or other custom DSP tasks. In order to have more control over the sound, we implemented our own positional sound system with various attenuation functions to choose from. However, there does not seem to be any way to hook into the sound sample processing pipeline at all, so implementing custom effects remains impossible.

The music player system is even more limited, only allowing you to play one track at a time and not allowing you to seek within a track to arbitrary positions. This means that we cannot crossfade between tracks, nor can implement smooth looping on tracks that have an intro. There also appears to be a strange bug wherein a song can only be paused and resumed once, after which pausing does nothing.

4.2.3. XAML Weirdness

The Microsoft XAML UI framework consists of a combination of a widget UI class framework and an XML description language. The set of widgets offered to the designer is very limited, especially under UWP. Several options, functions, and widgets are simply not available under UWP. For instance, setting the scaling filter for images is not possible under UWP, which is annoying for pixel-perfect graphics like what we're doing.

The XML description language is extremely cumbersome to use and especially so for styling. Here, too, several limitations presented by UWP prevented us from implementing certain customisations, such as changing the look of a UI element when the element is focused.

The information available on the net about XAML is also confusing, as several resources speak of features and solutions that are no longer present, or again prevented by UWP restrictions.

4.3. Future Work

Things that we consider interesting additions to the game that might be worth exploring are:

- **Better Ground AI** The current ground drone AI is very simplistic and only moves left to right with brief stops in between. If it could traverse certain obstacles, it would already be a lot more interesting.
- **Switches** Currently there's no way to lock level content behind keys or switches. Adding this would allow a minor puzzle aspect and encourage exploration some more.
- **Ambient Sound** As it is, only the player and the enemies generate noise. The feel of the game could be improved greatly if other parts of the level such as vents, computers, etc. would also generate ambient noises.
- **Distractions** The player strategy could be expanded greatly if there was a way to distract enemies, say by throwing rocks to cause noise, to cause alarms on a fake location, or to cloud enemy vision with gas or similar.
- **Patrol Routes** The aerial drone AI simply performs a random walk within a constrained region when idle. The tactical aspect of the game could be improved greatly if drones could be given specific patrol routes to follow instead.
- **Human Guards** Having human guards with the same moveset as the player would present a much more interesting opponent, but would also be very challenging to program from an AI perspective.
- **Dynamic Music** If the music tracks were composed correctly they could be seamlessly cross-faded from idle music to alert music and back, aiding immersion.

5

Playtest

5.1. Playtesting Session

We individually recruited friends to do a playtest sesion at home. Each session was organised into four parts:

1. **Introduction:** First the player was briefly introduced to the project with a short outline on how the playtest would work and what this is about.
2. **Survey:** Then they were asked to fill out a basic questionnaire to establish their demographic, gaming background, and testing expectations.
3. **Playing:** After that they were asked to play through the game, while a team member would take notes of their progress and behaviour during play.
4. **Evaluation:** Once the game had been completed, the player was once again asked to fill out a questionnaire, this time on their experience playing the game.

See the next section for a breakdown of the questionnaires and the evaluation thereof.

5.2. Questions and Comments

The questionnaires were conducted using Google Forms.

5.2.1. Survey

The survey included the following questions:

- Name
- Age
- How often do you play?
- Which controls are you most familiar with?
- What are some of your favourite games?
- Based on your favourites, what do you think attracts you to games?
- What would you like to see in a game made just for you?
- How complete do you expect the game to be?
- How fun do you expect the game to be?

5.2.2. Evaluation

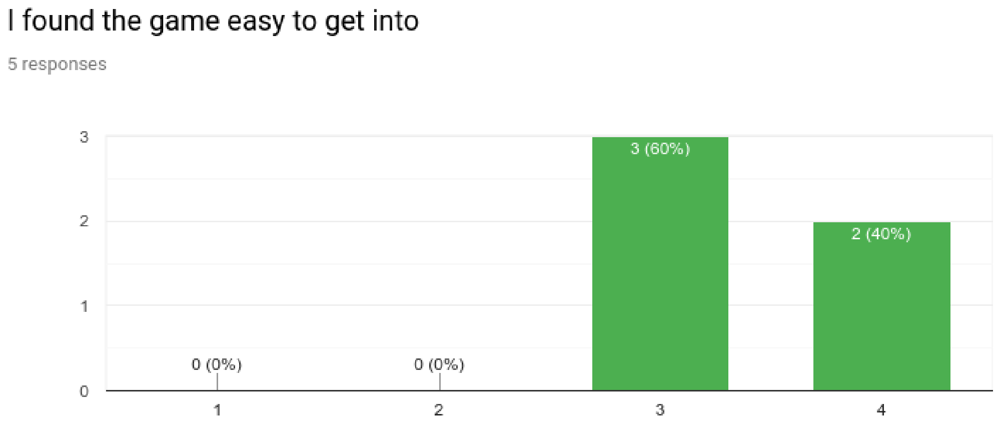
The survey included the following prompts and questions:

- I found the game easy to get into
- I found the game awkward to play
- I felt confused a lot while playing
- I felt confident while playing
- I felt immersed in the game
- I was taken out of the game a lot
- The game was too difficult
- The game did not challenge me enough
- What part of the game did you enjoy the most?
- What needs to be changed to make the game more enjoyable for you?
- What was the most difficult part of the game for you?
- How complete did you find the game to be?
- How fun did you find the game to be?

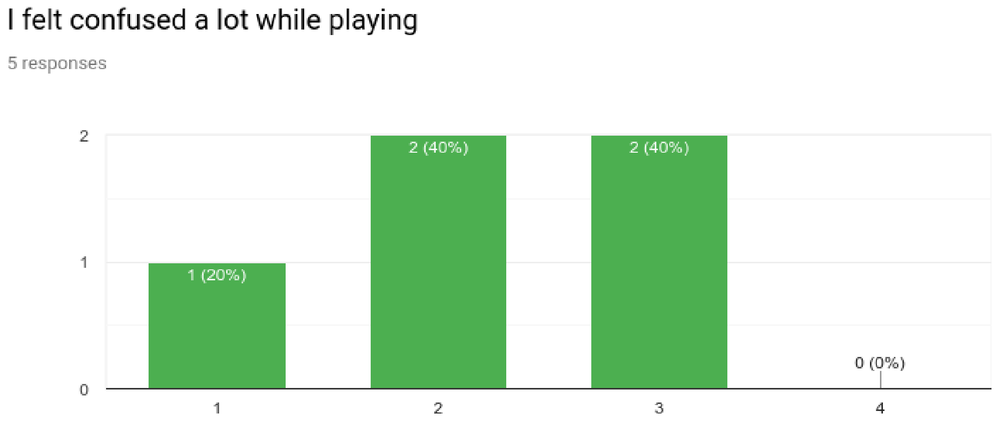
Each of the prompts was answered with a scale of 1-4, with 1 being “Strongly Disagree” and 4 being “Strongly Agree”.

5.2.3. Results

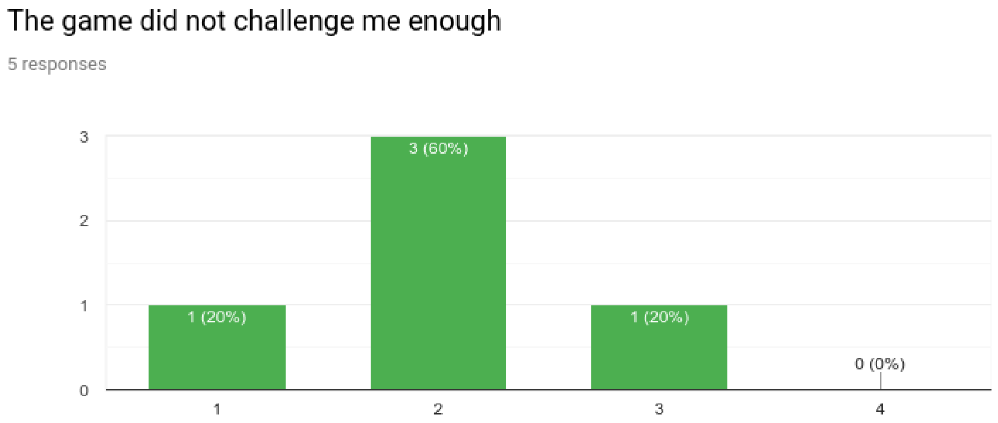
The full results of the surveys as well as the manual notes taken can be found in section A.2. We list a few of the more interesting results here:



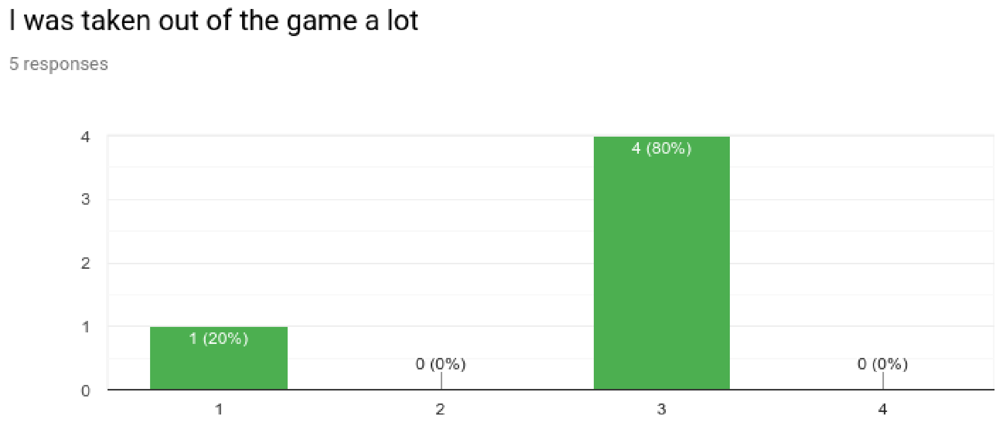
The players reported that they found the game easy to pick up and get going with it, though there were some issues related to the sneaking mechanic that some found confusing, particularly pressing in the left stick to sneak was noted as an awkward control choice.



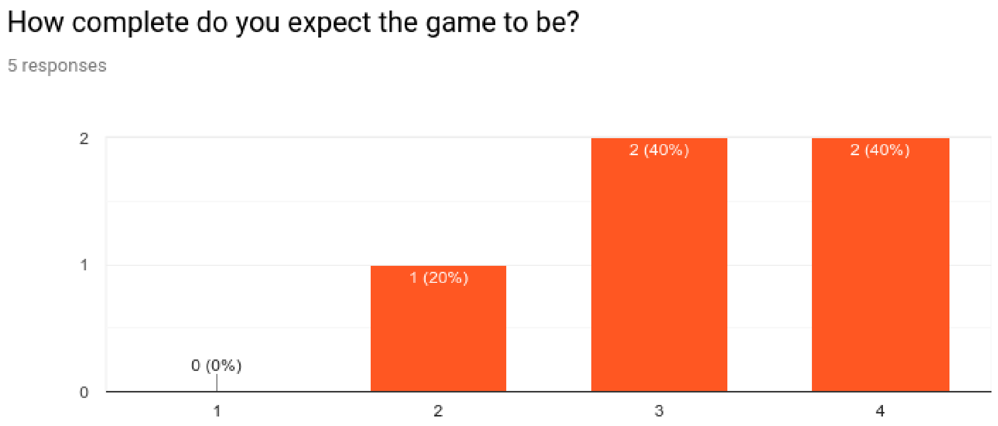
All players reported confusion during some levels, especially the Factory one. In order to fix this, the levels would have to be redone in a more streamlined manner, with clearer player guidance.

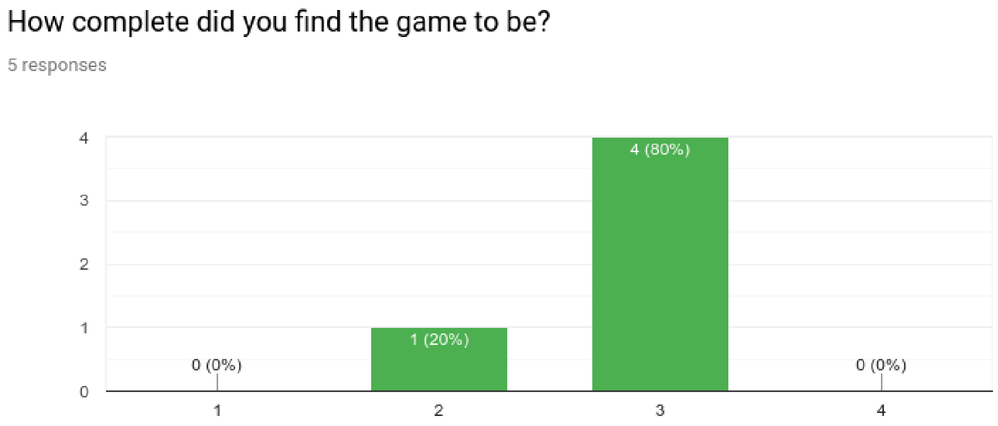


While one player reported the game to be too difficult, most players noted that the game was appropriately scaled in challenge.



Players mostly reported that they felt immersed in the game, but were also taken out of it a lot. During the playtesting this mostly occurred due to frustration with having to wait in the hiding spots for a long time and due to the sometimes confusing level design.





The player's evaluation of the completeness of the game shifted downwards after playing, due to the lacklustre level design and lack of polish.

5.3. Design Revisions

We mostly had to clean up some issues in the level design, particularly the Factory, but a completely smooth experience would require redoing all of the levels entirely, and expanding on the story, for which we did not have time.

We also discovered a few bugs that were introduced by last-minute changes and reduced the alert countdown when the player is in hiding, as most playtesters complained about it being too long.

6

Conclusion

6.1. Final Results

We were able to deliver on our intended goals and did not deviate far from the initial plans aside from cutting down on a few features that would have taken up too much time to implement. The most significant change would be the usage of pathfinding aerial drones instead of a simple wall of death to implement the chase mechanics, as seen in Figure 6.1. This gives rise to a more classical stealth game experience.

Unlike initially assumed, we were able to implement a few additional features as well, such as a fully-fledged level editor, a full in-game menu, and background music tracks. The editor in particular, as seen in Figure 6.2, can be used in any browser and does not require any other parts of the game. It reads and writes to a specified format, and should thus be easily adaptable for other games.

The final game delivers on the crucial features for a fully-fledged game, though definitely needs more work in terms of level design, story writing, and polish.

6.2. Experience

Our initial game idea translated pretty well, though with as little actual game data as we could gather, it's hard to say whether it would be possible to expand it to the length of multiple hours. The initial scheduling was quickly abandoned and we started managing work ad-hoc between ourselves as need demanded. After the initial base systems such as animation, tile rendering, and collision detection were in, we then organised tasks and todos using a Google sheet, adding and ticking off things over time. We also kept weekly meetings, and the main contributors actively partook in Slack chat to exchange ideas and discuss the project.

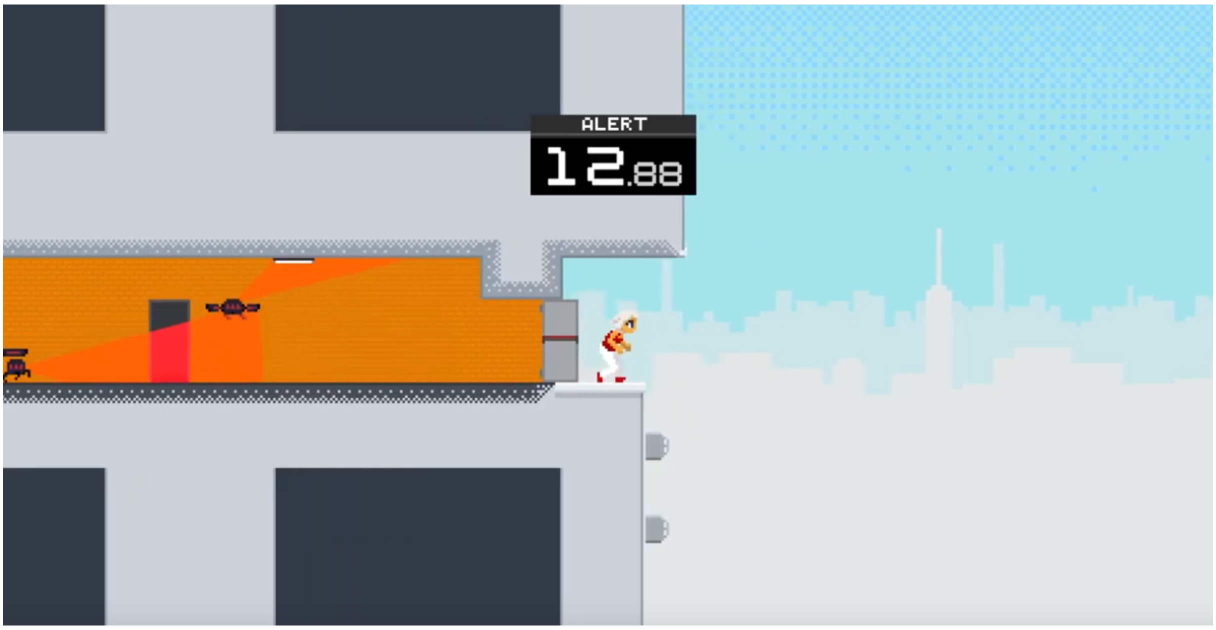


Figure 6.1.: Getting seen by enemies will cause an alert during which drones will actively seek you out and chase you down.

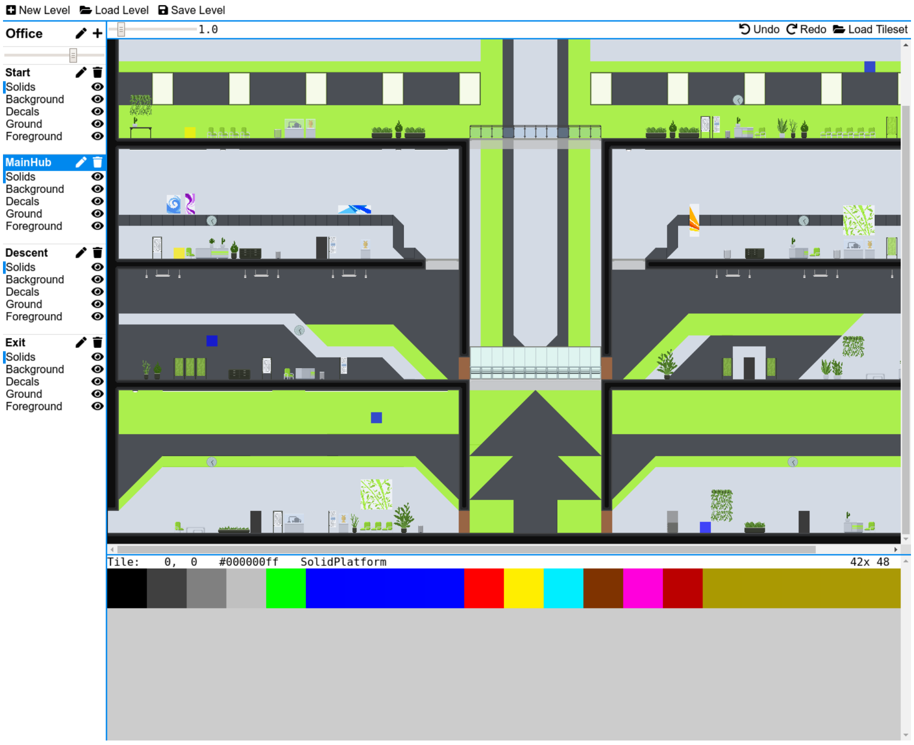
We still feel that the paper prototype as it was done did not help us much and our time would have been better spent implementing the base gameplay systems, so that we would have had more time to polish the game and work out gameplay and level design issues. The gameplay session was also way too short-term and we were not able to privately test the game enough or get in enough people to get a lot of useful feedback that we did not already discover ourselves.

Difficulties in terms of programming mostly lied in getting the collision and enemy AI to work perfectly, which led to many bugs and revisions over the course of the development. Almost all of the credit for the work on that goes to Gengyan Li.

The biggest hindrance by far was the communication between teammates and the lack of coordinated academic schedules to allow this project to flourish. Team members were often heavily burdened by other tasks such as workshops or time-intensive exercises from other courses.

6.3. Personal Impressions

This project was a very stressful and hugely time-intensive endeavour that would require at least twice the amount of credit points to properly justify the amount of work that went into producing a presentable game. While the joint operation between ETH and ZHdK for this course is a welcome idea, it is executed rather poorly. The huge discrepancy in credit points means ZHdK students are given far less time to contribute to the project over other subjects. The lack of schedule coordination also meant that ZHdK students were absent for large spans of time due to workshops or other obligations. As students are thrown together into groups rather abruptly without knowing each other at all there is also a very significant challenge in



```
Tile:    0,  0    #000000ff    SolidPlatform                                42x 48
```

Figure 6.2.: The office level in the separate level editor. See Figure 6.3 for an in-game screenshot.

managing and coordinating the team, which further adds on to the already massive amounts of time commitment necessary for this course.

Being restricted to the Windows 10 platform and the Monogame framework also proved rather annoying. A lot of time was wasted trying to figure out how to coerce Monogame into doing what's required and debugging Monogame, Windows 10, or Xbox related issues, to which there was often very scarce amounts of information to be found online, if any at all. The Windows 10 and development environment setup restrictions further burdened our team and led to the fact that the ZHdK students were not able to test the game at all outside of the weekly meetings. This was extremely troublesome during the level design phase, as there was no direct way for them to get feedback on the playability as they were developing the levels.

If I were to make another game in a group, I would definitely seek out people I've already known for a good while and commit to the project as a 100% job. Otherwise the amount of stress produced and time wasted that builds up would hamper the progress and quality of the final result far too much. I would also not choose a framework that is locked into the Windows 10 ecosystem, or target the unpopular Xbox One console. Depending on the project goals, I would either go for a fully fledged commercial engine like Unity or Unreal, or use a fully home-grown engine based on OpenGL and other platform-portable technologies.

Overall I am not quite happy with the final result, though that is mostly due to extremely high standards on my part. For the vast amounts of hindrances and restrictions that were placed on this project, I would say that we did surprisingly well. The game looks like something that could be finished, if it were expanded upon and the existing content polished further. The theme,

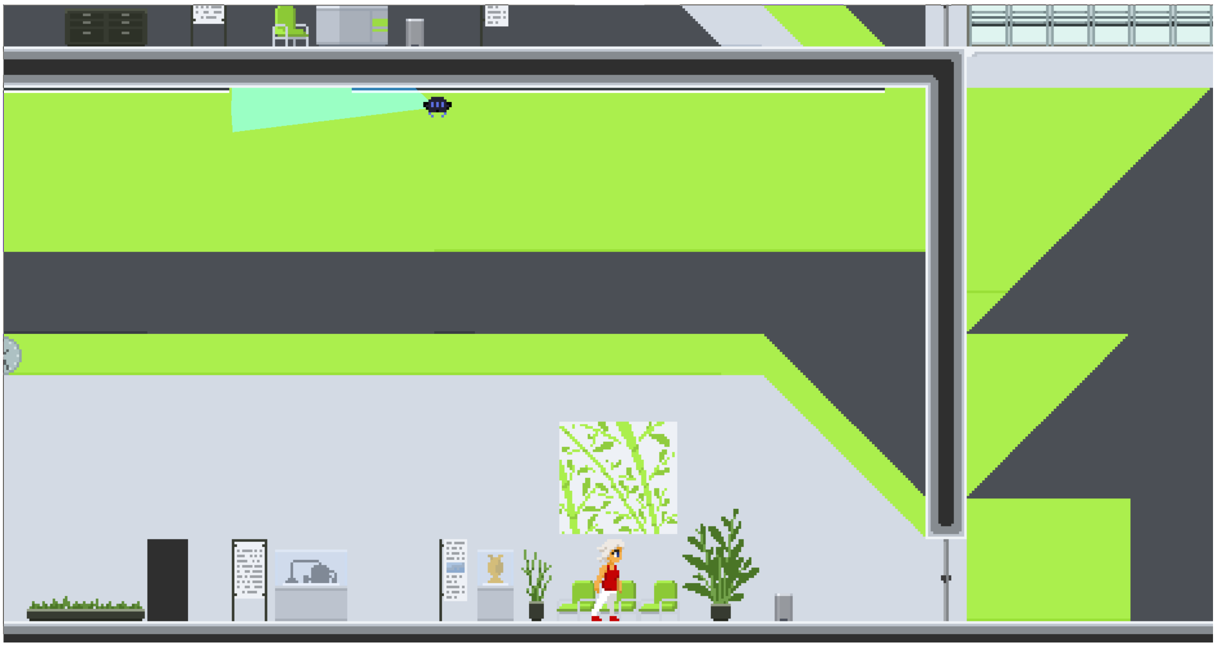


Figure 6.3.: An in-game snapshot of the office level

to us, was mostly a side-thought that spawned the initial story, but it did not inspire or impact the game design by much, as the same game could have been implemented under many other themes by swapping out the art design and story.

The course could be significantly improved if the collaboration with ZHdK was either removed entirely, or revamped significantly to match up the amount of credit points awarded and to synchronise the student schedules better as to not collide so frequently. The idea of building up a game from scratch is great, but then the course should probably be renamed to *Game Development*, rather than *Game Programming*. The latter implies less of a full game development and more of a hands-on teaching of game and engine programming techniques. That, too, could be an interesting and exciting course, but as it stands the title of this course seems like a misnomer.

The lectures presented in the course felt mostly superfluous and only contained sparsely useful information. The content could be improved radically by focusing more on case studies of existing games and the critical analysis of gameplay, art, and presentation. The current lectures often felt aimless and factually dry. By focusing on an active discussion about existing games and their systems, the points could be delivered in a more tangible manner and could be livened up by connecting the information with games the students might already care about, or might want to investigate on their own after seeing it in the course. Being able to see a system work or fail in action would also be a far more effective way of demonstrating why or how these points are actually important. For this reason, the presentation by the Studio Gobo member was the most insightful and interesting.

Game development has been my passion all my life and I am thankful for the chance of being able to work in this area as part of a university class. However, there are a lot of things that could be done to improve the experience of this class for students, especially for those that do not already have prior experience working in game development.

A

Appendix

A.1. Level File Format Specification

The level format specification follows, formatted in Markless.

```
&# Level File Specification
This is a specification for a level data package. A level is a ZIP
 ↪  file with a set file structure. You can find an example at the
 ↪  end of the document.

## ZIP Contents
The only requirement about the zip file is that it must contain a
 ↪  `level.json` file at its root with the data structure conforming
 ↪  to the specification in the following section.

## Data File Format
This is a specification for the level data file format. The
 ↪  specification declares the structure in abstract terms of types,
 ↪  including structure types, and is thus not dependent on any
 ↪  particular interchange format. However, for the purposes of this
 ↪  game the format should be assumed to be **JSON**, encoded in
 ↪  **UTF-8** when serialised to binary.

### Specification Syntax
We employ a special syntax to denote the format.

:: BNF
Specification  ::= TypeDefinition*
TypeDefinition ::= name ':' Type '\n'
```

```
Type           ::= Structure | Array | Union | name | baseType
Structure      ::= '{' Entry (',' Entry)* '}'
Array          ::= '[' Type (' ' integer)? ']'
Union          ::= Type '|' Type
Entry          ::=  name ':' Type
baseType       -- One of the basic types noted in the following
 ↪  section
name           -- A name for an entry or type, alphanumerics only
integer        -- An integer larger than zero
::

White space may be used in between tokens as desired.

### Base Types
The specification assumes the following types to be present:

- //String// A sequence of arbitrary characters.
- //Integer// A numerical integer.
- //Float// A decimal number of limited precision.
- //Boolean// Either ``true`` or ``false``.
- //Null// An empty value.

### Data Validity
In order for the data to be valid, it must abide the following rules:

- The data must fit the ``Level`` type.
- Data specified to be a structure must contain all the defined
 ↪  fields and no more.
- Each field's value must be of the associated type.
- Data specified to be an array must only contain entries of the
 ↪  associated type.
- If the array type contains an integer, data specified to be of that
 ↪  array type must contain exactly as many entries.
- If the type is a union, the value must be of one of the two types
 ↪  of the union.

An implementation must signal an error if the data to be processed is
 ↪  not valid.

### Level Data Structure
We will now describe the level file format using the established
 ↪  conventions.

::
Level: {
  name: String,
  description: String,
  preview: Null | Path,
  next: Null | String
  startChase: Boolean,
  startChunk: Integer,
  storyItems: [ [ String ] ],
  chunks: [ Chunk ]
}

Chunk: {
  name: String,
  position: Vector,
  outside: Boolean,
  tileset: String,
  background: Null | String,
  storyItems: [ [ String ] ],
  layers: [ Path ]
}

Path: String
Vector: [ Float 2 ]
::

## Example ZIP
The following are the file records for an example level ZIP:

::
level.json
preview.png
chunks/first-0.png
chunks/first-1.png
chunks/first-2.png
chunks/first-3.png
::

And the following would be the JSON encoded contents of the
 ↪  ``level.json`` file:

:: JSON
{
  "name": "Example",
  "description": "An example level for the specification.",
  "preview": "preview.png",
  "next": "other-example",
  "startChase": false,
  "startChunk": 0,
  "storyItems": [[".. Wrong number."]],
  "chunks": [
    {
      "name": "first",
      "position": [0, 0],
      "outside": false,
      "layers": [
        "chunks/first-0.png",
        "chunks/first-1.png",
        "chunks/first-2.png",
        "chunks/first-3.png"
      ],
      "tileset": "sampleset",
      "background": null,
      "storyItems": [["Hello there, welcome to the game!"]]
    }
  ]
}
::
```

A.2. Playtest Results

A.2.1. Tester 1

```
Date: 2019.05.08
Play time: ~45m

- The player remarked multiple times about the lack of polish
 ↪  in the office level, though that should be fixed now
- The player got stuck and confused a lot in the factory level,
 ↪  due to a ton of enemies and sometimes confusing tiling,
 ↪  though that too should be fixed now I hope.
- Some of the story beats got a chuckle out of them, but others
 ↪  fell flat.
- Overall the player seemed to get caught too often, which
 ↪  means it might be too punishing to get caught.
- The ending did not work as intended. The call trigger runs
 ↪  the animation, but the player falls into the goal before
 ↪  the call actually happens. We need to make sure that goal
 ↪  collisions are ignored when the player is in the calling
 ↪  state.
- The final chase sequence had not worked as intended either,
 ↪  as the alarm didn't continue and the player actually never
 ↪  encountered the horde of drones overhead as they moved
 ↪  ahead too quickly

Play session: https://www.youtube.com/watch?v=8c325nfTl0M
```

A.2.2. Tester 2

```
Date: 2019.05.09
Play time: ~35m

- The hiding spot in the office section after the unpassable
 ↪  camera is a bit hard to reach, and not strictly speaking
 ↪  necesary
- A few one way directions
- Alarms take very long, you spend too much time waiting in
 ↪  hiding spots
- Drone at the office exit can block the way for quite a long
 ↪  time
- Left stick is unintuitive and hard to use for crouching
- Had difficulty finding exit in office, main hub (possibly
 ↪  lower exit?)
- Drones are too sensitive to sound (will react from outside
 ↪  vision)
- Factory is broken/incomplete as fuck
- Cameras rotate a lot, reaching much farther and much faster
 ↪  than one might expect
- Factory floor is too wide when the doors are open, the ground
 ↪  drones then wander across the entire floor
- You can very easily drop through multiple platforms
- Viewcones obscure too much still (possibly lower the alpha)
- No parcour is absolutely necessary until the last level,
 ↪  where it gets a bit too hard due to lack of practice.
- Player Camera is not wide enough for possible strategy.

Play session: -
```

A.2.3. Tester 3

```
Date: 2018.05.19
Play time: 37m

-Main complaints was the alarm length as its too long so the
 ↪  better option is to die and restart rather than wait it out
-The direction of progress in the Factory level isnt clear
-the stealth mechanic went unnoticed most of the time
-the player complained about lack of options and would have
 ↪  liked some offensive ability
-The outro is a bit too hard and sudden esp when encountering
 ↪  the blue railings near befor the door
```

```
-Wall jumping wasnt used much and holding the climb to go up
 ↪  was much prefered
-Not much effort was made to collect the books

Play session: -
```

A.2.4. Tester 4

```
Date: 2019.05.10
Play time: ~30m

- Player quickly became conditioned to sneak almost constantly
- Player could not follow the story and started skipping dialog
- Office and especially Factory level proved very confusing
- The player noted several times that he didn't understand
 ↪  where he was supposed to go
- Player did not understand why he visited the locations and
 ↪  what he was supposed to do beyond sneaking around
- Player felt immensely frustrated during final chase and got
 ↪  stuck in the S hallway
- Player noted he would have liked the final sequence to be
 ↪  longer, but less punishing
- The player noted that the drones were too sensitive to sound

Play session: -
```

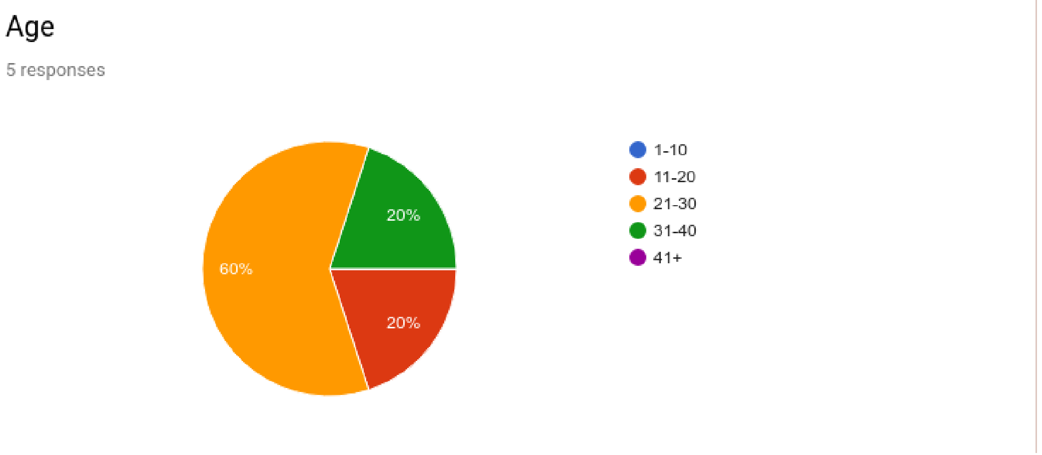
A.2.5. Tester 5

```
Date: 2019.05.19
Play time: ~45m

- Main complaints were S shape corridor in the outro and the
 ↪  length of the timer
- The exit of the Main hub in the office was frustrating to
 ↪  find
- Died multiple times in the Factory because of having no idea
 ↪  about how to proggres eventually finding the exit by
 ↪  continiously going to the right
- There is a dialog trigger in the Factory that triggers even
 ↪  while being chased that caused the player to die which was
 ↪  frustrating to the player.
- Didn't really understand the difference between the alarm at
 ↪  the outro since it was sudden and confusing
```

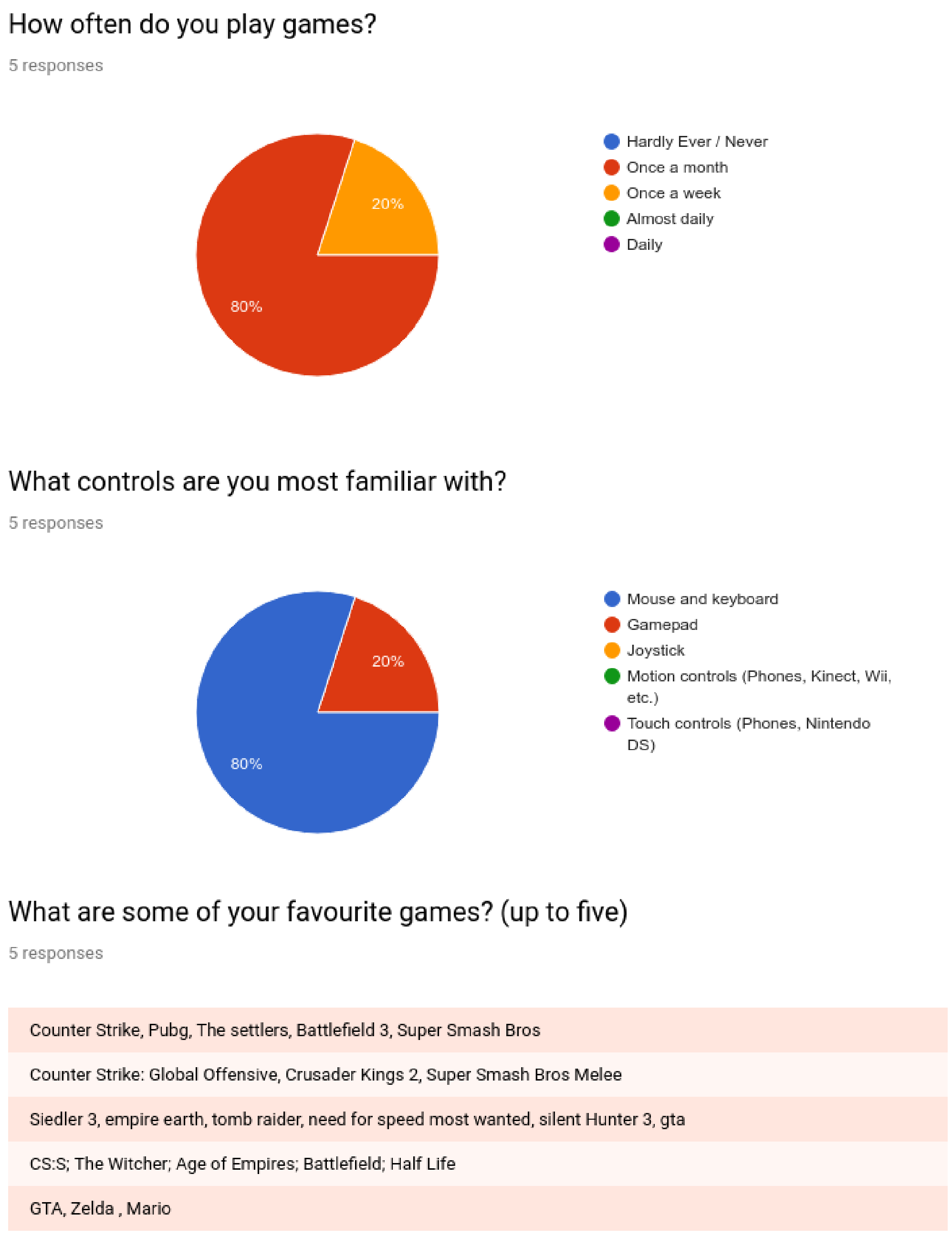
```
- Multiple pass throughs are very easy to die from by because
↪  the players holds the down key
- Got caught on average of 6-10 times per level (except the
↪  outro)
- Died on average of 10 times each level ( intro: 7 , office :
↪  10, Factory 15, outro : 9)
- If the player doesn't care about the story there is no
↪  incentive to pick up the story items

Play session: -
```

A.2.6. Survey responses

```
Timestamp,Name,Age,How often do you play games?,What controls are you most familiar with?,What are some of your favourite
↪  games? (up to five),"Based on your favourites, what do you think attracts you to games?",What would you like to see in a
↪  game made just for YOU?,How complete do you expect the game to be?,How fun do you expect the game to be?,I found the game
↪  easy to get into,I found the game awkward to play,I felt confused a lot while playing,I felt confident while playing,I
↪  felt immersed in the game,I was taken out of the game a lot,The game was too difficult,The game did not challenge me
↪  enough,What part of the game did you enjoy the most?,What needs to be changed to make the game more enjoyable for
↪  you?,What was the most difficult part of the game for you?,How complete did you find the game to be?,How fun did you find
↪  the game to be?
08/05/2019 15:13:09,Tester 1,21-30,Once a week,Mouse and keyboard,"Counter Strike, Pubg, The settlers, Battlefield 3, Super
↪  Smash Bros","Adrenaline, Action, Motorial Skills, Story, Multiplayer experience","Realistic nature, player freedom, open
↪  world",3,2,4,1,2,4,4,3,1,2,"(very smooth) jumping behaviour, wall grabbing","more sophisticated background textures,
↪  better ending, more books, books are somewhat unnatural placed",impulse suppression,2,2
09/05/2019 18:23:25,Tester 2,21-30,Once a month,Mouse and keyboard,"Counter Strike: Global Offensive, Crusader Kings 2, Super
↪  Smash Bros Melee",High skill ceiling,,4,4,4,2,2,4,3,1,2,3,Avoiding Robots and Parkour.,The different rooms could be more
↪  distinguishable from one another so as to not get lost. Also: the timer could go a bit faster when you're in a door so you
↪  don't have to wait 20s.,,3,3
10/05/2019 19:33:25,Tester 3,21-30,Once a month,Mouse and keyboard,"Siedler 3, empire earth, tomb raider, need for speed most
↪  wanted, silent Hunter 3, gta ","Open earth, freedom of life, Beautiful landscapes, water reflections, possiblities to do
↪  whatever you want ","Entrepreneurship simulation, open world, GTA for your hometown",4,2,3,1,1,3,2,3,2,2,The jumping ,Mehr
↪  action buttons and less waiting for being free again from the cameras ,Handling in the last level and the waiting ,3,2
10/05/2019 20:03:09,Tester 4,31-40,Once a month,Mouse and keyboard,CS:S; The Witcher; Age of Empires; Battlefield; Half
↪  Life,"Graphics, Fun, Joining another world",I want to feel myself in another world,2,3,3,1,3,3,4,3,2,2,variety,"clear
↪  directions, a way to kill the drones / Some drones, Stuff to collect which makes me more poweful / Fast / silent",the
↪  factory level - > I had no orientation where to go and what to do. i missed a waypoint,3,3
11/05/2019 19:24:10,Tester 5,11-20,Once a month,Gamepad,"GTA, Zelda , Mario","Adventure , variety",Big freedom of
↪  exploration,3,3,3,2,3,2,2,3,4,1,"Controls , presentation","Alarm, level design",Escaping drones when detected,3,2
```



A. Appendix



How often do you play games?

5 responses

What controls are you most familiar with?

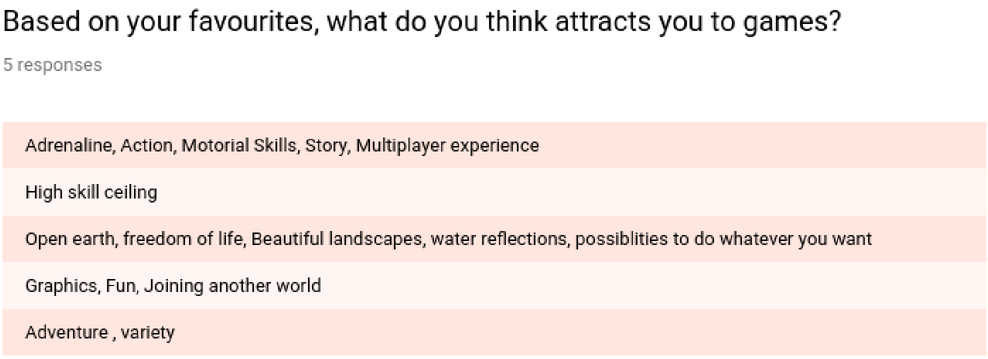
5 responses

20%
80%
● Mouse and keyboard
● Gamepad
● Joystick
● Motion controls (Phones, Kinect, Wii, etc.)
● Touch controls (Phones, Nintendo DS)

What are some of your favourite games? (up to five)

5 responses

Counter Strike, Pubg, The settlers, Battlefield 3, Super Smash Bros
Counter Strike: Global Offensive, Crusader Kings 2, Super Smash Bros Melee
Siedler 3, empire earth, tomb raider, need for speed most wanted, silent Hunter 3, gta
CS:S; The Witcher; Age of Empires; Battlefield; Half Life
GTA, Zelda , Mario



Based on your favourites, what do you think attracts you to games?

5 responses

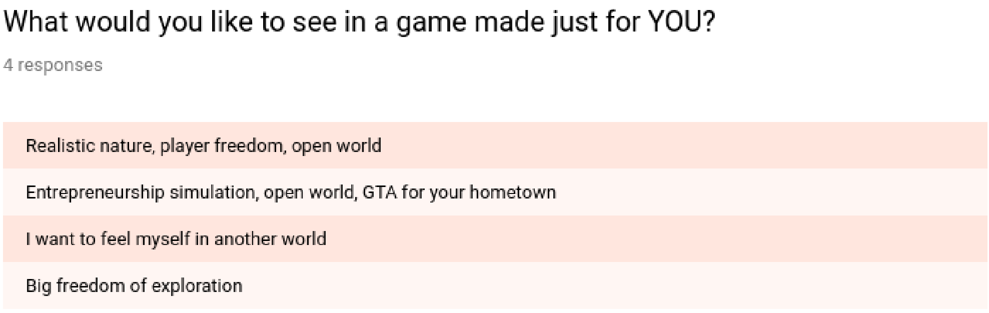
Adrenaline, Action, Motorial Skills, Story, Multiplayer experience

High skill ceiling

Open earth, freedom of life, Beautiful landscapes, water reflections, possiblities to do whatever you want

Graphics, Fun, Joining another world

Adventure , variety



What would you like to see in a game made just for YOU?

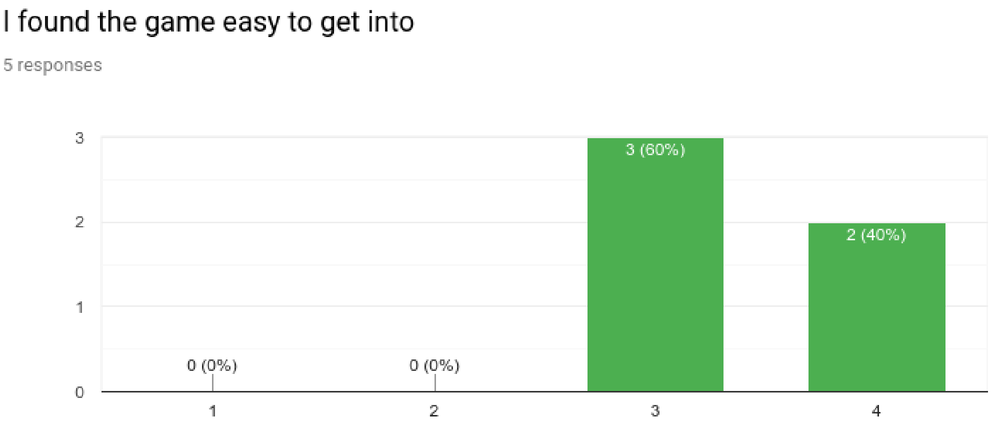
4 responses

Realistic nature, player freedom, open world

Entrepreneurship simulation, open world, GTA for your hometown

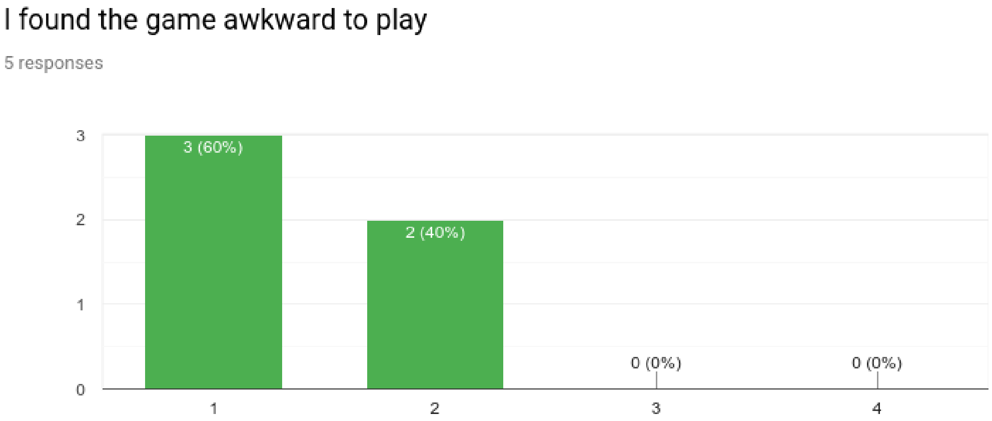
I want to feel myself in another world

Big freedom of exploration



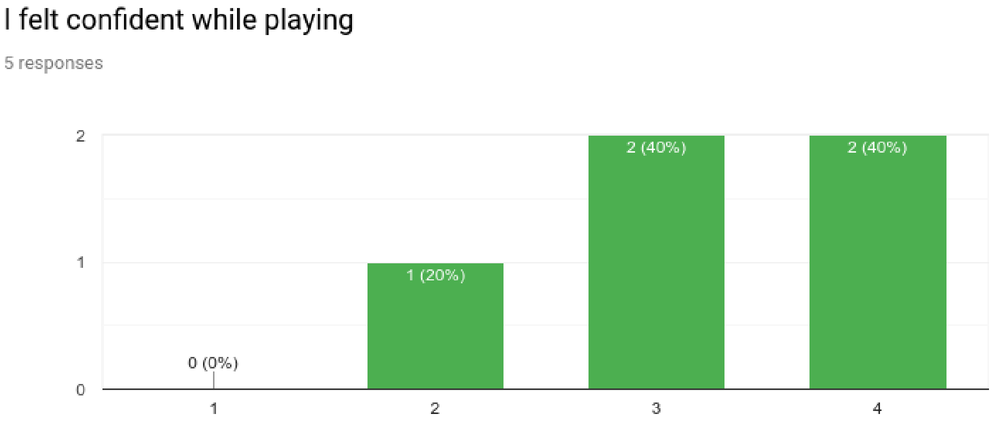
I found the game easy to get into

5 responses



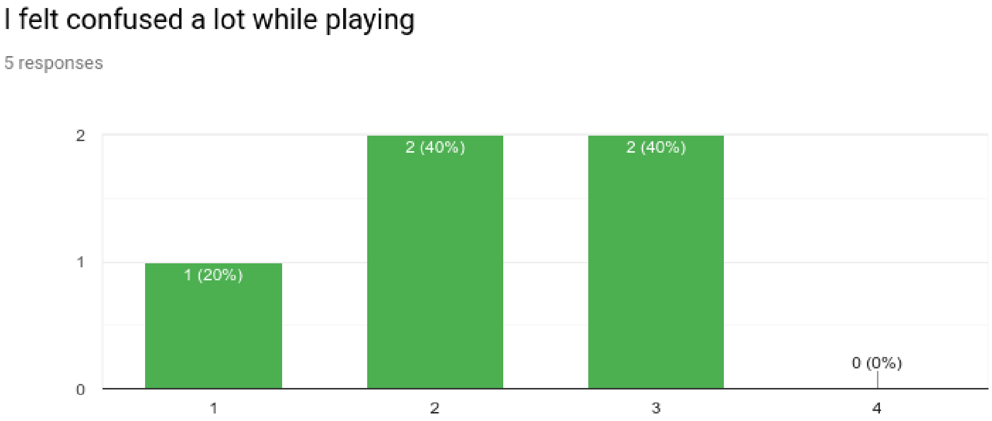
I found the game awkward to play

5 responses



I felt confident while playing

5 responses

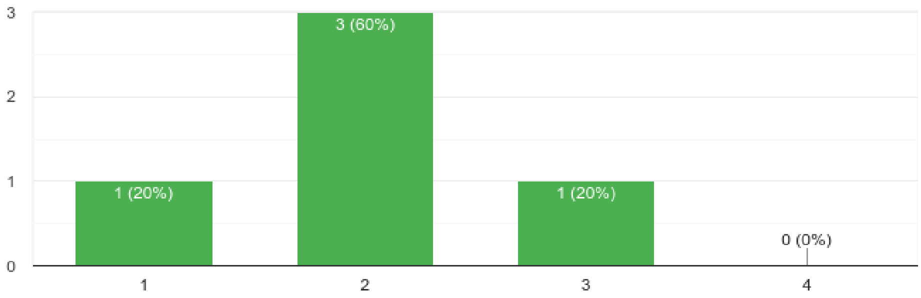


I felt confused a lot while playing

5 responses

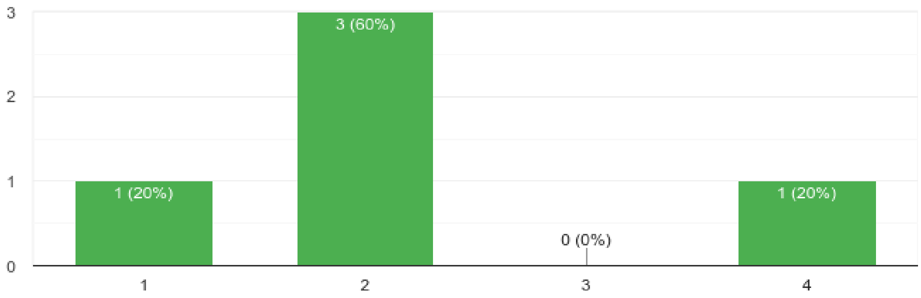
The game did not challenge me enough

5 responses



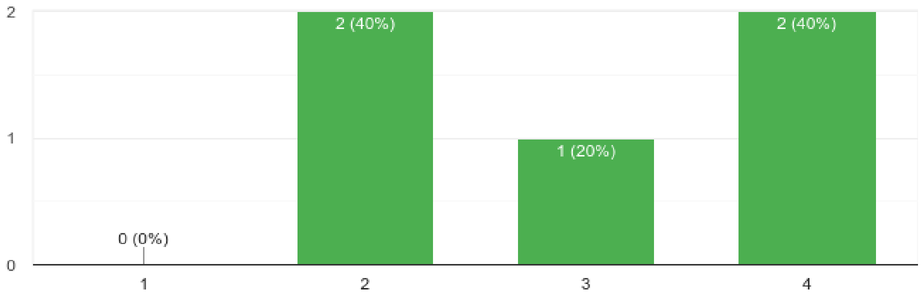
The game was too difficult

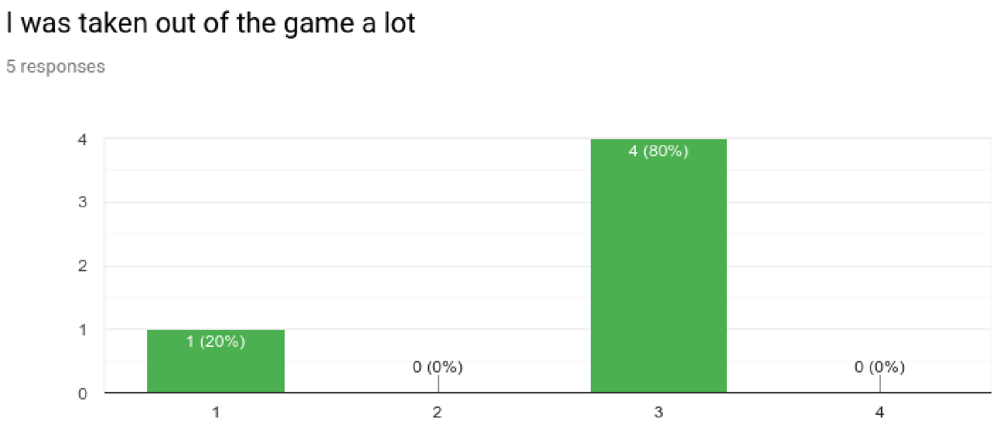
5 responses



I felt immersed in the game

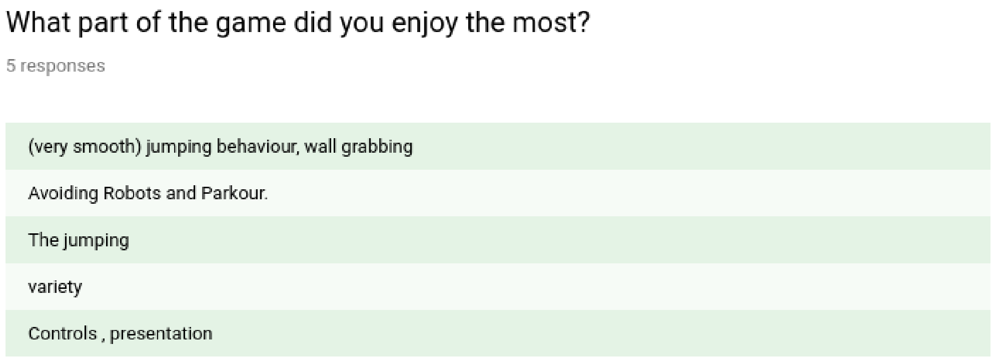
5 responses





I was taken out of the game a lot

5 responses



What part of the game did you enjoy the most?

5 responses

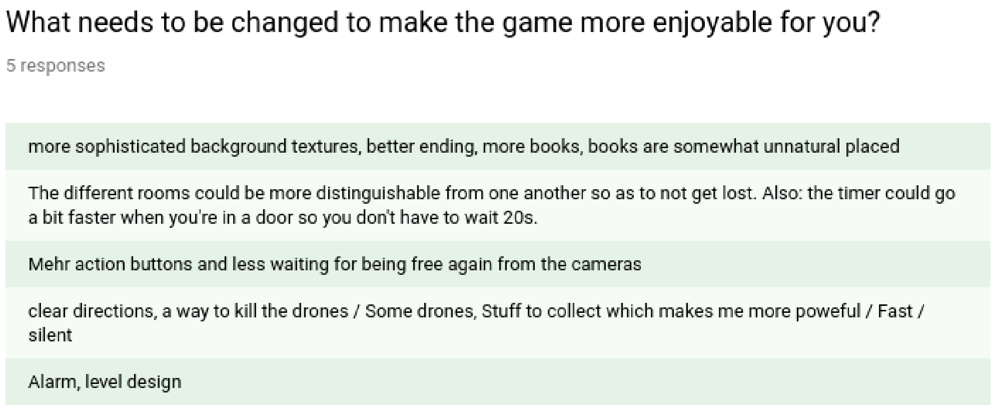
(very smooth) jumping behaviour, wall grabbing

Avoiding Robots and Parkour.

The jumping

variety

Controls , presentation



What needs to be changed to make the game more enjoyable for you?

5 responses

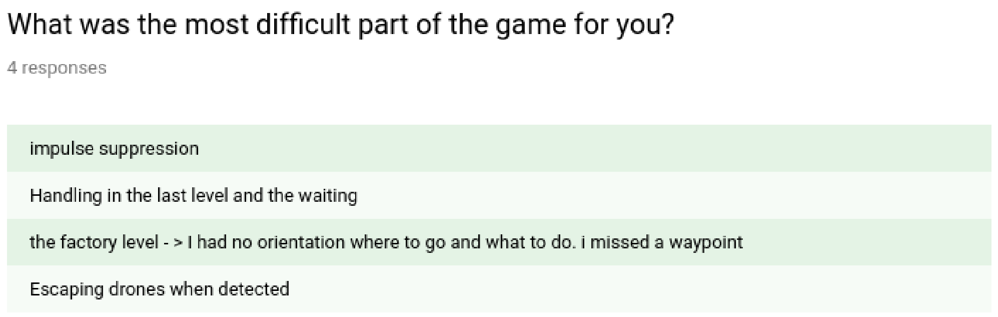
more sophisticated background textures, better ending, more books, books are somewhat unnatural placed

The different rooms could be more distinguishable from one another so as to not get lost. Also: the timer could go a bit faster when you're in a door so you don't have to wait 20s.

Mehr action buttons and less waiting for being free again from the cameras

clear directions, a way to kill the drones / Some drones, Stuff to collect which makes me more poweful / Fast / silent

Alarm, level design



What was the most difficult part of the game for you?

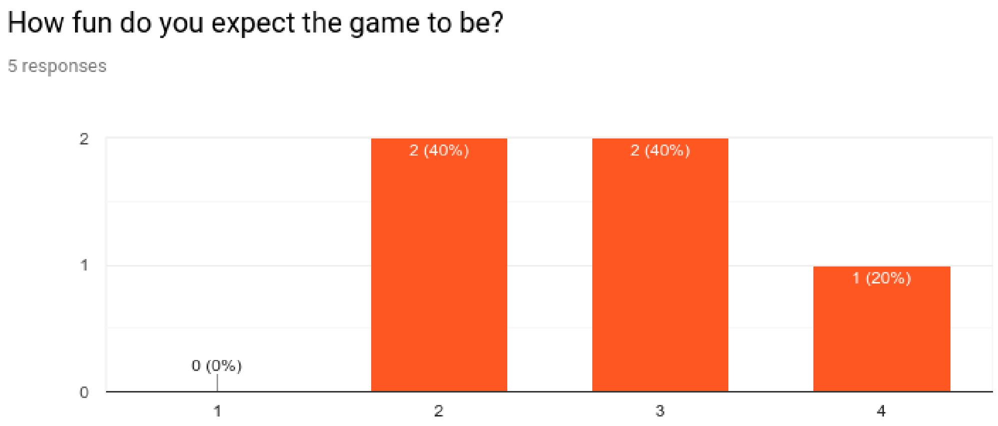
4 responses

impulse suppression

Handling in the last level and the waiting

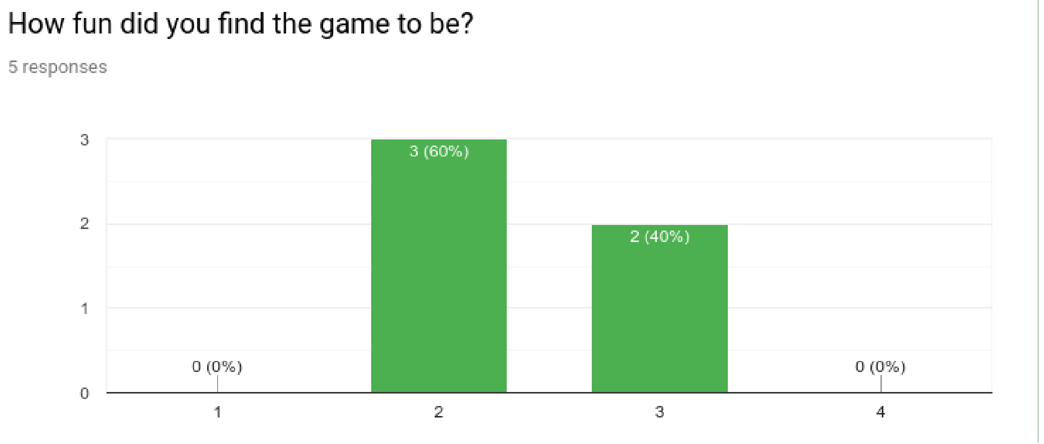
the factory level - > I had no orientation where to go and what to do. i missed a waypoint

Escaping drones when detected



How fun do you expect the game to be?

5 responses

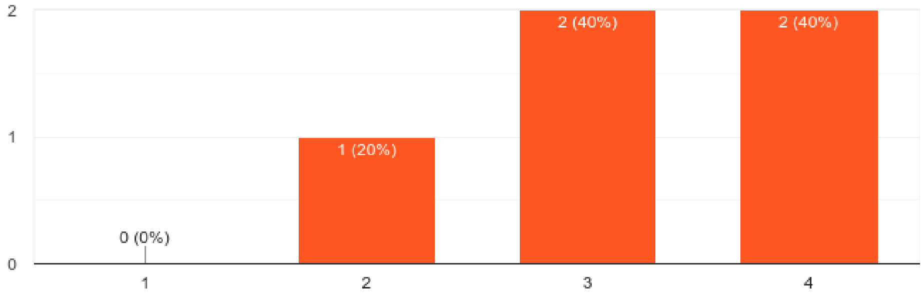


How fun did you find the game to be?

5 responses

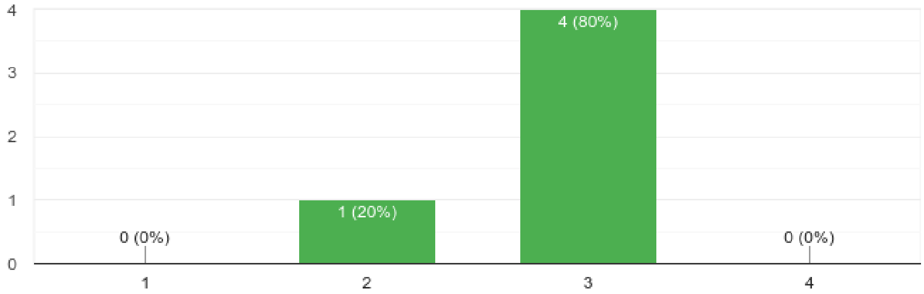
How complete do you expect the game to be?

5 responses



How complete did you find the game to be?

5 responses


4
3
2
1
0
4 (80%)
1 (20%)
0 (0%)
0 (0%)
1
2
3
4