


# Portfolio Document

DES310

Student Name	Filippo Casola
Student Number	1702870
Role	Game and Level Designer
Team Name	██████████
Brief	████████████████████

## Contents

Research .....	3
 .....	3
Products identified by the client as key references (including existing IP) .....	3
Other media such as films, animations, music, and art that provided me with inspiration.....	3
Books, magazines, and credible web sources that provided insight into your discipline and informed my response to the brief .....	3
Guides and tutorials that cover software, pipelines, and processes relevant to my role .....	4
Project Work .....	6
Minutes on 13-01-20.....	6
Feature list.....	7
Provisional list of spawner needed for single level.....	8
Provisional list of assets needed.....	8
Week 3 and 4.....	11
Week 5 .....	15
Dimension of map – directly influenced by the type of ambient.....	18
Enemy design- Combat design .....	19
Gameplay Cells V1.....	20
Presentation part .....	23
Further experiment with cells design .....	26
Cells v2 .....	27
Developing cells V2 .....	32
Cells V3.....	40
Developing cells V3 .....	49
Peer presentation.....	54
Documentation .....	58
Documentation 2.....	63
Miscellaneous notes.....	64
Development work.....	71
References.....	72

## Research

[REDACTED]  
[REDACTED]  
[REDACTED] I played and analysed Velocity 2X, of which, a more in-depth analysis can be found inside “Supporting level design document”. (This decision was made to avoiding repetitions).

### Products identified by the client as key references (including existing IP)

In the first few weeks, the client was quite foggy on the details of the project, they were not sure what kind of game it would be, and they were very confused. Once we asked them a couple of questions to narrow down the kind of project they were expecting from us, they told us to look at Dead Cells, saying that they wanted to replicate that game but in 3D.

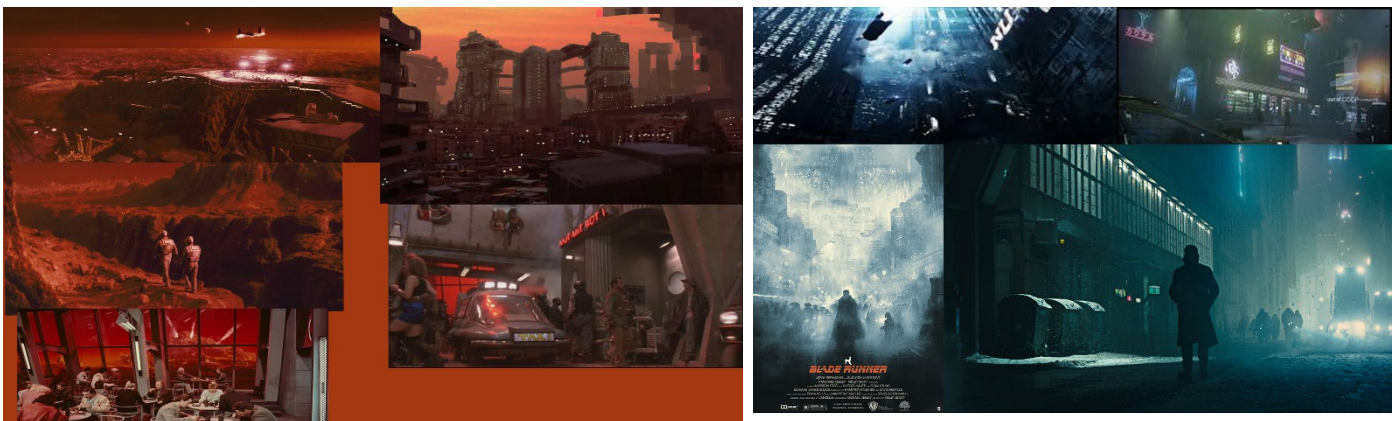
To do so, I played through the game, I analyse its game, flow, enemies, movement mechanics, upgrades, and map generations to better understand what the client wanted. A more in-depth analysis can be found inside the “Supporting level design document”.

[https://www.gamasutra.com/blogs/SebastienBENARD/20170329/294642/Building\\_the\\_Level\\_Design\\_of\\_a\\_procedurally\\_generated\\_Metroidvania\\_a\\_hybrid\\_approach.php](https://www.gamasutra.com/blogs/SebastienBENARD/20170329/294642/Building_the_Level_Design_of_a_procedurally_generated_Metroidvania_a_hybrid_approach.php)

[https://youtu.be/HZA\\_H9aQuo4](https://youtu.be/HZA_H9aQuo4) I made this video to analyse every enemy of Dead Cells. The video took me longer than expected, but it helped me to better observe the flow and enemy encounters.

### Other media such as films, animations, music, and art that provided me with inspiration

Total recall (Mars), Blade runner and Cyberpunk inspired games



These mood boards represent the vibe I was going for.

### Books, magazines, and credible web sources that provided insight into your discipline and informed my response to the brief

Level design concept, theory and practice

Gamasutra

80LVL

GDC vault

Gameindustry.biz

### [Guides and tutorials that cover software, pipelines, and processes relevant to my role](#)

*XCOM 2* generation of the level was a good direction for us, since the tool that we wanted to create was similar in function. To understand fully how it worked, I read the article on Gamasutra, watched the GDC presentation and watched the algorithm at work. The base concept for the plot and parcel of *XCOM 2* is simple. Each level has a standard dimension. Once the level has to generate, the algorithm chooses from one of the six different biomes.

Once the biome is chosen, the overall map is divided in various parcel of three standard dimensions:

- Small which is 12 by 12
- Medium which is 12 by 24
- Large which is 24 by 24

Once the parcels are placed, the algorithm chooses what specific biome's asset to place and where, following the rule of the mission, keeping in mind the type of mission and objectives that the player has to achieve.

Once that is done, a pathing algorithm try to navigate the map, to check if the player can traverse the map without any problem.

Then, the algorithm places the enemy pods (randomly assortment type of enemies) and build lights.

Video of XCOM level generation <https://www.youtube.com/watch?v=NhFLDAfqNp8>



*Phoenix point* uses a system in part similar to one found in XCOM. The system in Phoenix point choses a map layout from a pool. These layouts are pre-defined and fixed, and they define the position of roads, the path layout, and the type of building contained in it. Once the layout is chosen, the algorithm choses the type of building, how these buildings are tall and how they connect; then it fills the blank spaces between the pre-defined assets. This approach is less “automatized” than the Xcom’s, since the layout has to be man-made. Here the algorithm helps the level design to complete the level, rather than doing the whole level under the direction of rules as it happens in XCOM [https://phoenixpoint.info/blog/2017/11/28/haven-design-and-map-generation?utm\\_source=Phoenix+Point&utm\\_campaign=85a3095af3-NOV BLOG 2017 12 01&utm\\_medium=email&utm\\_term=0\\_0f91b4b9df-85a3095af3-178062745&mc\\_cid=85a3095af3&mc\\_eid=6c851a8ad3](https://phoenixpoint.info/blog/2017/11/28/haven-design-and-map-generation?utm_source=Phoenix+Point&utm_campaign=85a3095af3-NOV BLOG 2017 12 01&utm_medium=email&utm_term=0_0f91b4b9df-85a3095af3-178062745&mc_cid=85a3095af3&mc_eid=6c851a8ad3)

*Houdini with WFC Dungeon* No notes, interesting execution, but since they are not using Unity, just a good reference for <https://www.youtube.com/watch?v=c06bSBYsFT8>

Research into interesting forms deriving from Japanese movement (as seen from the collage of interesting shapes) <https://photos.app.goo.gl/A7hRA4rBwA9BY92b>

Enemy encounter and enemy design <http://www.fullbrightdesign.com/2009/02/basics-of-effective-fps-encounter.html>

System in Unity that creates cities with the Wave Function Algorithm. Interesting to read, but sadly not applicable for our project <https://selfsame.itch.io/unitywfc>

## Project Work

In the first week, the roles weren't really set in stones, so I took a very crude version of minutes to have something in case the other designer did not want to take the role of the producer.

### Minutes on 13-01-20

- -Meeting with Gaz and introduction to the project
- -Heavy emphasis on communication between group from all members is required
- -GitHub for source control
- -Example of how the wave function collapse algorithm works was shown
- -Bladerunner and total recall as design influences. Dark mood, pretty light, everywhere is Times Square, spotlight going casually around the city. What it means to be human and such
- -Example of Oskar Stallberg, irregular grid and bad north as an example of wave function collapse
- -We happy few as in two districts, one rich and one poor, different style and how well they are kept.
- -Mario Maker as an example of adding blocks until the background changes
- -Design decisions are hard to make until [REDACTED] can clarify some things.
- -Trello as a way to coordinate the team and keep track of what has to be done
- -What kind of production (agile or waterfall)? Towards agile at the moment
- -Someone needs to do some basic UI for the project
- -Artist pipeline? Need more time to decide
- -Obj/fbx for 3d models
- -Introduction to unity/source control for those who did not know how to
- -Each one has a level in the source control project that can uses
- -List of minimum deliverables made
- -Deadline for grey box for 28th of February
- -Discord set up with different category, channels and role

- -GitHub repository created, and all team members invited
- -List of minimal deliverables both spreadsheet and drawn on board

## Feature list

- -Weapon dropping from killed enemies
- -Zone dependent variation in enemy design, weapon and tactic
- -If the scale of a single level is medium to considerable, there could be multiple ways to quick travel in the city, via the sewers or via the monorail. Each mode of transportation would have its advantages and disadvantages. Not sure if portals are considered as these two or as portal through the level - need clarification from F(utur)L(ab)
- -Portals? Or just quick travel through maps?
- -Different shops with different colours and items.
- -Maps have coherency and thematic link to each other
- -If unique selling point is traversal at high speed, upgrades to movement and player character is needed
- -Boost scattered around the map to further the traversal speed.
- -Verticality in every map but also thematically (from the slums to heaven on earth)
- -Map structure similar to Dead Cell
- -Spotlight impeding and breaking up the player traversal of the map, forcing them to consider secondary routes.
- -Skyscrapers as vantage point/observational towers
- -Combat upgrades for melee but no upgrades for range, only weapons for range are unmodifiable
- -High tech weapons are highly effective but not upgradable, while low tech weapons are highly moddable but quite weak in their bare form
- -Interactable ambient
- -The start room is a station so similar level to level with a small shop just to refill health etc.
- -Stealth takedown

- -Sneaking past enemies by staying in the shadows and avoiding the spotlight in the streets
- -Verticality to avoid enemies and surpass highly fortified post
- -Interior section treated like boss section?
- -Interior section used to give player break for pacing and forwarding the story
- -Enemies checkpoint to force the player to upgrade or find another route

#### Provisional list of spawner needed for single level

- 1) Player Character (Single spawner)
- 2) Enemies (multiples spawner across the map 3/4)
- 3) Keys (3 spawner)
- 4) Doors (3 spawner)
- 5) Portals (two spawner)
- 6) Stages Doors (two spawner of level exits)
- 7) Shop (single spawner/multiple spawner)
- 8) Spotlight (three spawner)

#### Provisional list of assets needed

Type	Asset	Priority
Hero	Shop	High
Hero	Start room	High
Primary	Road	High
Primary	sidewalk	High
Primary	windows	High

Primary	door	High
Primary	building	High
Primary	Roof	High
Primary	Archway/Arcade	High
Primary	Key	High
Primary	Ladder	High
Primary	stairs	High
Primary	Bridge	High
Primary	Portal	High
Secondary	Lamppost	Medium
Secondary	Barrel	Medium
Secondary	Car	Medium
Secondary	Trees?	Medium
Secondary	Rocks	Medium
Secondary	Debris	Medium
Secondary	Bins	Medium
Tertiary	Tufts of grass	Low
Tertiary	trash bag	Low

Tertiary	small stones	Low
----------	--------------	-----

Tertiary	billboard	Low
----------	-----------	-----

Tertiary	manhole cover	Low
----------	---------------	-----

Tertiary	Grate	Low
----------	-------	-----

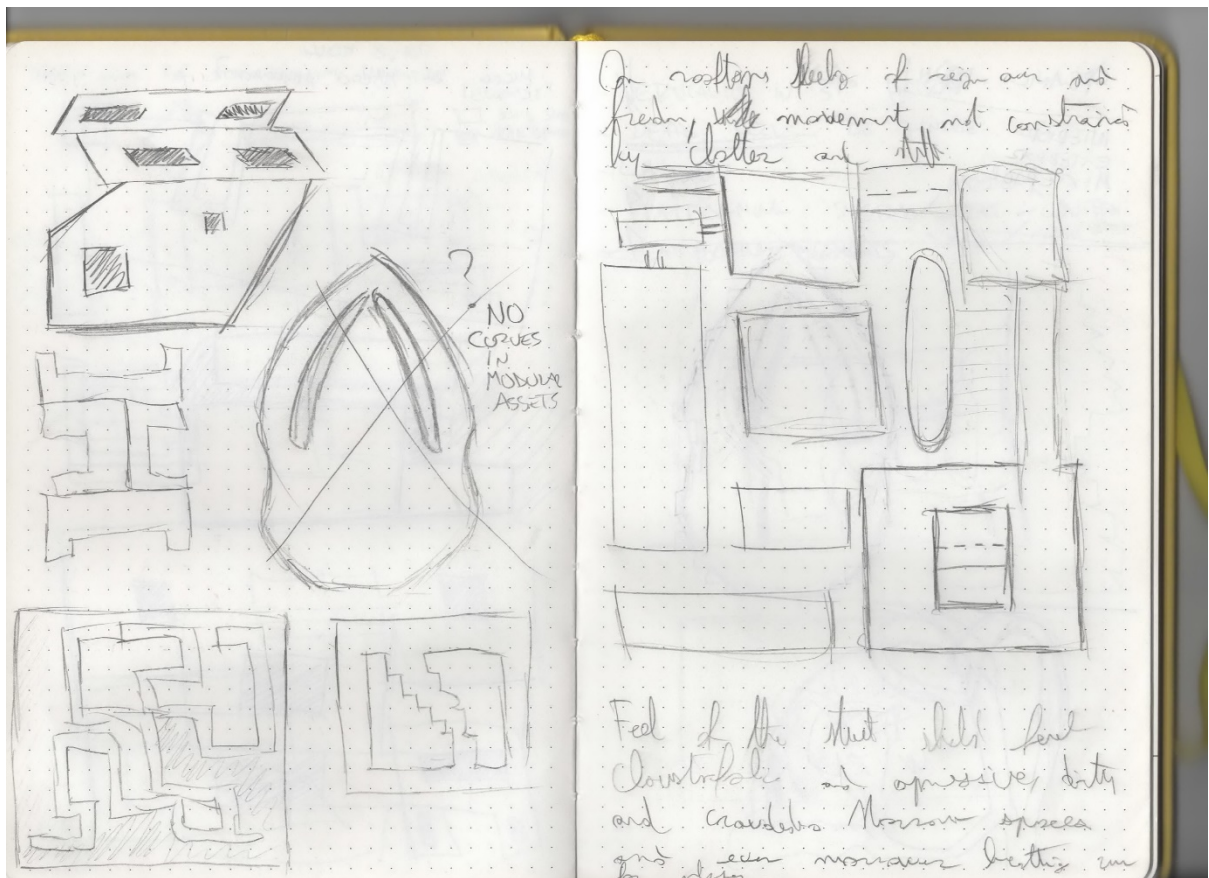
Tertiary	Pick up upgrade	Low
----------	-----------------	-----

## Week 3 and 4

After [REDACTED] gets back to us with answers to our questions, level design as a role is back on the table. From the answer is pretty clear that the guys at [REDACTED] want a tool rather than a game. The deliverable is planned to be a level (or a couple if time allows for it) showcasing the tools used to create both the level and the architecture present in the virtual space.

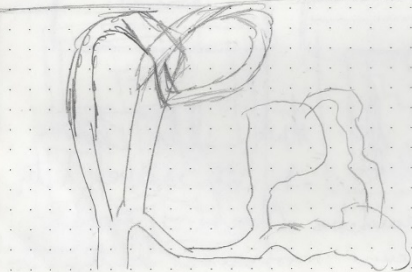
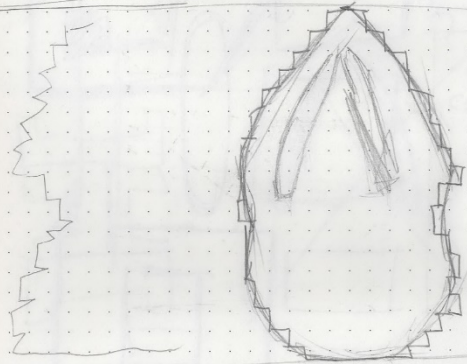
At the moment I was concentrating on blocking out interesting shapes for the map, keeping in mind that the smallest building block for the algorithm is a tube. This will reduce the amount of possibility for shapes, with more complex shortcuts and layout.

This shouldn't have discouraged me since that from the simplicity of a square I could have built a more modular and regular outline. So far, the most promising map was the one with three isles. While it was not the most original concept, the possibility to traverse the map on both bridges or use the gravitational pull to traverse the cliff (avoiding so to fall into the lava/glacier underneath) was appealing. Once this design was refined, I mocked the position of the buildings as if I was using the tool, to get a good grasp at what we are trying to achieve, since we (designers) are the intended demographic for the tool.

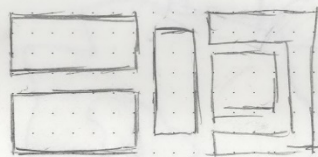
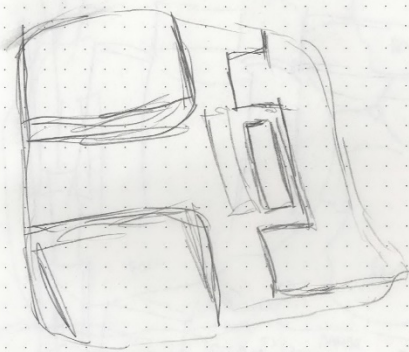
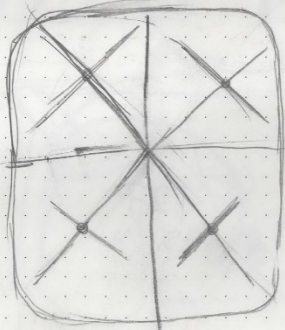
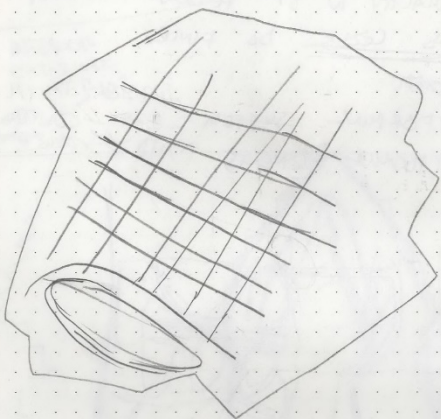
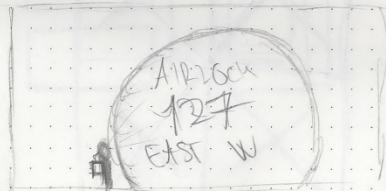
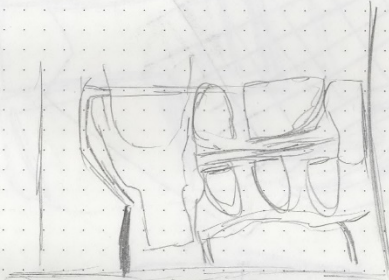


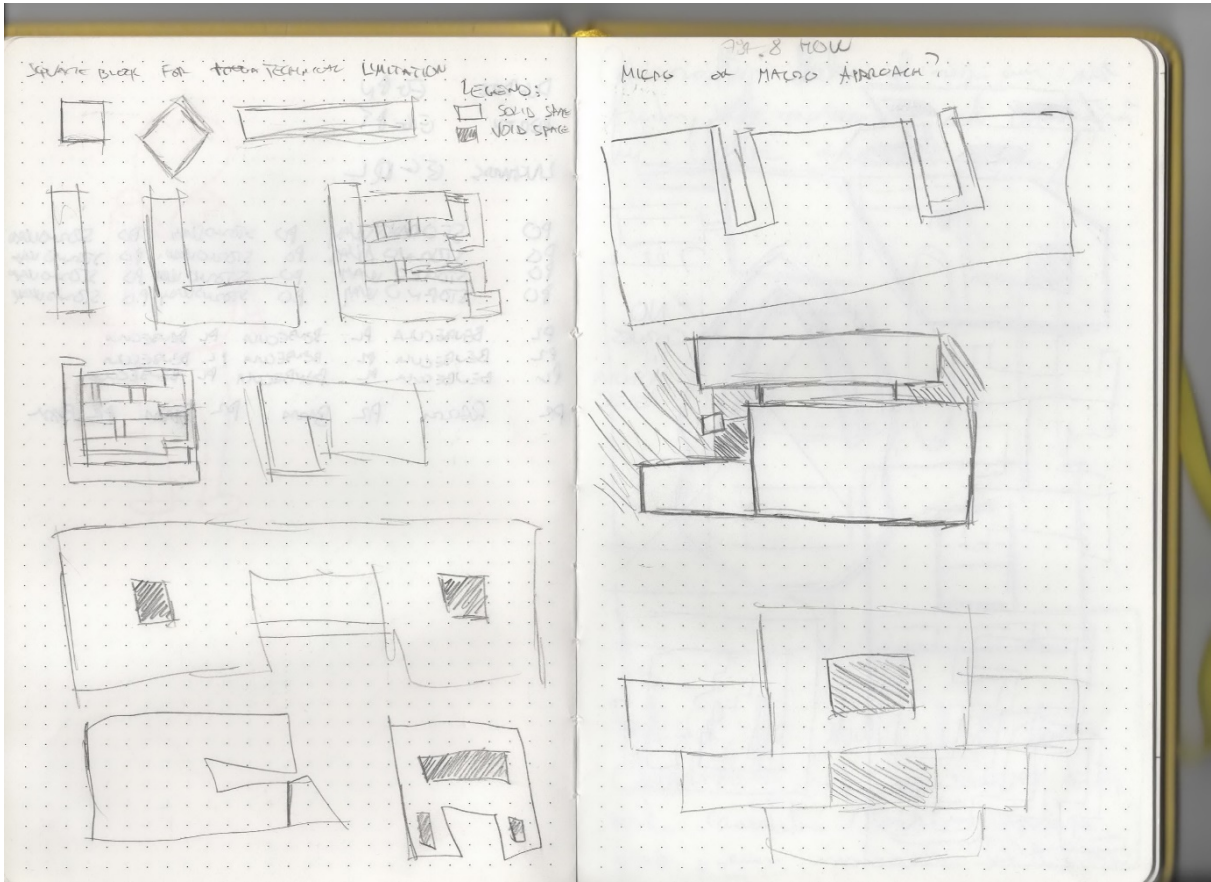
ORGANIC FORM

INTERIOR  
EXTERIOR  
MIXED

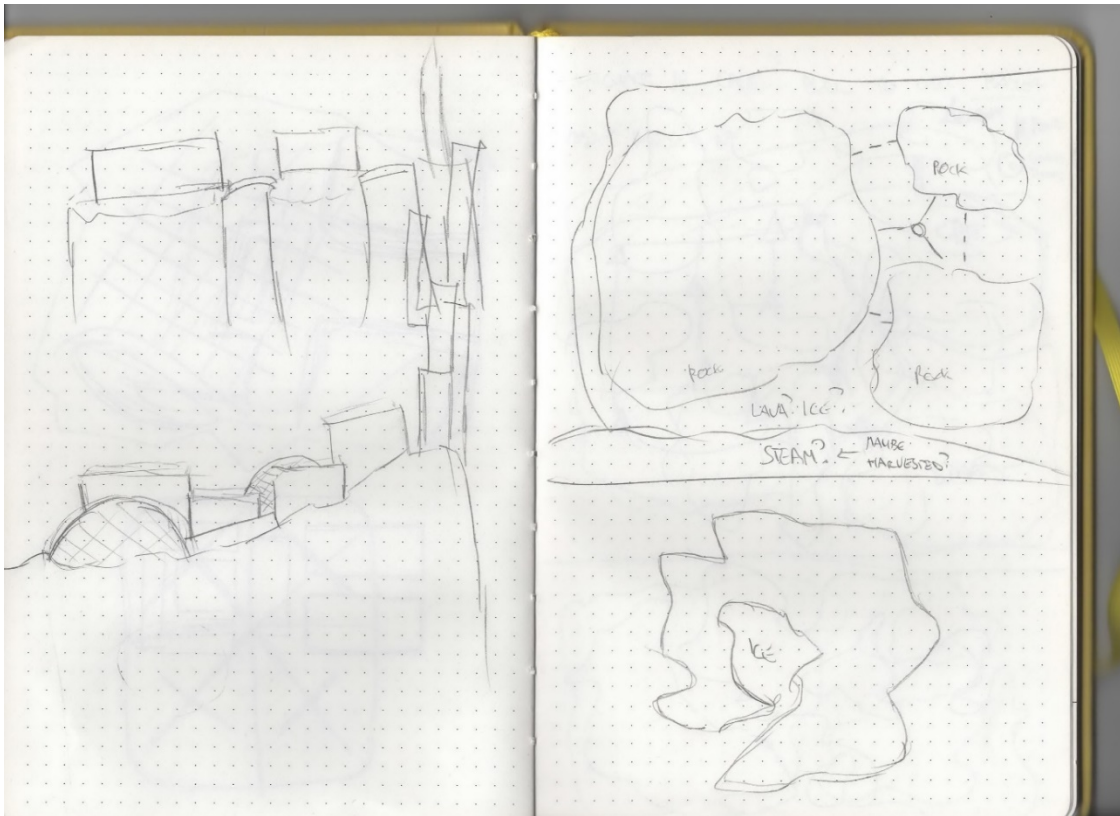


- VERTICALITY W 3RD ANGLE
- DEAS CELLS DA FINES
- LAPOORT
- PLATFORM SYSTEM / VERTICAL - EXT - CONTRAST
- PLATFORM ELEMENTS / MID / CONTRAST

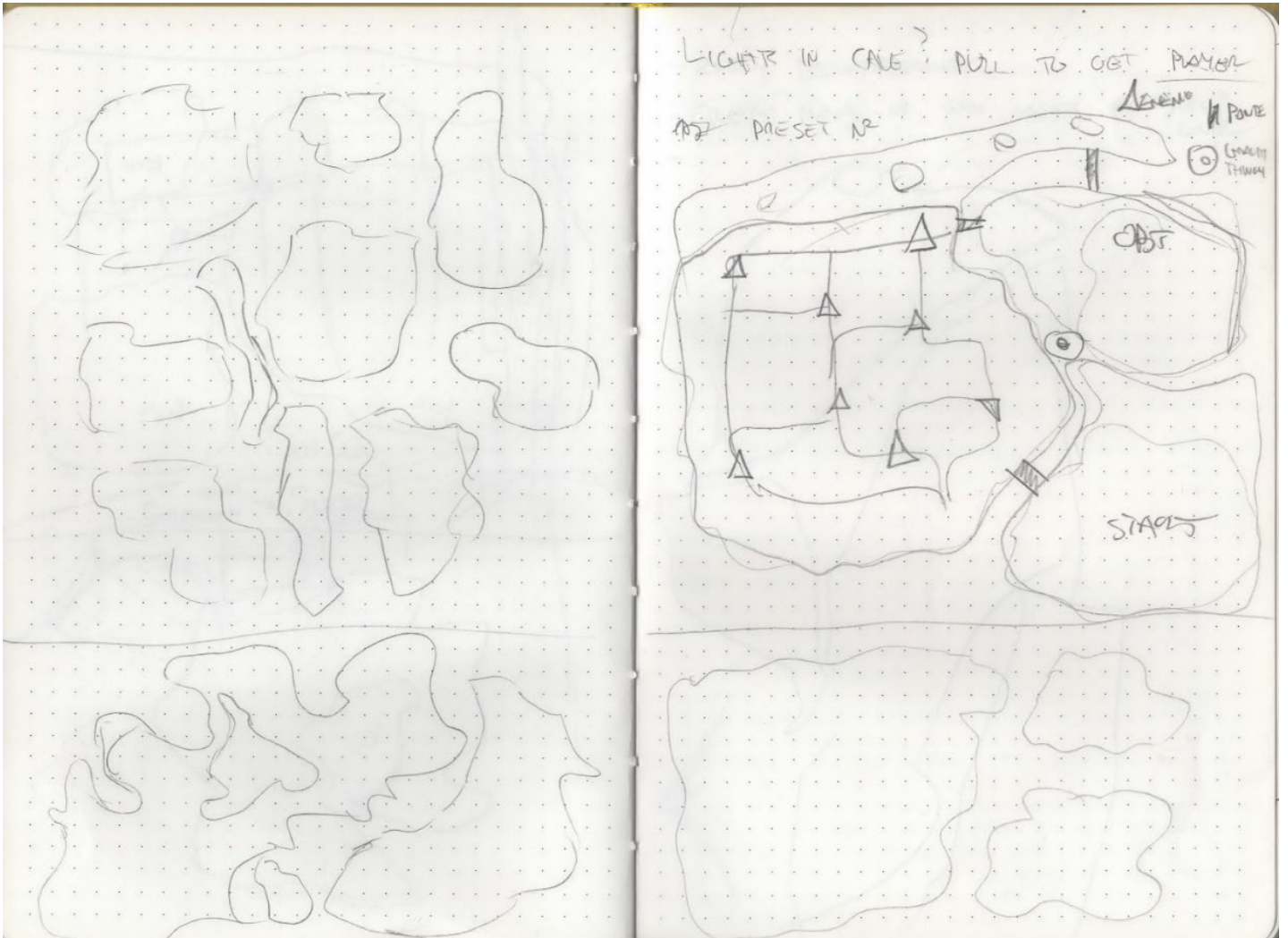




The top right layout is the layout chosen to work on after a sit in with the other designer.



Here's a mock layout for the chosen design on the top right



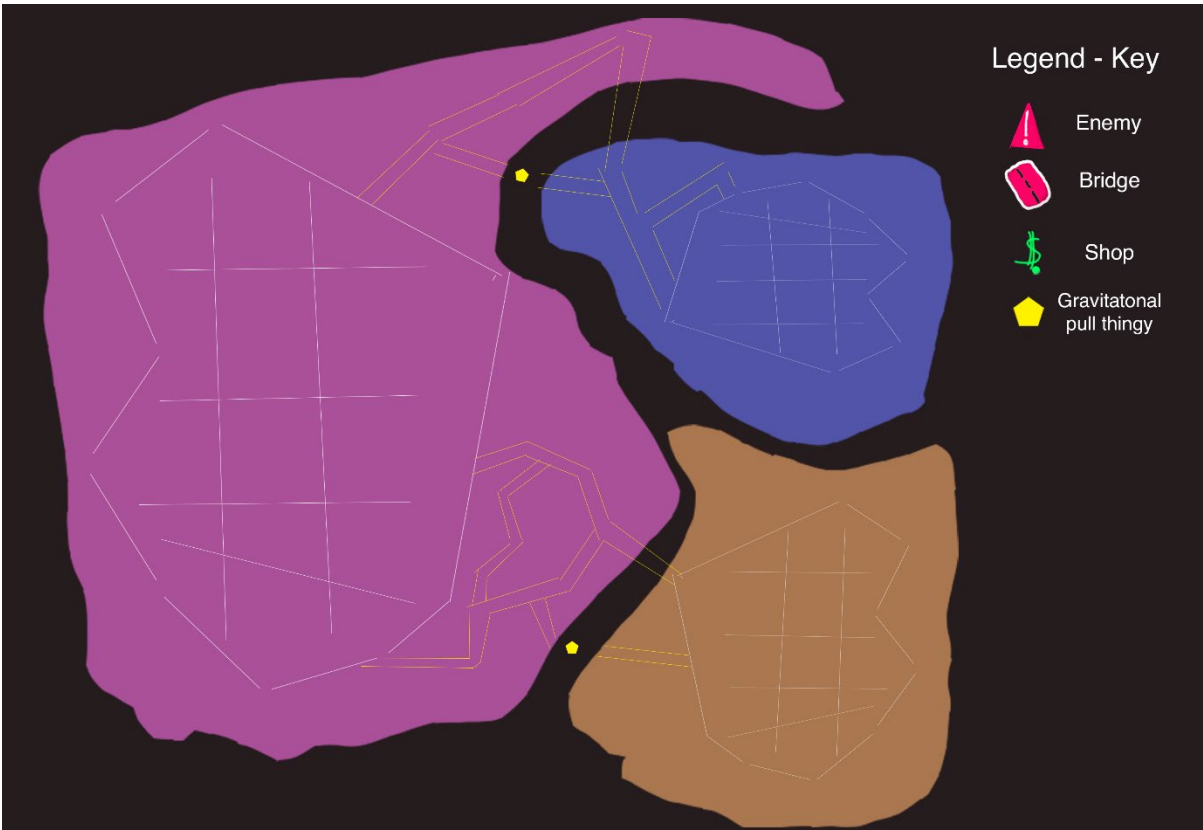
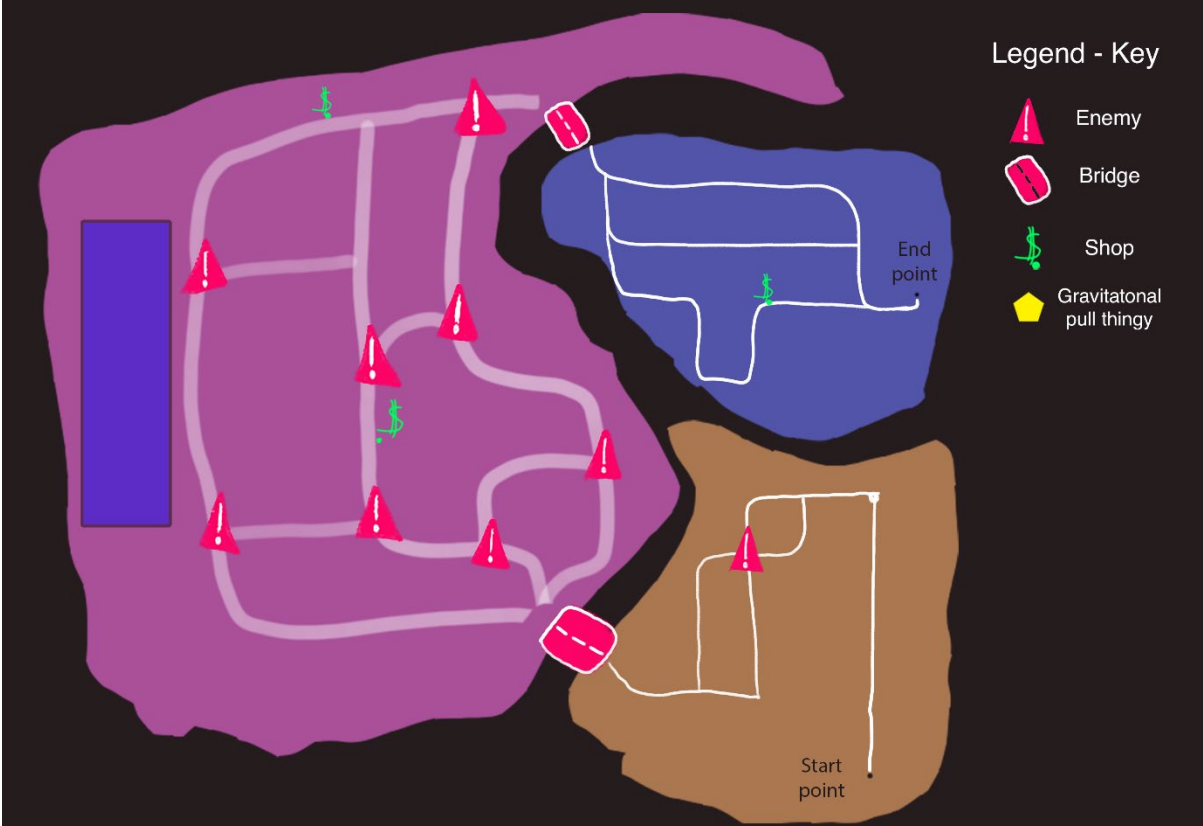
## Week 5

This week the focus was on usability and how well the map layout flow compares against each other's. As always, I was intentioned to try to respect the rule of 40 sec even though, as previously stated, the USP is traversal at high speed, so this shouldn't have been a problem.

Since the guys at [REDACTED] were extremely vague about the scale and length of levels/maps, we couldn't just stumble in the dark and build a tool without a clear scope or framework that would allow us to scale up and down the relative length and size. The image bellow was the foundation of this framework. It just represents (via different colours to help differentiate between the isles) the terrain of the map. There are three isles which are connected via manmade and natural structures, depending on the scope and on the kind of ambient they were supposed to be.

A concrete example would be the first draft, it represents a settlement scattered around three islands. The scale for this map would be a small map, located in a settlement, with a 60\*60mt.

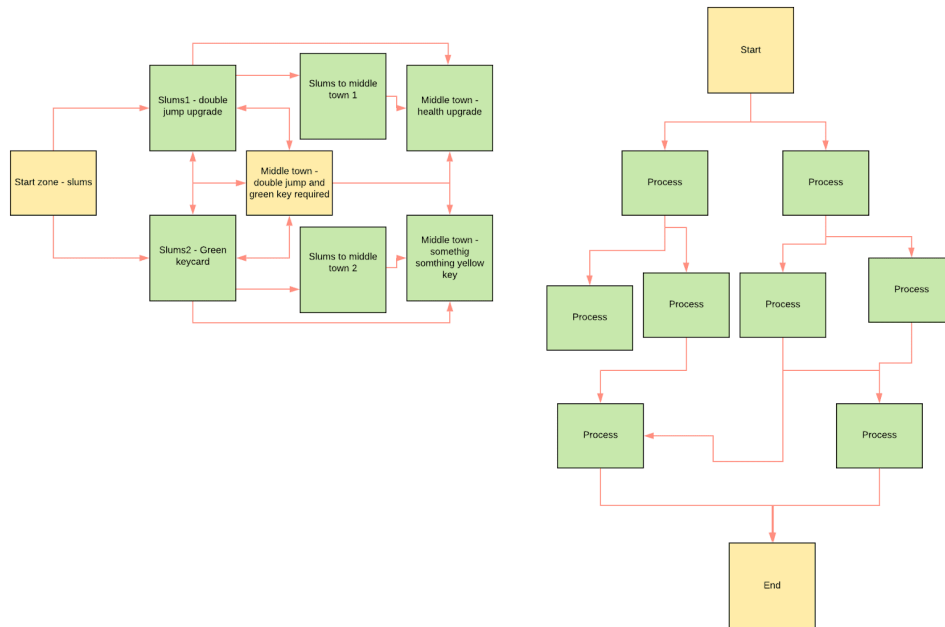






These three designs are to show that the scale of the level can change. Each level has a different traversal experience, as well as different overall experience.

Below is a diagram illustrating the progression that we had in mind at this point for the map.



Dimension of map – directly influenced by the type of ambient.

Small map – interior/city

- Shops
- Enemies
- Bridges
- Buildings
- Roads

Middle sized map – mixterior

- Shops
- Enemies
- Gravitational thingy
- Bridges
- Buildings
- Roads
- Lava tunnels + props

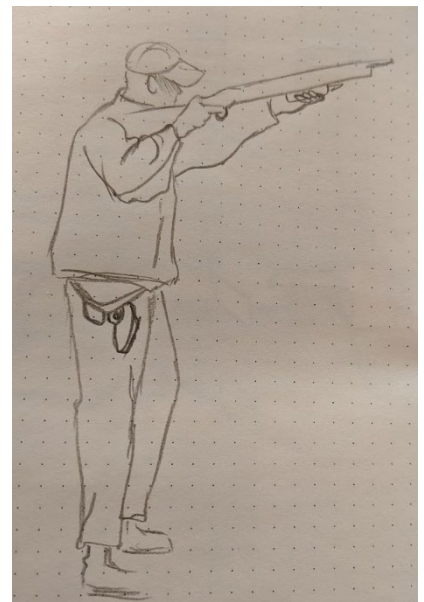
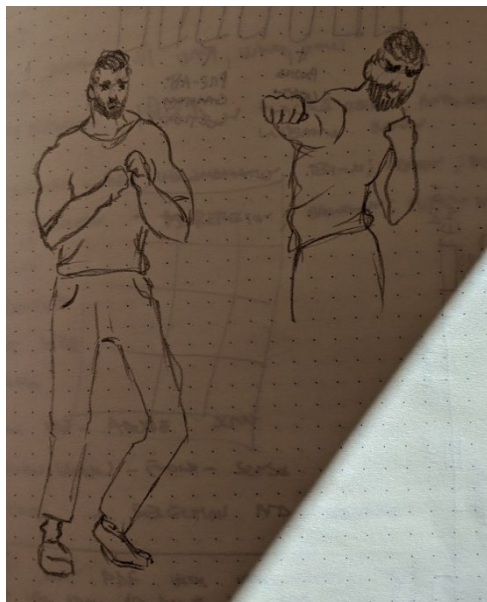
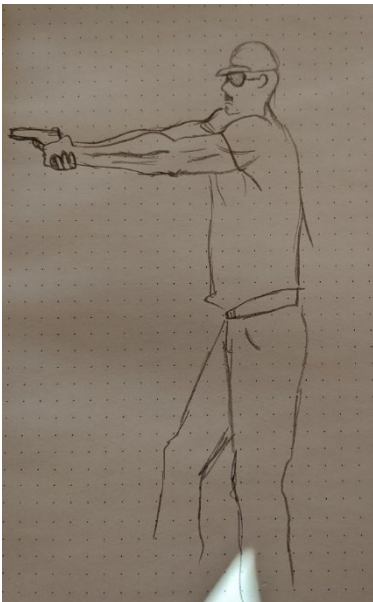
Large map size – interior/ mixterior

- Shops
- Enemies
- Gravitational thingy
- Bridges
- Buildings
- Roads
- Lava tunnels + props

## Enemy design- Combat design

Since we knew that the pace of the game was going to be pretty fast, the combat had to follow the same rule. So far, the gameplay that we had seen was pretty slow, with only a few mechanics implemented.

Different kinds of enemies in the level could have been melee and long range. Maybe one hybrid could have been medium range at the beginning then close range once the range attack is depleted. One enemy could stun the player or negate the gain of momentum, slowing it and forcing the player to fight it. Another encounter could be a trap disguised as gravitational object. Once the player starts to get near the trap, it could activate and blow up after a couple of seconds, so once the player has learnt about this trap, it can be easily avoided if given enough attention to the traversal.



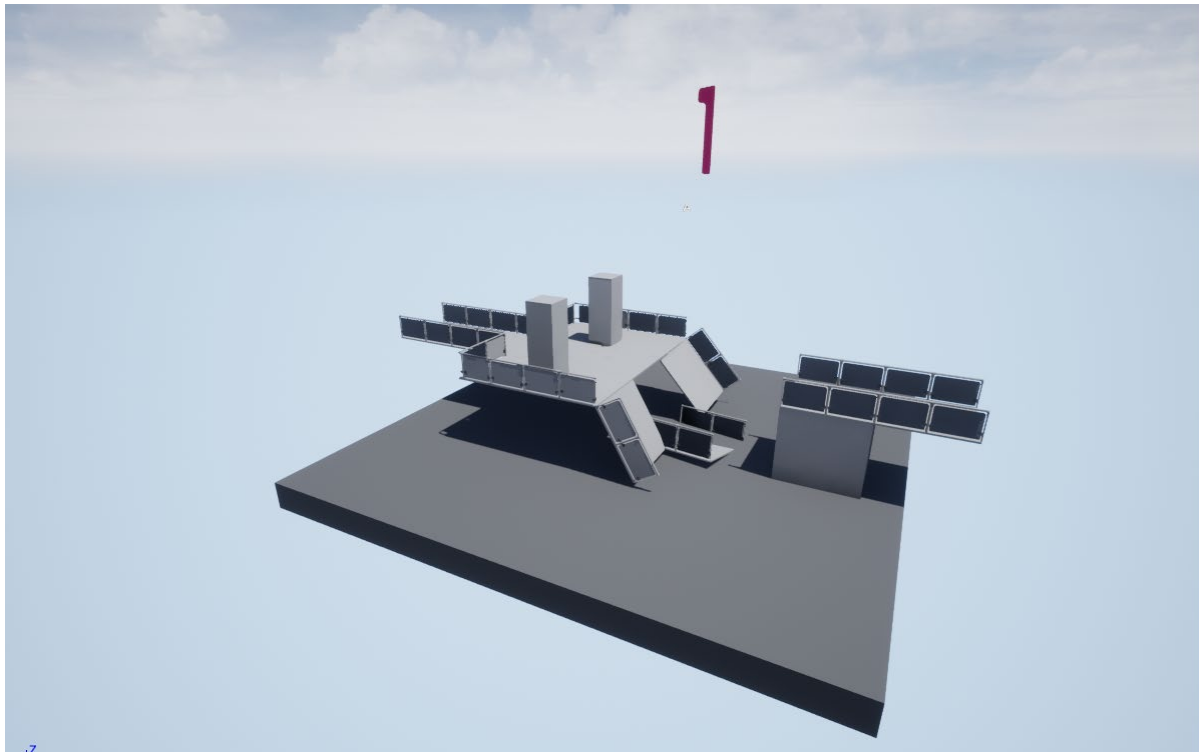
## Gameplay Cells V1

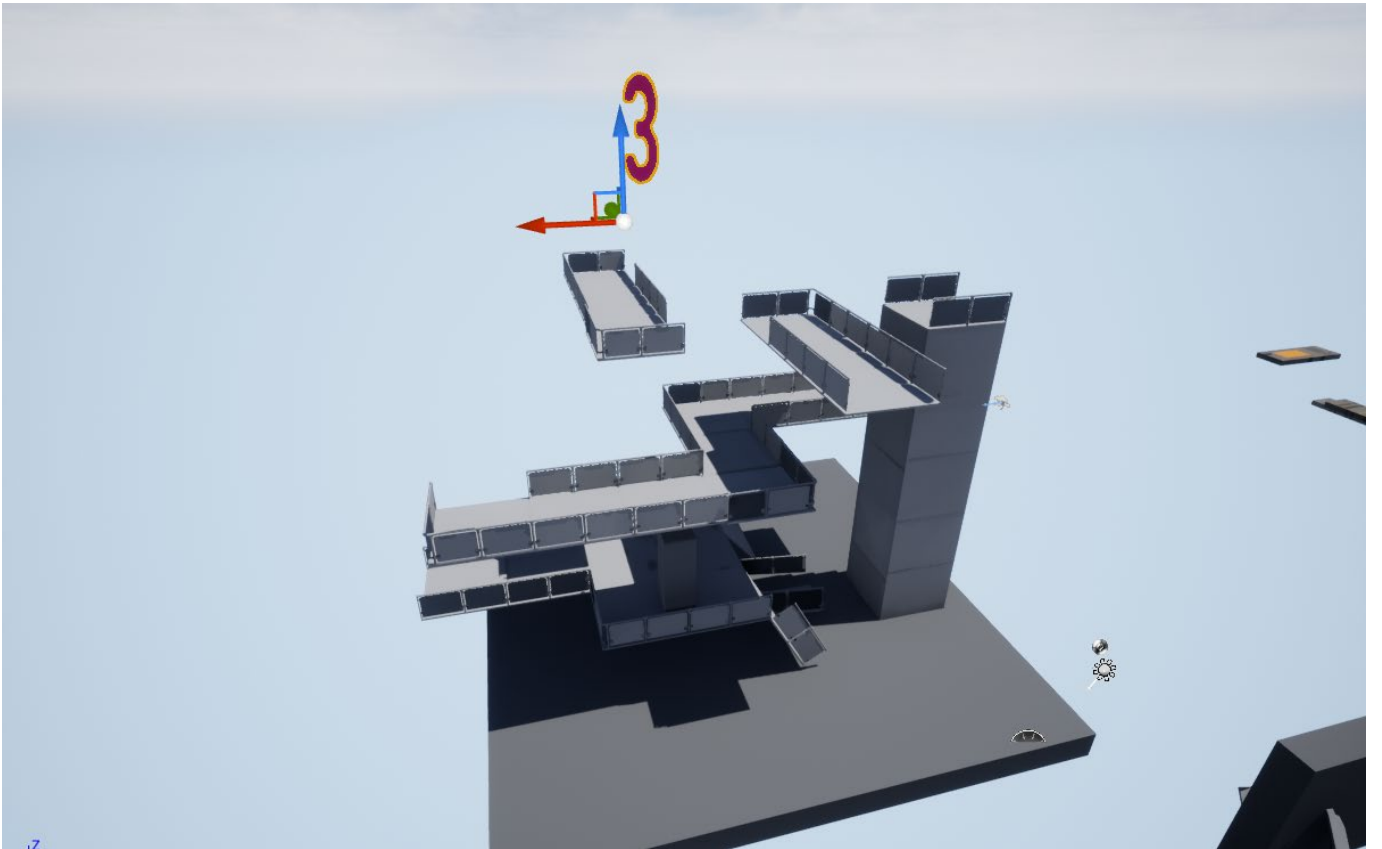
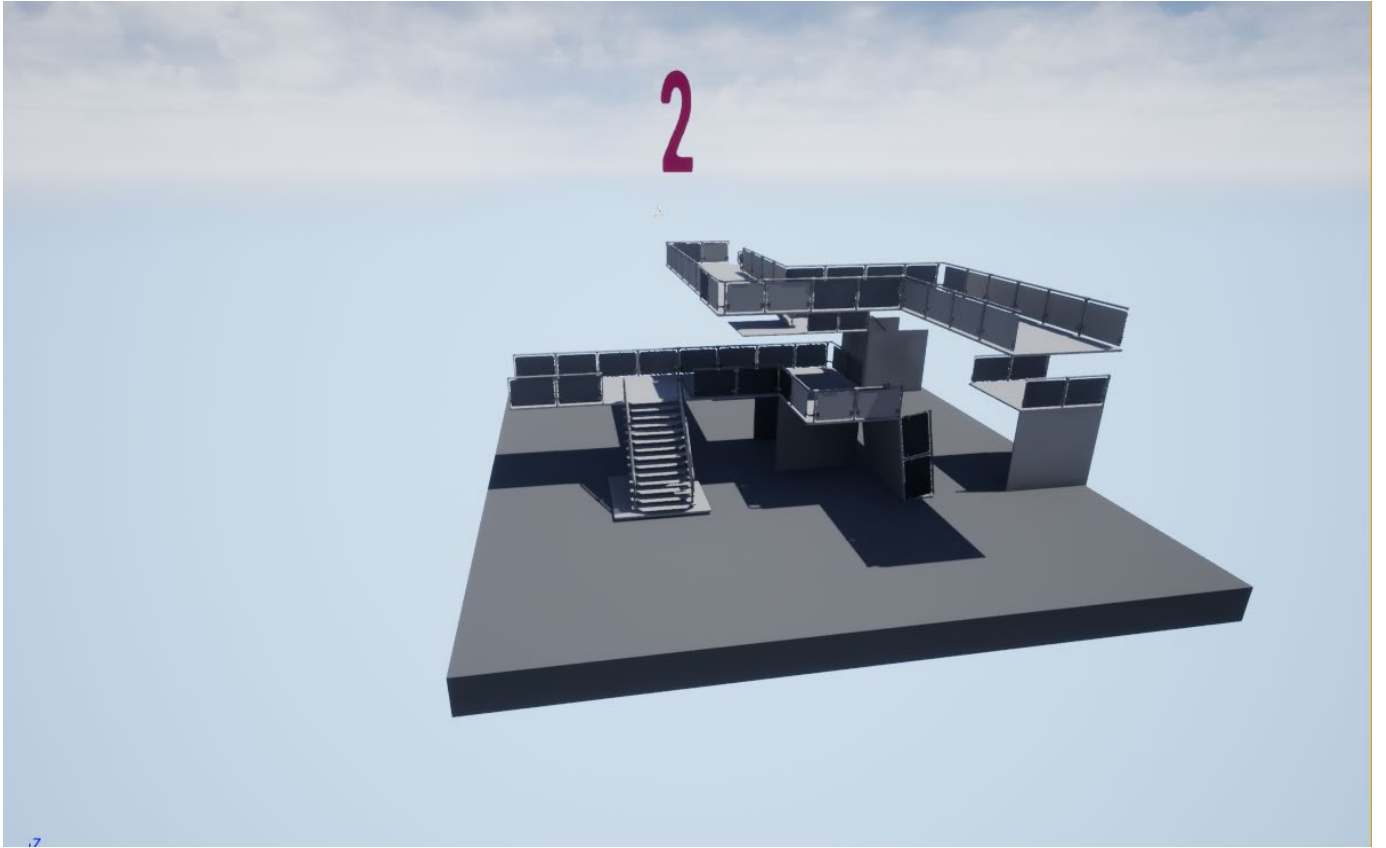
The first batch of cells was made in Unreal Engine. They were experimental cells which I designed them before meeting the client and before we decided to switch from a block basis to a cell-based focus. The block basis was how we approached the building of the level before switching to the approach of the cells similar to Twinmotion. The block approach was such a lengthy process that it would have taken the designer hours to create a single level, and since the client wanted a tool that would allow him to build swiftly good-looking levels, we opted to switch from it.

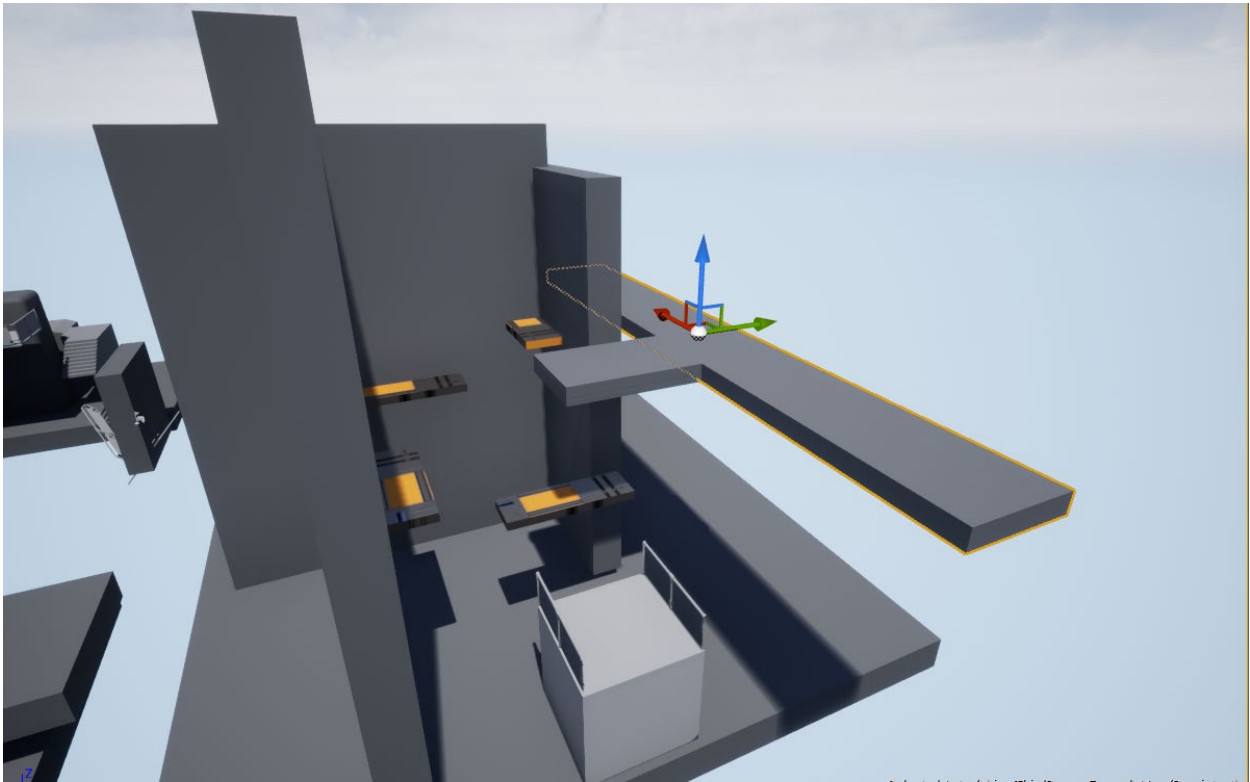
Every cell is centred around the same mechanic: the jump. This is because at this stage I wanted to design only a handful of cells to get a good pre-vis for the scale of the cells and how they would interconnect. At this stage we did not have a clear vision from the client of what kind of gameplay they would expect from the game, so I took the jump mechanics because it was the only thing confirmed from the client.

These cells were made in Unreal Engine 4 with assets made by the art team. I opted to use Unreal Engine due to the fact that the editor is easy to use and was easier to iterate than Unity. These assets were provisory and made by the art team.

Below a couple of screenshots showing the cells








## Presentation part


This is the presentation for the client. Here are the slides I did.



The slide features a diagram titled "Level Frame" on a dark background. A horizontal line contains ten colored circles: green, red, blue, pink, light blue, red, yellow, orange, red, and pink. Above the line, labels with vertical lines pointing to the circles are: "Start" (green), "Teleport" (red), "Exit Number 2" (blue), "Treasure" (pink), "Shop" (light blue), "Key for exit n2" (red), "Exit Number 1" (yellow), and "Teleport" (orange). Below the line, labels with vertical lines pointing to the circles are: "Treasure" (blue), "Teleport" (red), "Key for exit n2" (yellow), and "Teleport" (orange). Below the diagram is a list of five bullet points.

- Start + End of Level
- Teleporters - Fast Travel
- Treasure - Loot
- Shop - Buy Items
- Key + Locked Door

Ok, so this is the level frame. it's a visual representation of how levels are going to flow and the progression that the player is expected to make. Every level frame is made out of various elements such as shops, teleporter, etc...  
Each element is a rule for the algorithm, which is going to dictate how to make the procedural level.



The slide features a diagram titled "Level Frame" on a dark background. A horizontal line contains ten colored circles: green, blue, red, yellow, light blue, red, pink, red, and pink. Above the line, labels with vertical lines pointing to the circles are: "Start" (green), "Teleport" (blue), "Shop" (red), "Treasure" (yellow), "Exit Number 1" (light blue), and "Exit Number 2" (red). Below the line, labels with vertical lines pointing to the circles are: "Treasure" (blue), "Key for exit n2" (yellow), "Teleport" (red), and "Teleport" (orange).

Thanks to this modular approach, swapping elements around will change the player experience. For example, in a hard levels, the shop can be placed at the end while in easier levels shops can be scattered around the level to ease the player's journey

Thanks to these various elements, the player will experience each level without being fatigued by the repetition, and each level will play and feel different, engaging the player and saving both time and resources

I'd like to bring on feliks to talk about development and implementation

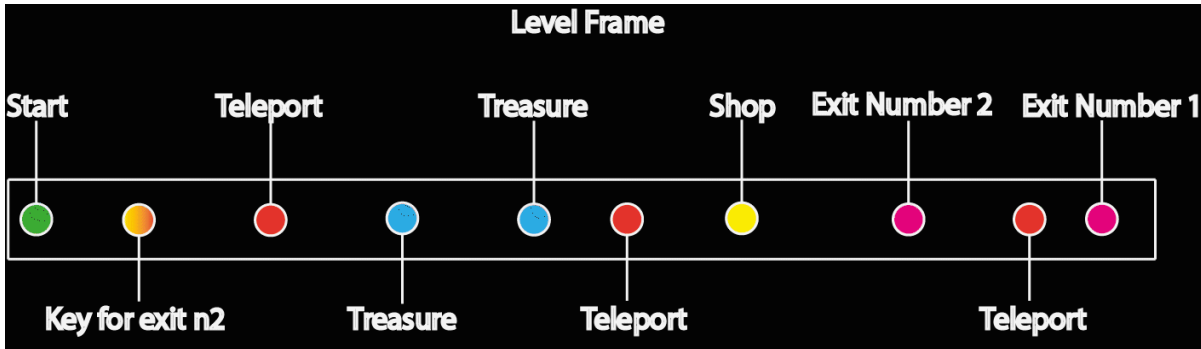
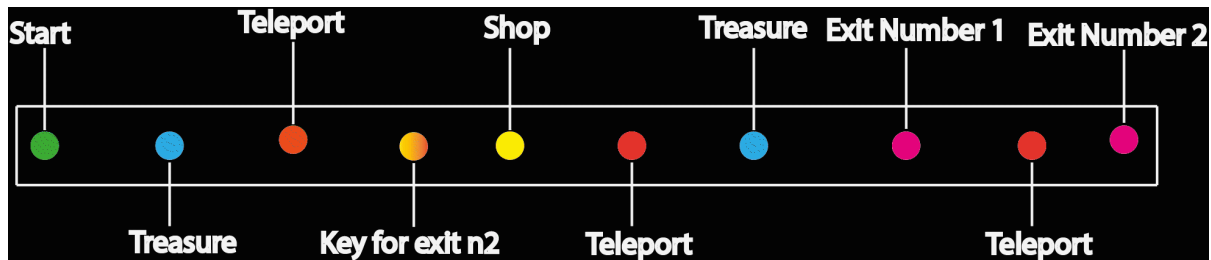
Visual representation

Modular approach>player experience. Example>shops

No fatigue>save both time and resources.

Elements are rules for the algorithm

Graphs made for the presentation



Script for the presentation

**First slide:** Ok, so this is the level frame. it's a visual representation of how levels are going to flow and the progression that the player is expected to make. Every level frame is made out of various elements such as shops, teleporter, etc...

Each element is a rule for the algorithm, which is going to dictate how to make the procedural level.

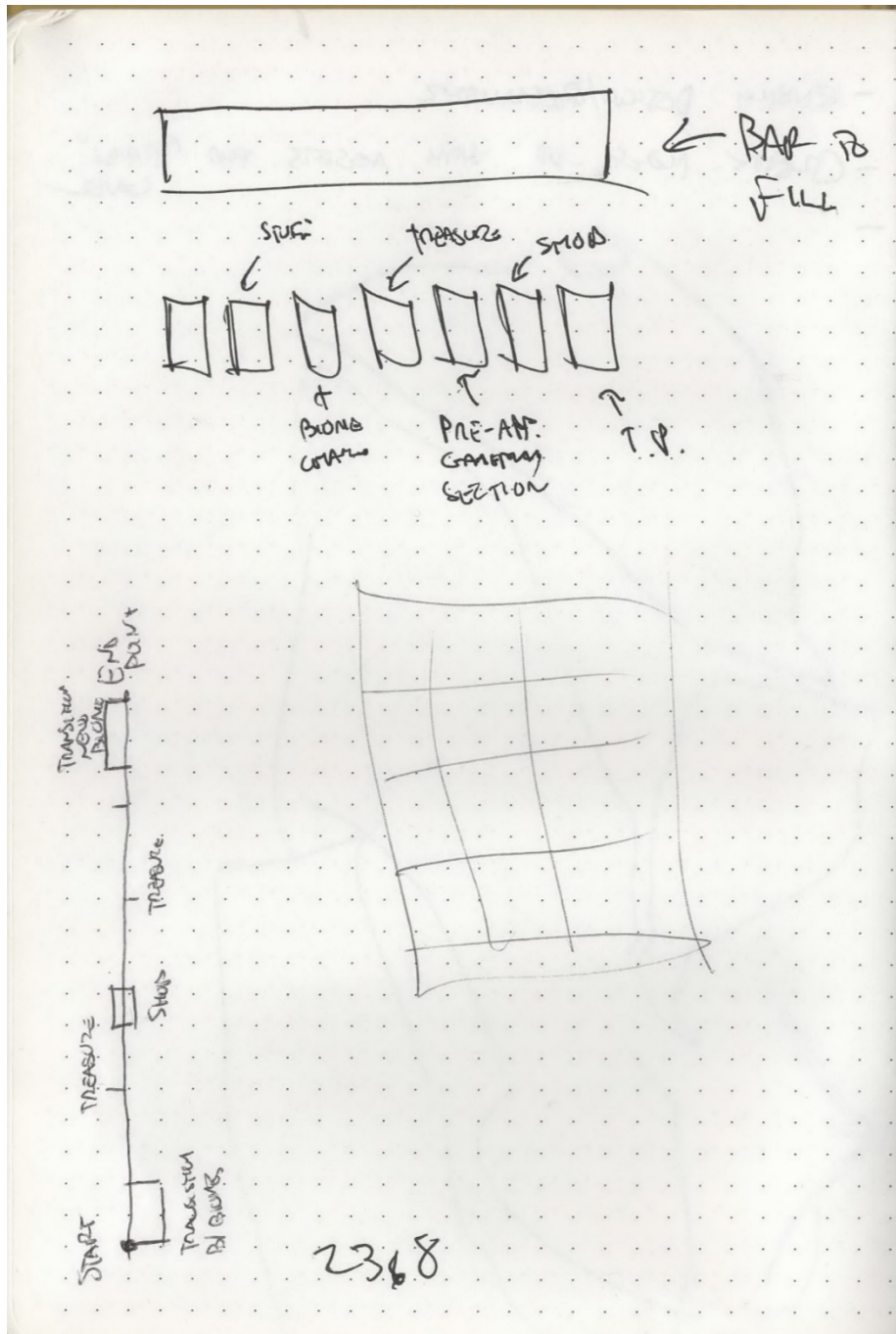
**Second Slide:** Thanks to this modular approach, swapping elements around will change the player experience. For example, in a hard level, the shop can be placed at the end while in easier levels shops can be scattered around the level to ease the player's journey

Thanks to these various elements, the player will experience each level without being fatigued by the repetition, and each level will play and feel different, engaging the player and saving both time and resources

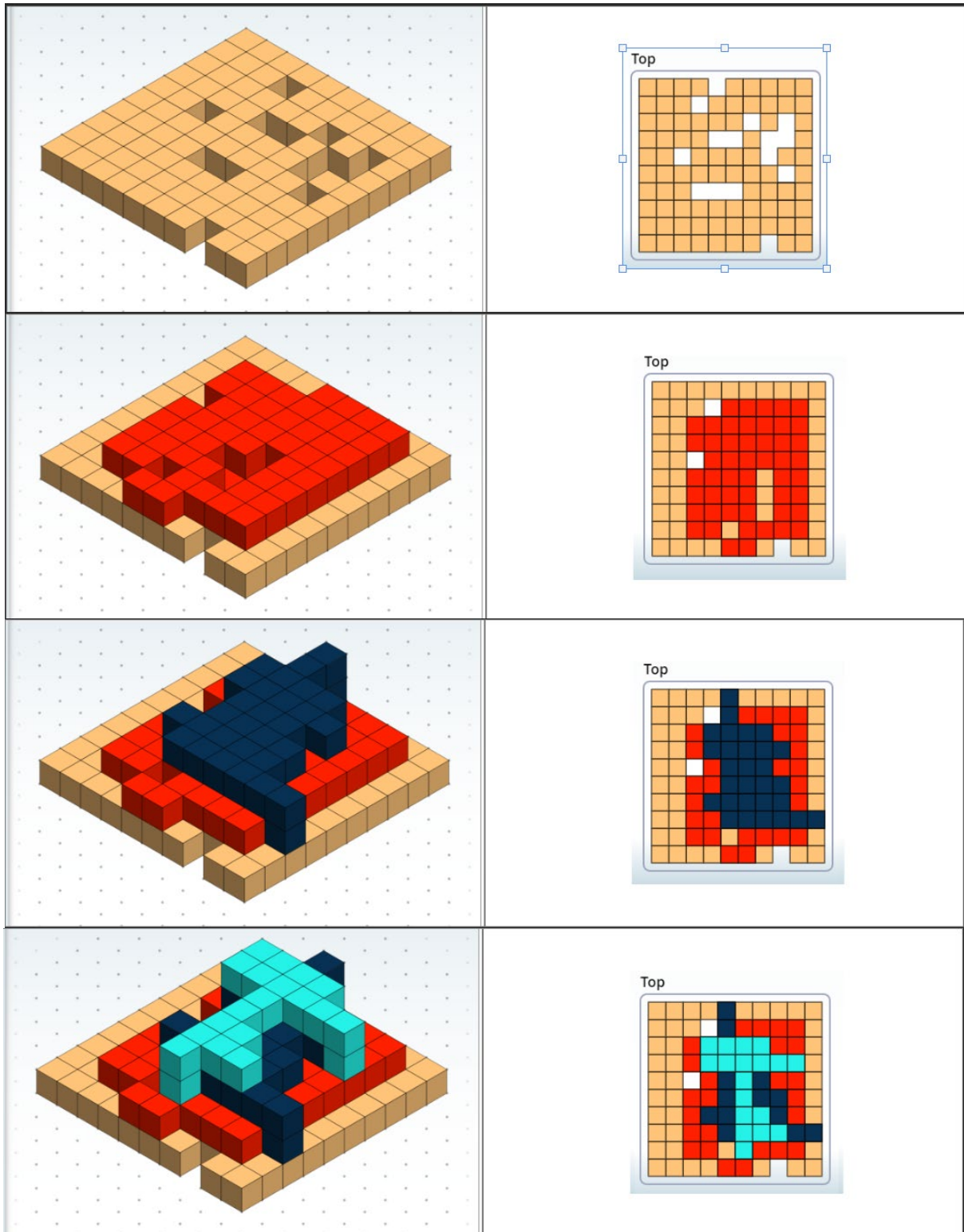
I'd like to bring on feliks to talk about development and implementation

- Visual representation
- Modular approach>player experience. Example>shops
- No fatigue>save both time and resources.
- Elements are rules for the algorithm

Early version of the graph representing the level frame

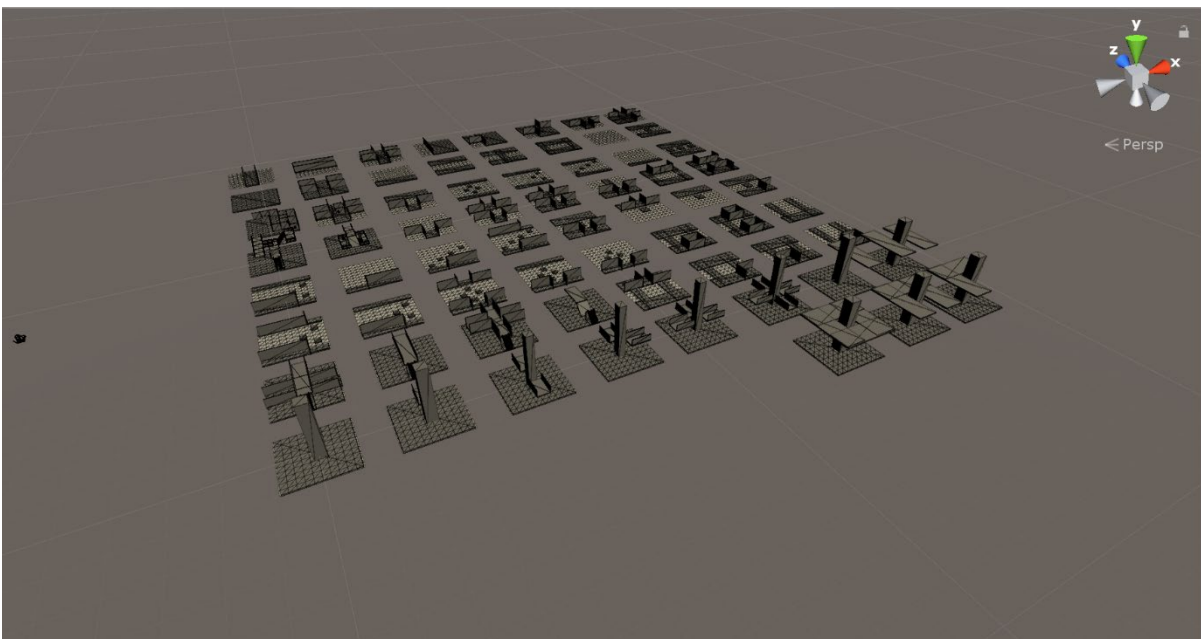
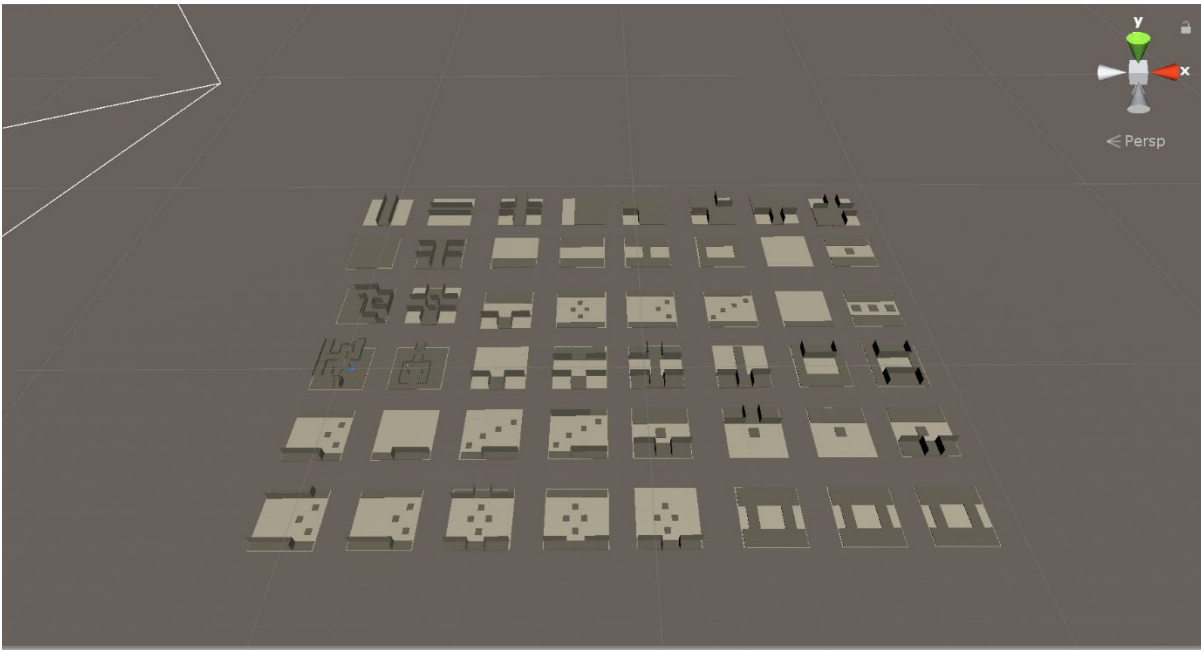


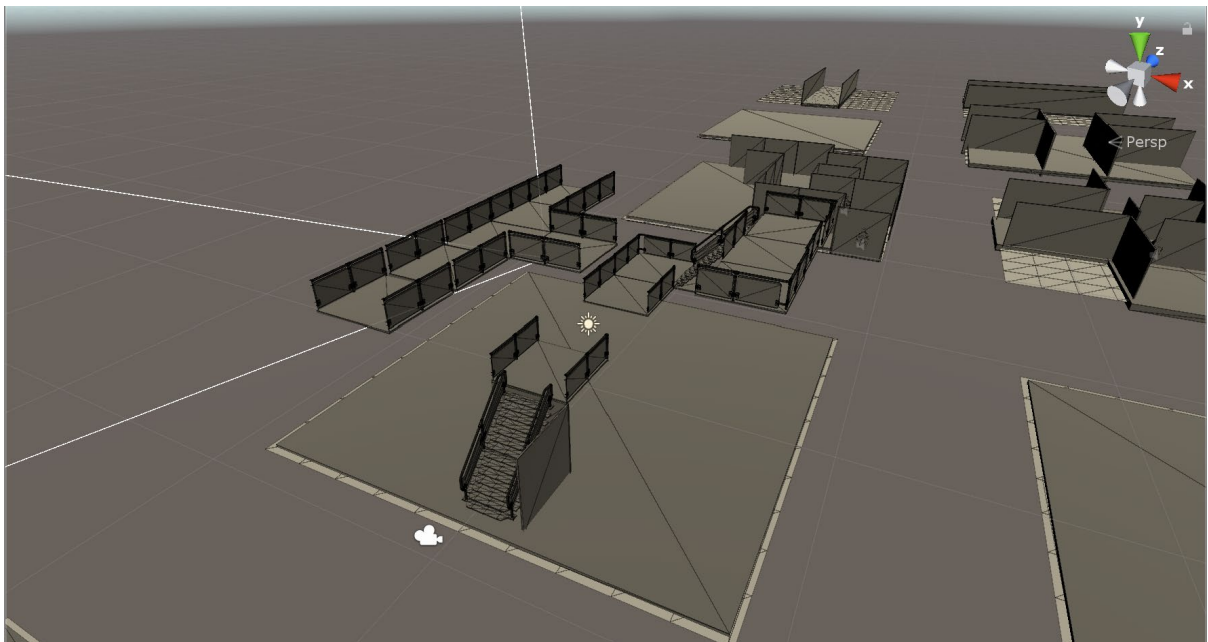
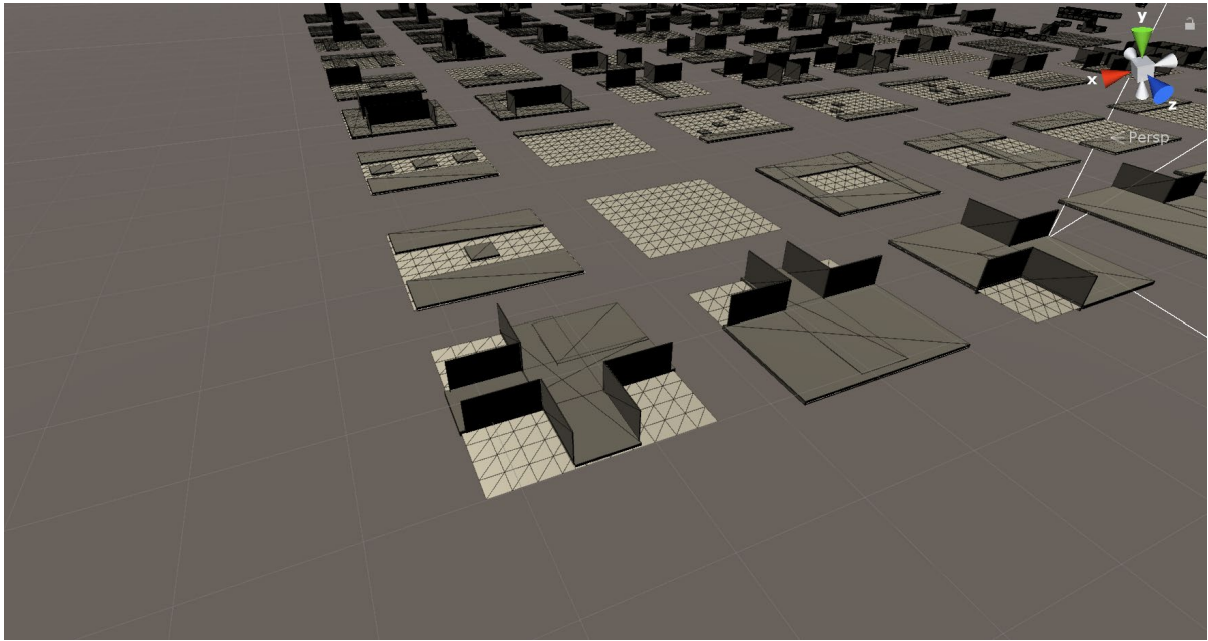
Further experiment with cells design

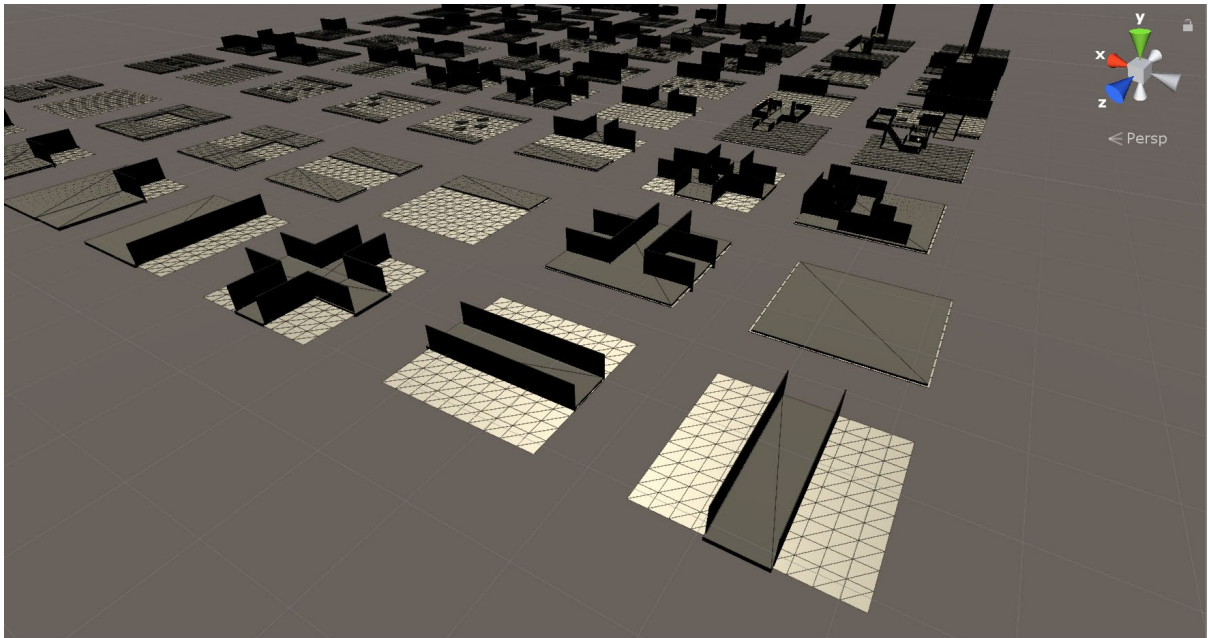
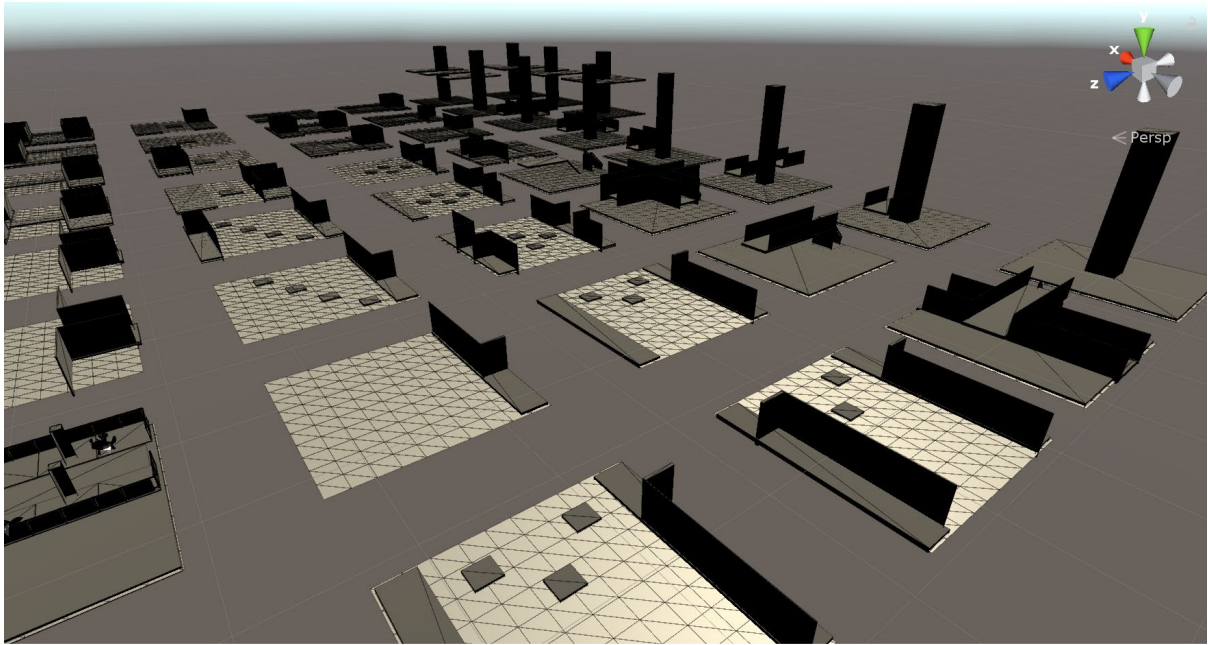


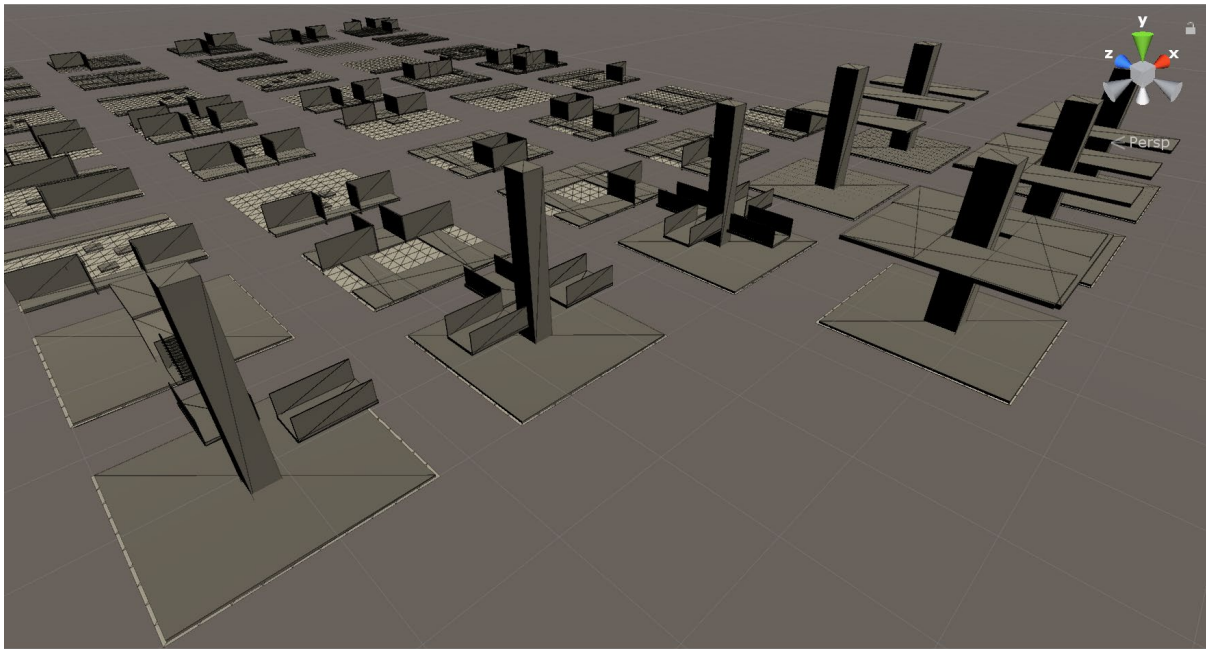
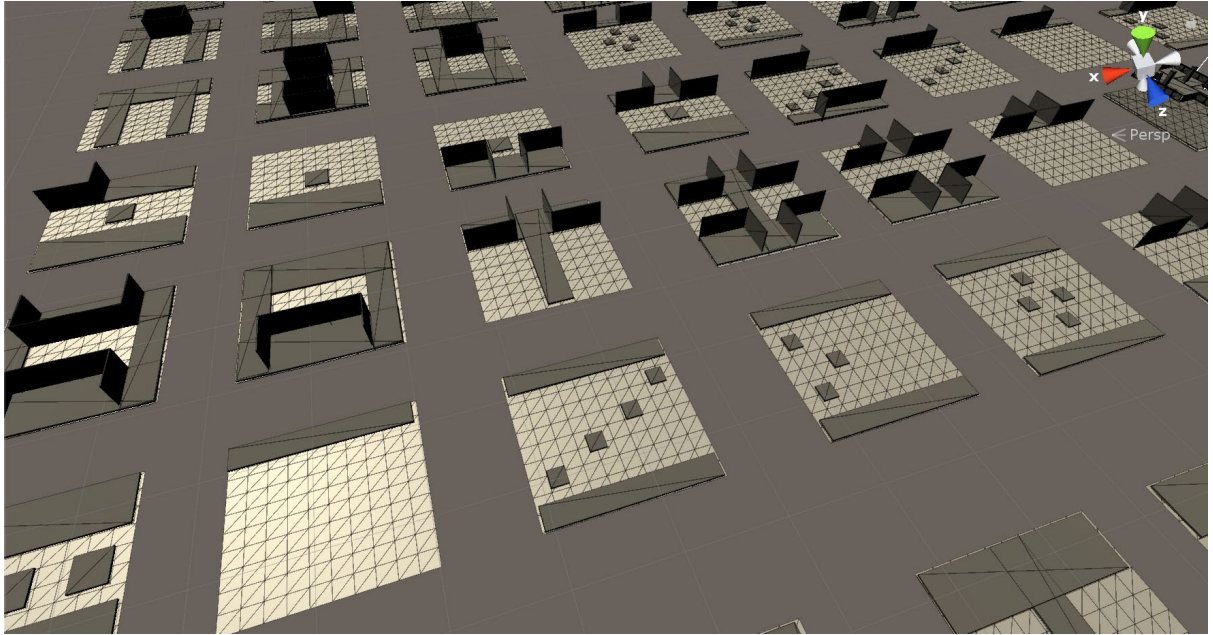
## Cells v2

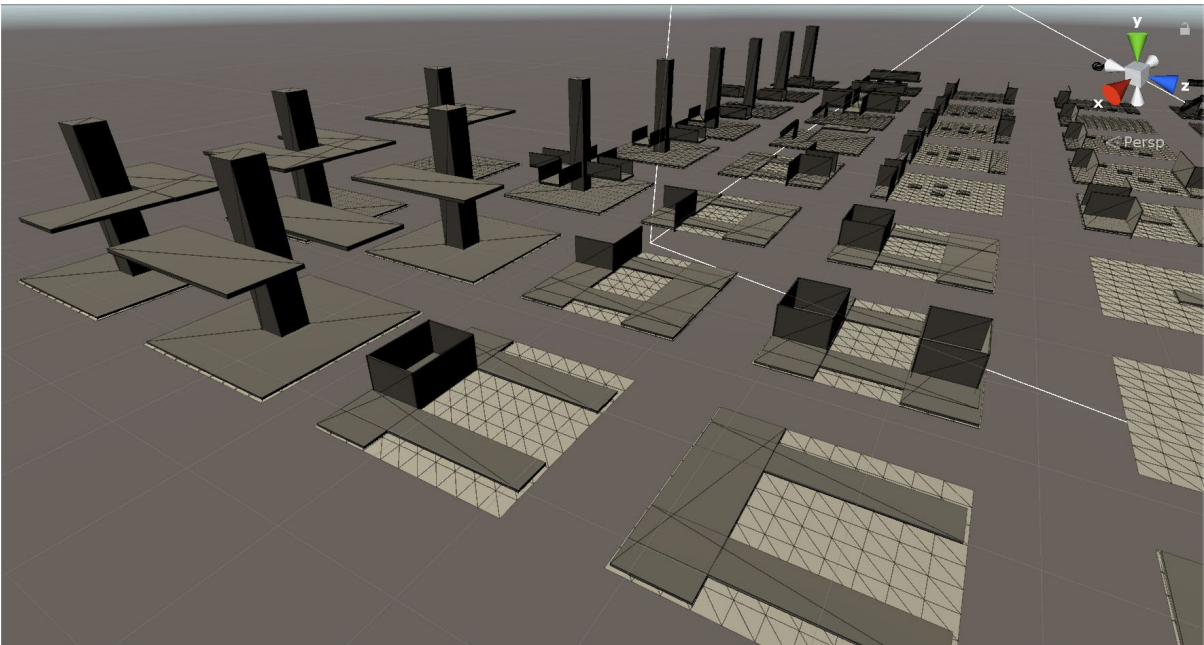
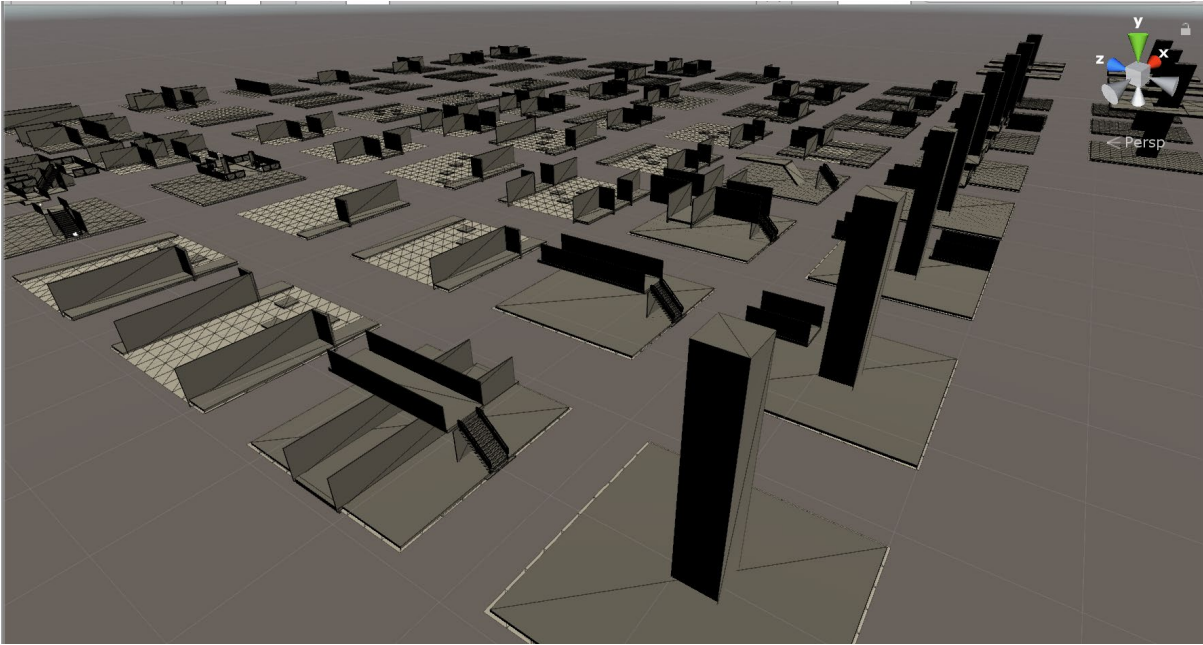
These cells were made in Unity with the initial modular assets provided by the art team. At this stage of development, the cells were changed to smaller ones, to accommodate a vision of complete modularity. To accomplish this vision, I designed 65 base cells to feed to the algorithm to see what kind of result we could receive or if they could at least work well together. The dimensions were 10x10x10. Compared to the V3 cells, the V2 are simpler, as I never got to design more advanced cells. But this design was quickly scrapped due to technical problems with the way the cells worked with the algorithm the programming team created.



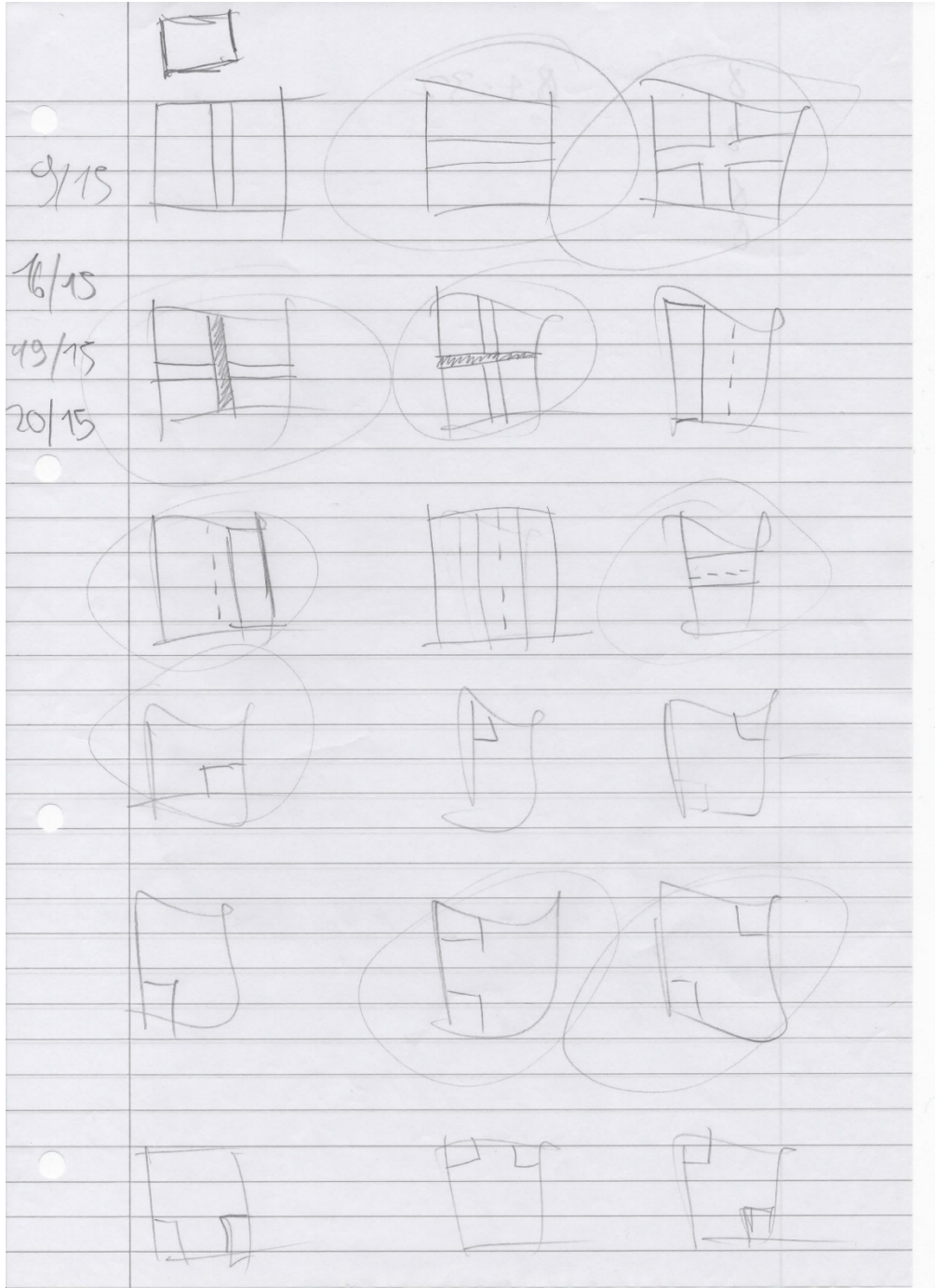




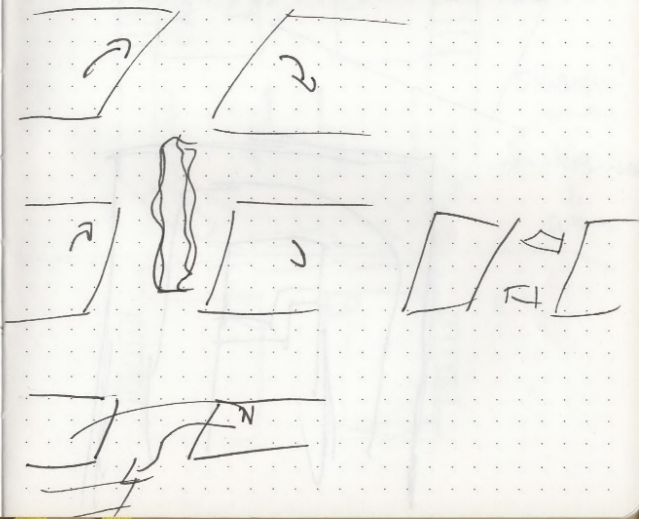
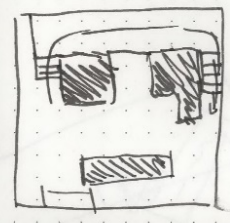
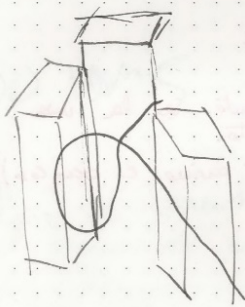




Developing cells V2

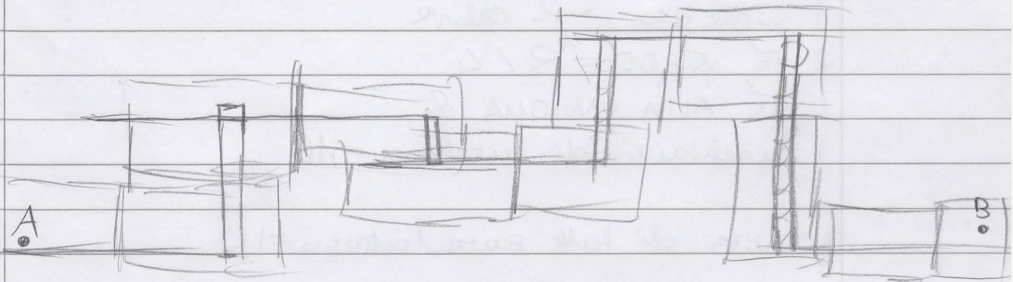
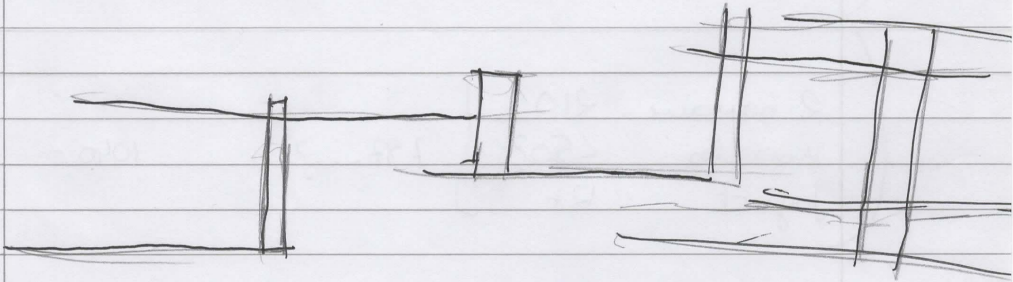
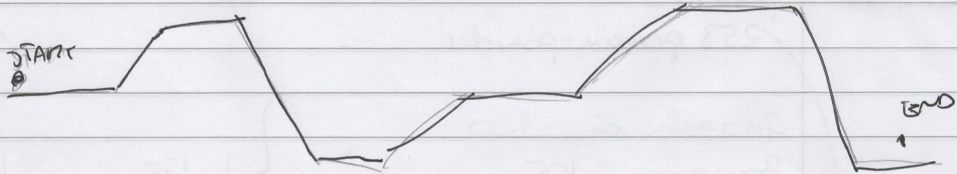


TASK	COMPLETED?
UDD UPDATE	FINISHED BY 15TH
ZDD UPDATE	" " 15TH
LENS	10 DIME, 15 SIMPAC DESIGN, 8 MORE BY 16/07
Velocity	86%
DEAD CON ANNU	76%
INENS	5%

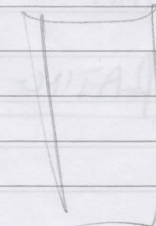
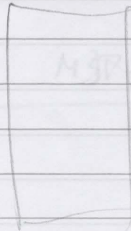
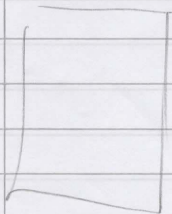
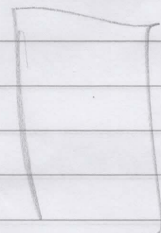
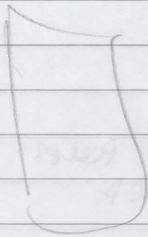
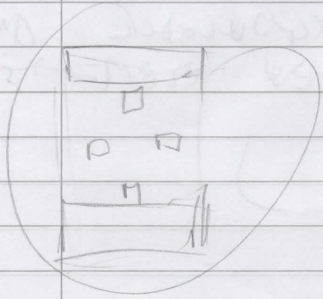
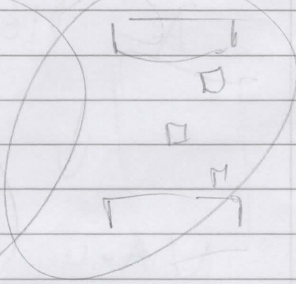
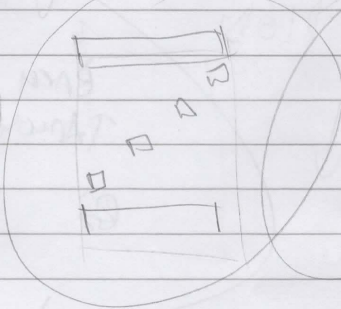
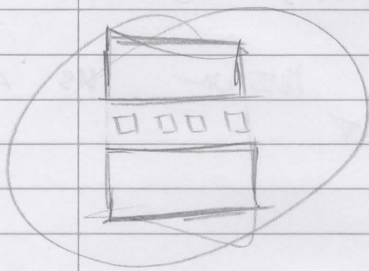
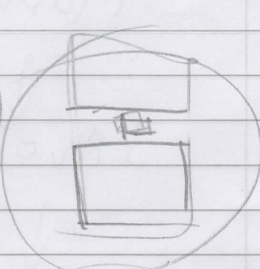
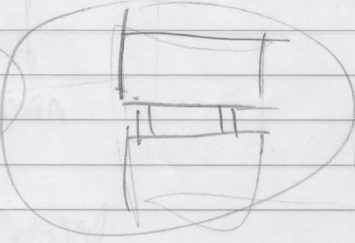
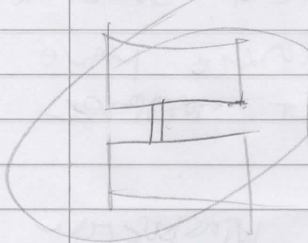
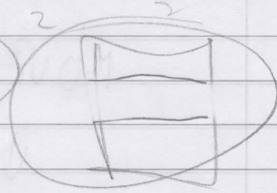
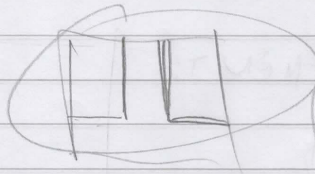
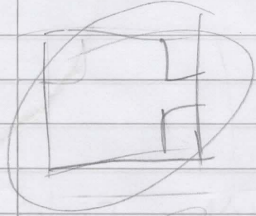


# JAMES VERTICALITY TAGMT

## SUBVIEW

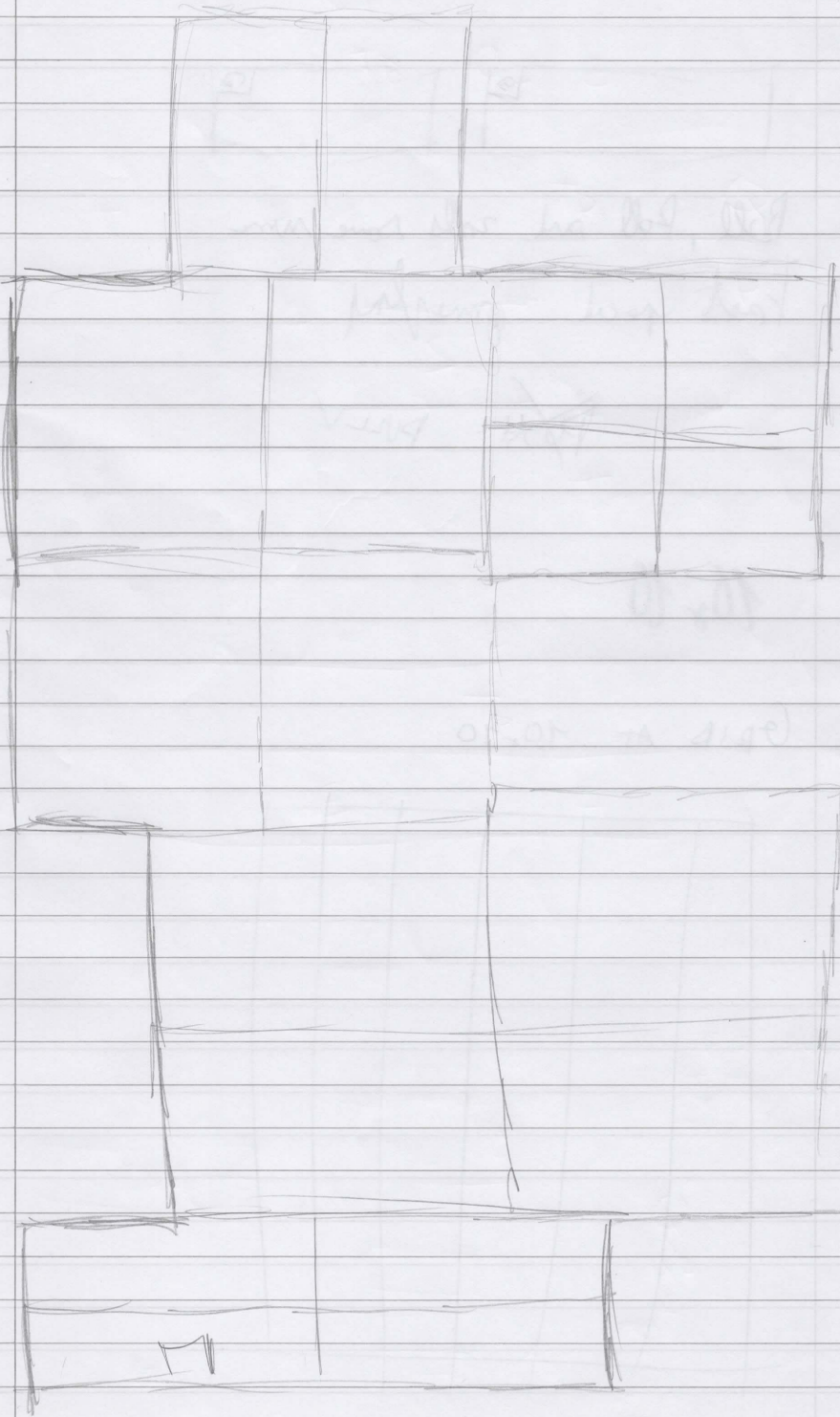


$60:2 = x \cdot 60$      $x = \frac{60 \cdot 60}{2} = 1800$

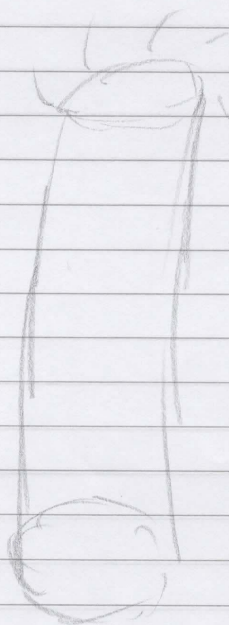


TOP DOWN

25/02/2020



	1	2	3	4	5	6	7	8	9	10
1										
2					\	\	\	\	\	
3		/	/	/	/	/	/	/	/	
4		/	/	/	/	/	/	/	/	
5			/	/	/	/	/	/	/	
6		/	/	/	/	/	/	/	/	
7							/	/	/	
8							/	/	/	
9							/	/	/	
10							/	/	/	



-MOVEMENT

	SIMPLE	EASY	MEDIUM	HARD
COMBAT				
SIMPLE				
EASY				
MEDIUM				
HARD				

photoshop

adobe cloud

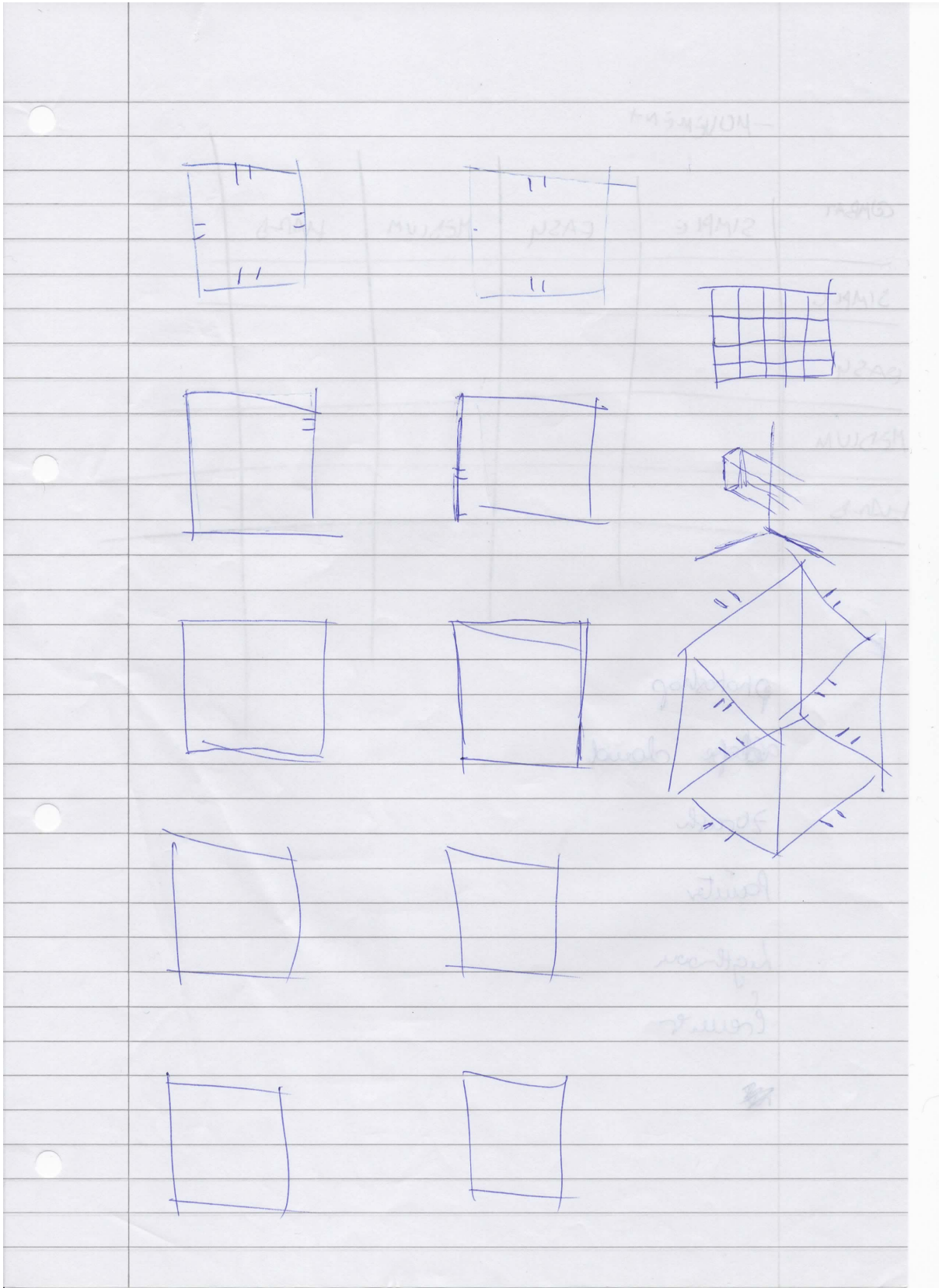
zbrush

painter

lightroom

premiere

~~sketch~~



## Cells V3

### Brief description of guidelines

Each cell must have the exact dimensions of 30 x 30 x 30 metres due to technical constraints (programmers told us to use these dimensions, since the algorithm could only handle these exact dimensions)

On average, cells have 2-3 levels of elevation. If time would have allowed it, we could have created cells with more levels of elevation, but for the time being, only a couple of cells would have suffice.

Game flow had to be consistent across cells. Levels were dense, players encountered new things every 3-6 seconds.

Relationships between player and map remains the same: heavy emphasis on player movement and navigation. Key point as always is the traversal at high speed.

Key abilities in the game that were required by the client (so they were an imperative to account for them while designing the cells) are:

- Wall running
- Sliding/Surfing
- Gravity thingy that pulls the player as a slingshot, utilizing the momentum to get in places

### Visual Reference:

- <https://www.artstation.com/artwork/OXWlv5>

### Verticality (method to change elevation)

- Stairs/industrial ramps
- Gravity thingy/swirly gravity pillars
- Jump (same as platforming)
- Ladders (gravity ladders just to avoid breaking the flow?  
<https://www.youtube.com/watch?v=R6i7oSwllto> this could be a visual inspiration, (mechanically is too slow for the desired flow).
- Teleport or quick travel (tied ambient to gameplay? A hub like metro system?)
- Dropping down? (bloodborne style <https://youtu.be/WrEw3YWdN04?t=45>)
- Platforming (same as jump), could be further explored to moving platforming, excavators and other environmental pieces
- Elevators? (similar to half-life 1 head crab elevator see reference at <https://youtu.be/yoPsRARx2wY>)
- Zip lines/random cables? (similar to Titanfall ziplines [https://youtu.be/C3BtOeQO\\_n0](https://youtu.be/C3BtOeQO_n0))

### Cell Design Rules:

Entrances/exits should be placed at 5m, 10m, 15m, 20m, 25m increments

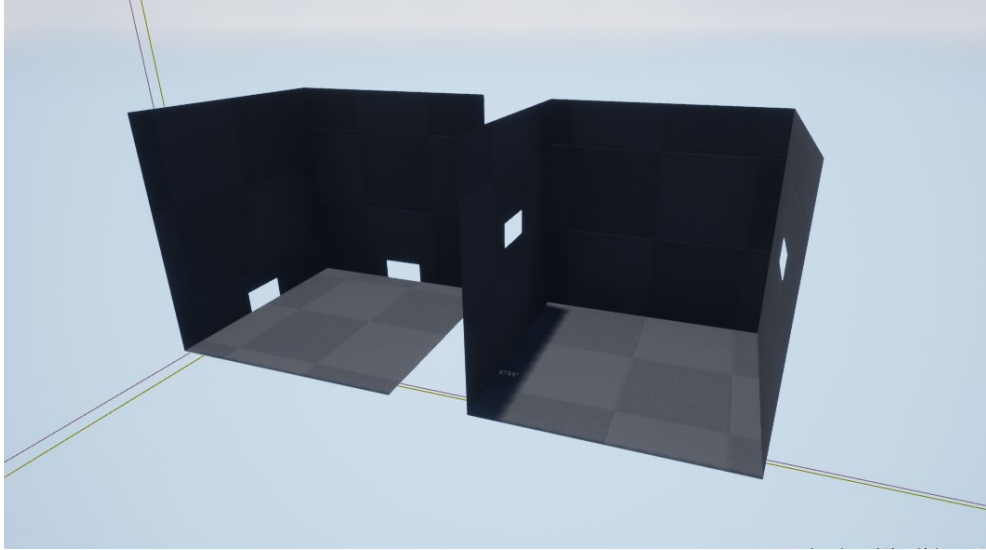
### Sizes of assets (metres):

- Concrete walls/floors: 3 x 3 x 0.275

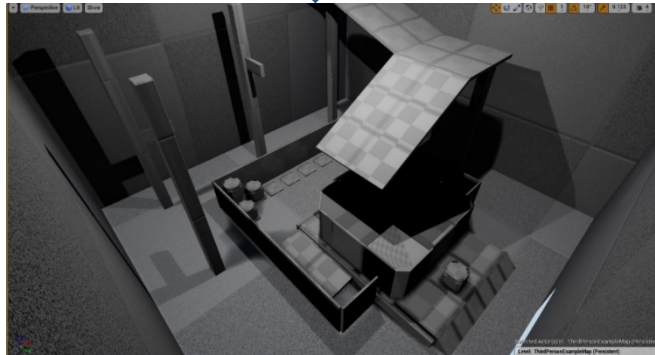
- Metal walls/floors: 3 x 3 x 0.14
- Concrete pillars: 0.8 x 0.8 x 3

Each final cell is made of two elements:

- Core cell layout. This core layout is a cube with two holes that are the entry/exit



- Props. Assets made from the art team





Selected Actor(s) in:

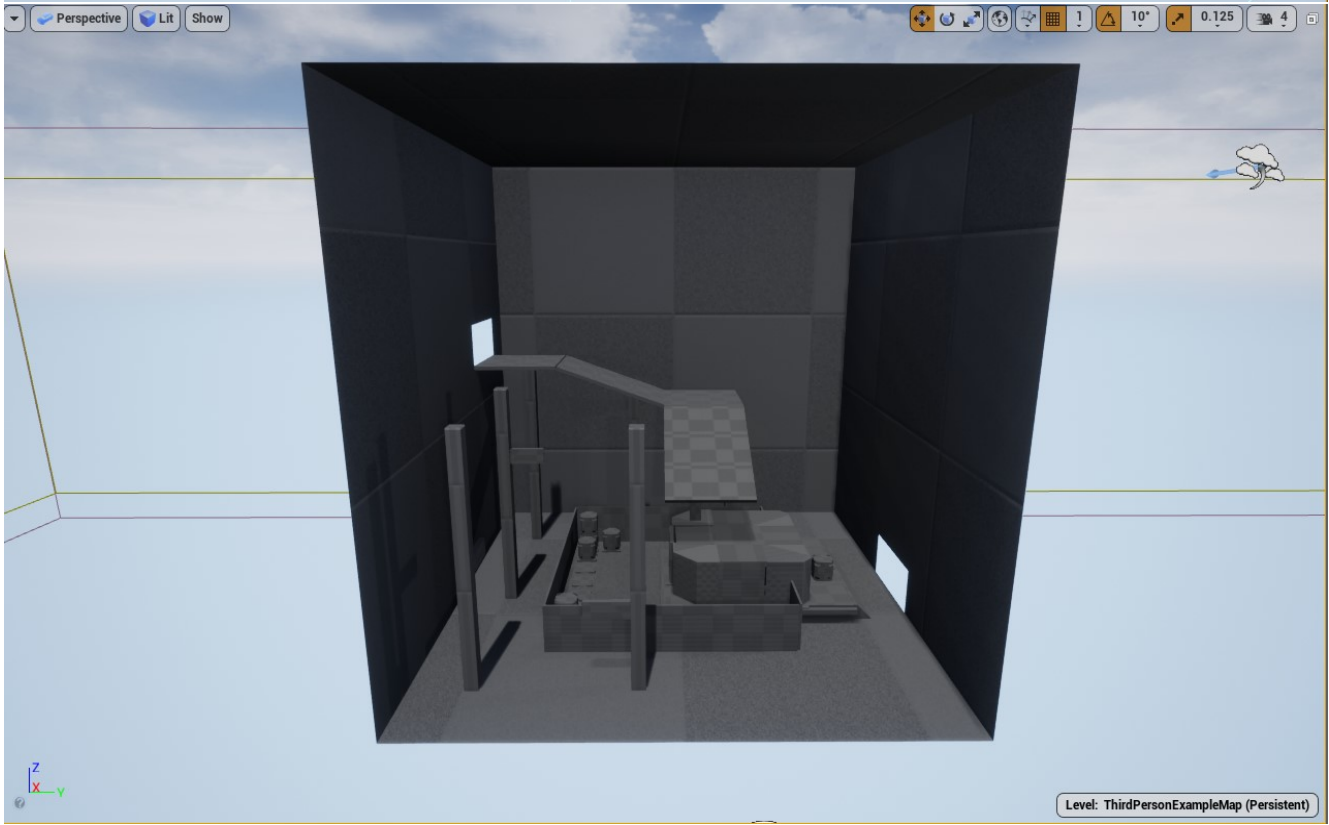
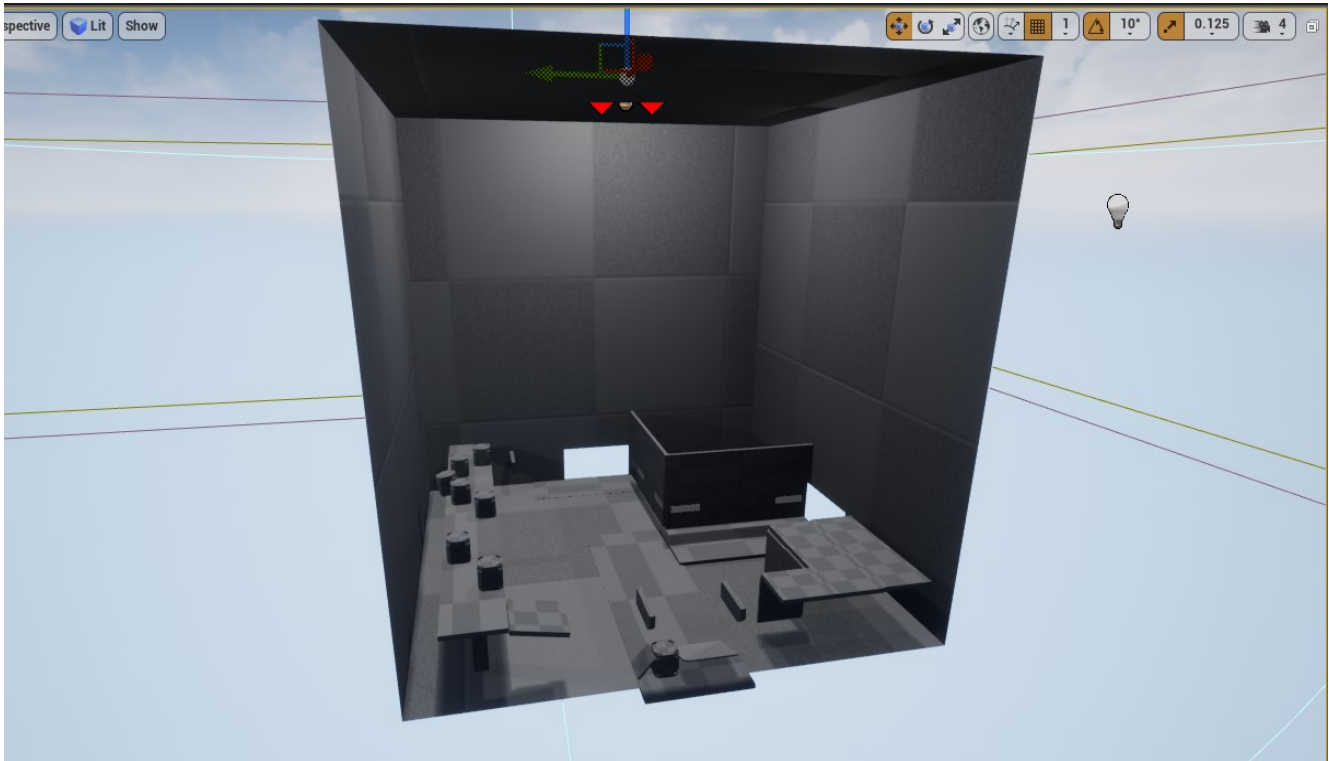
Level: ThirdPersonE

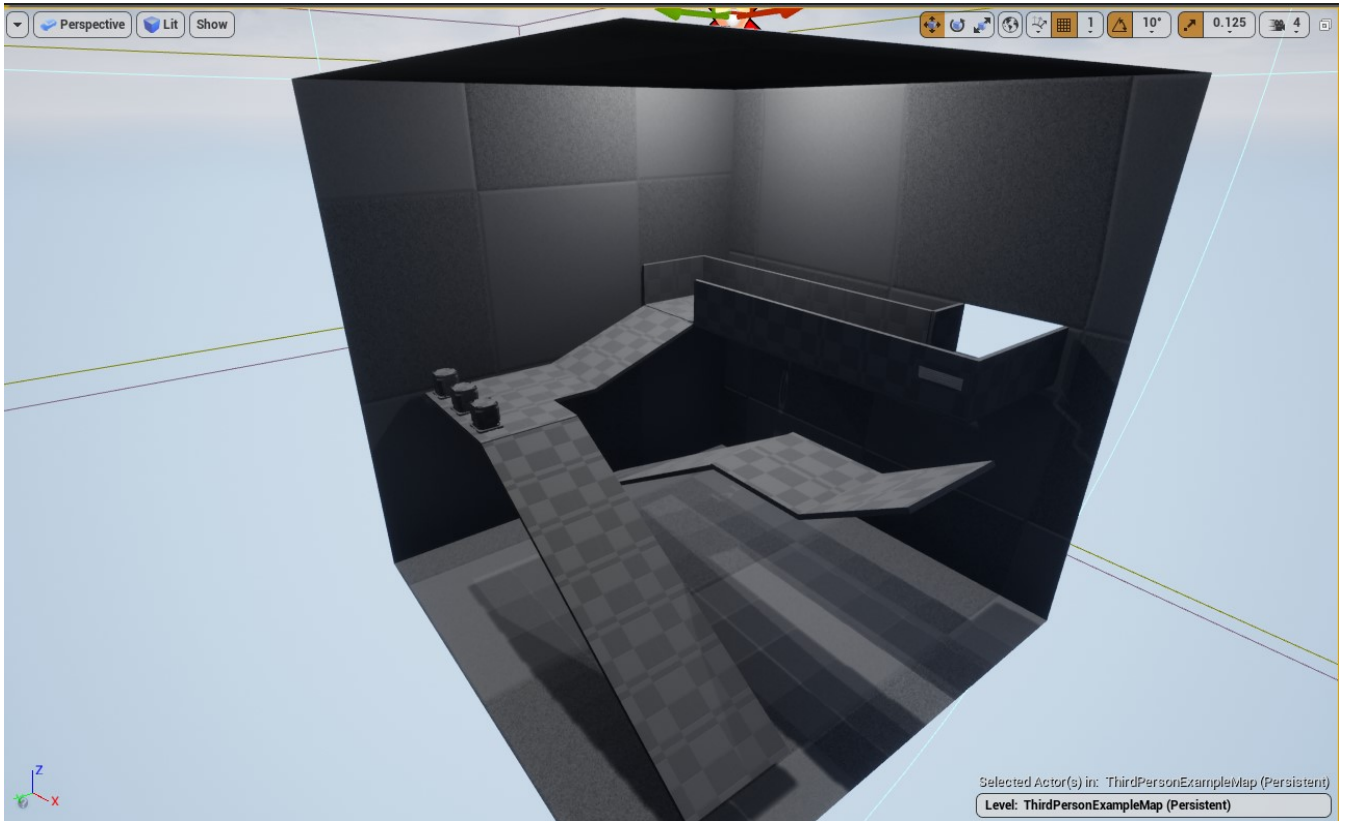
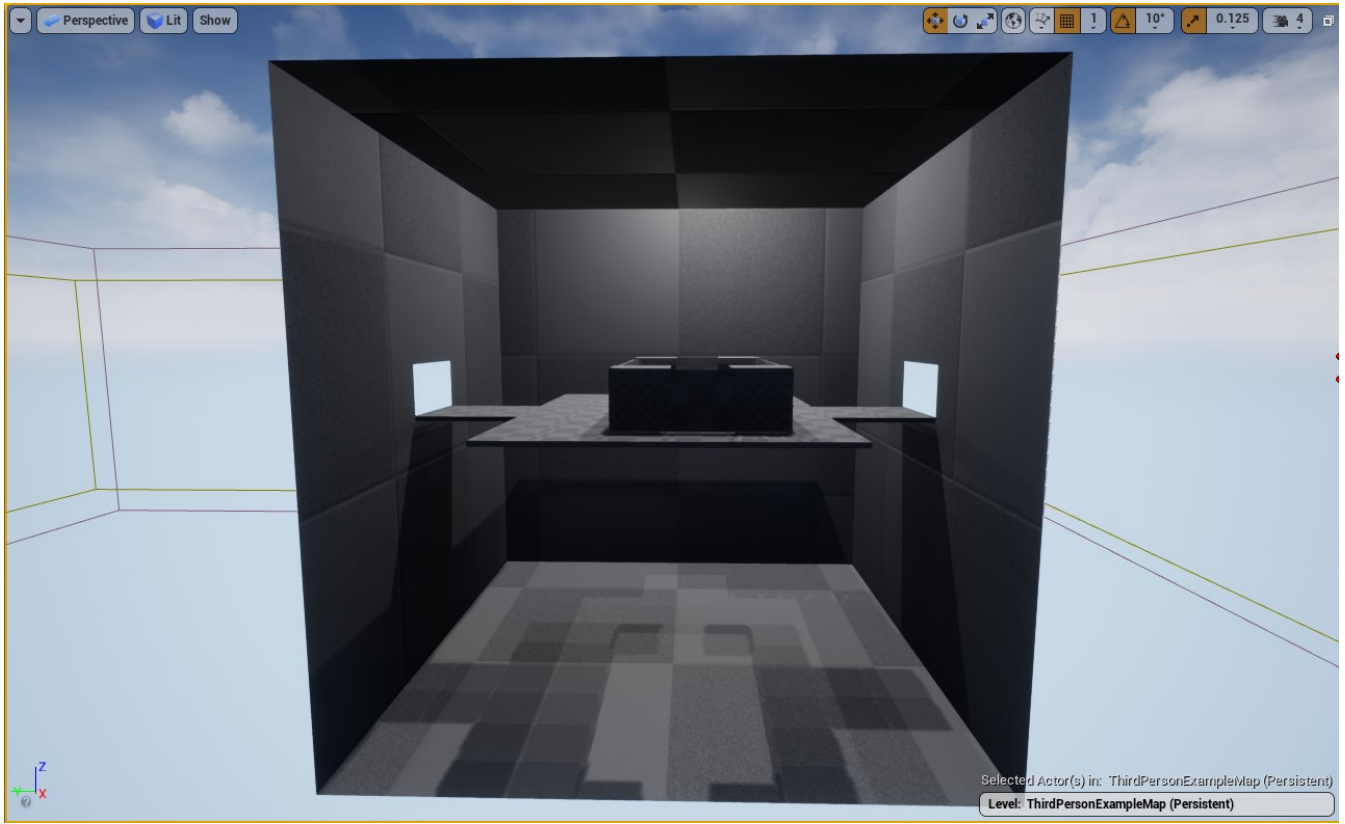
Perspective Lit Show

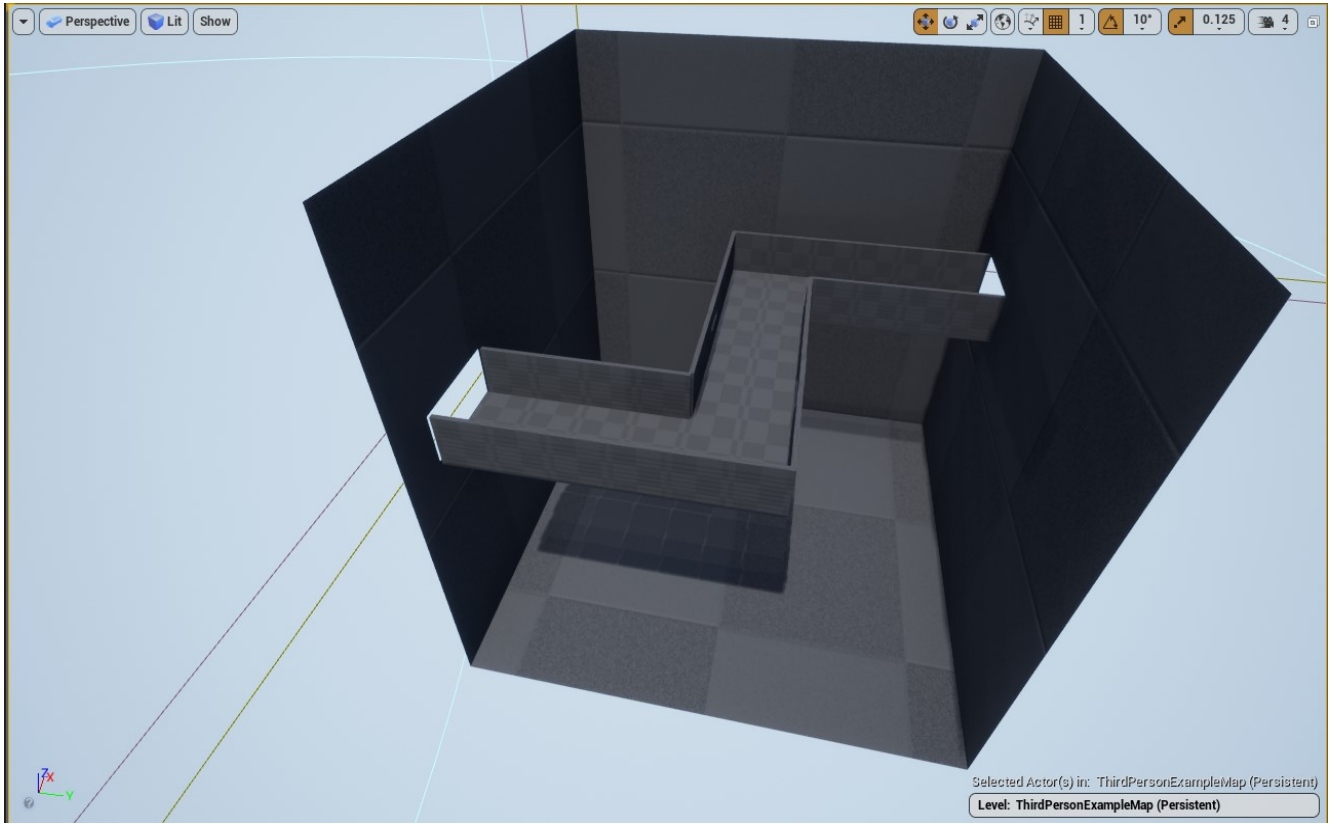


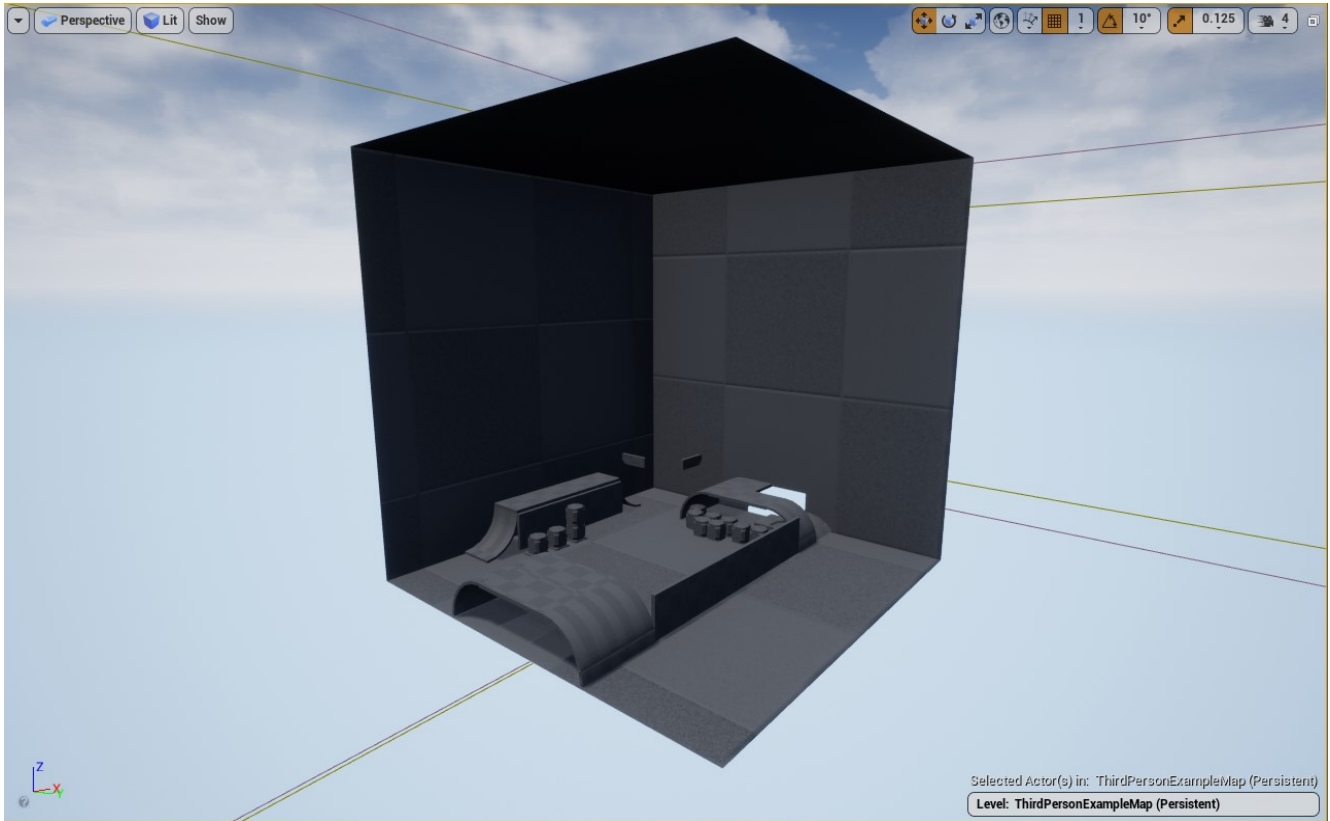
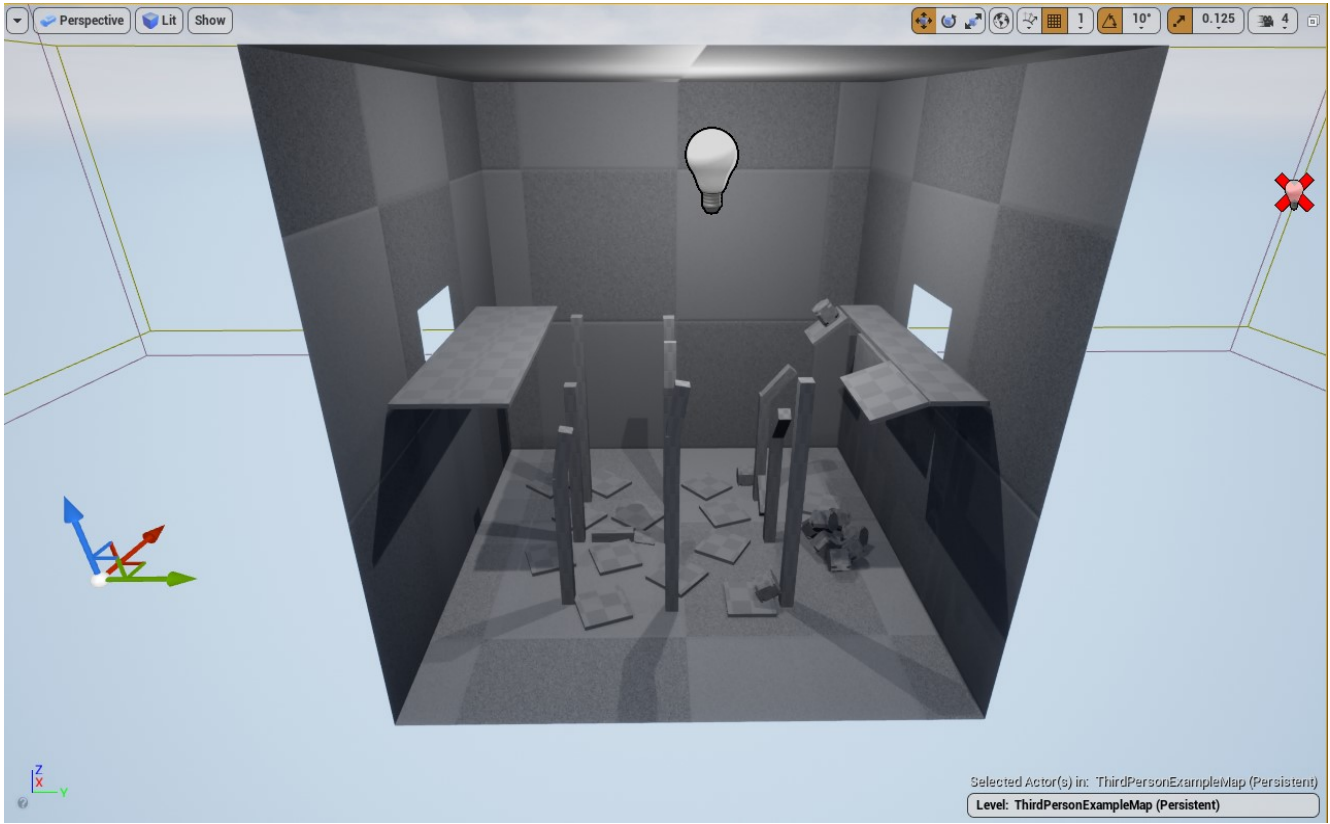
Selected Actor(s) in: ThirdPersonExampleMap (Persistent)

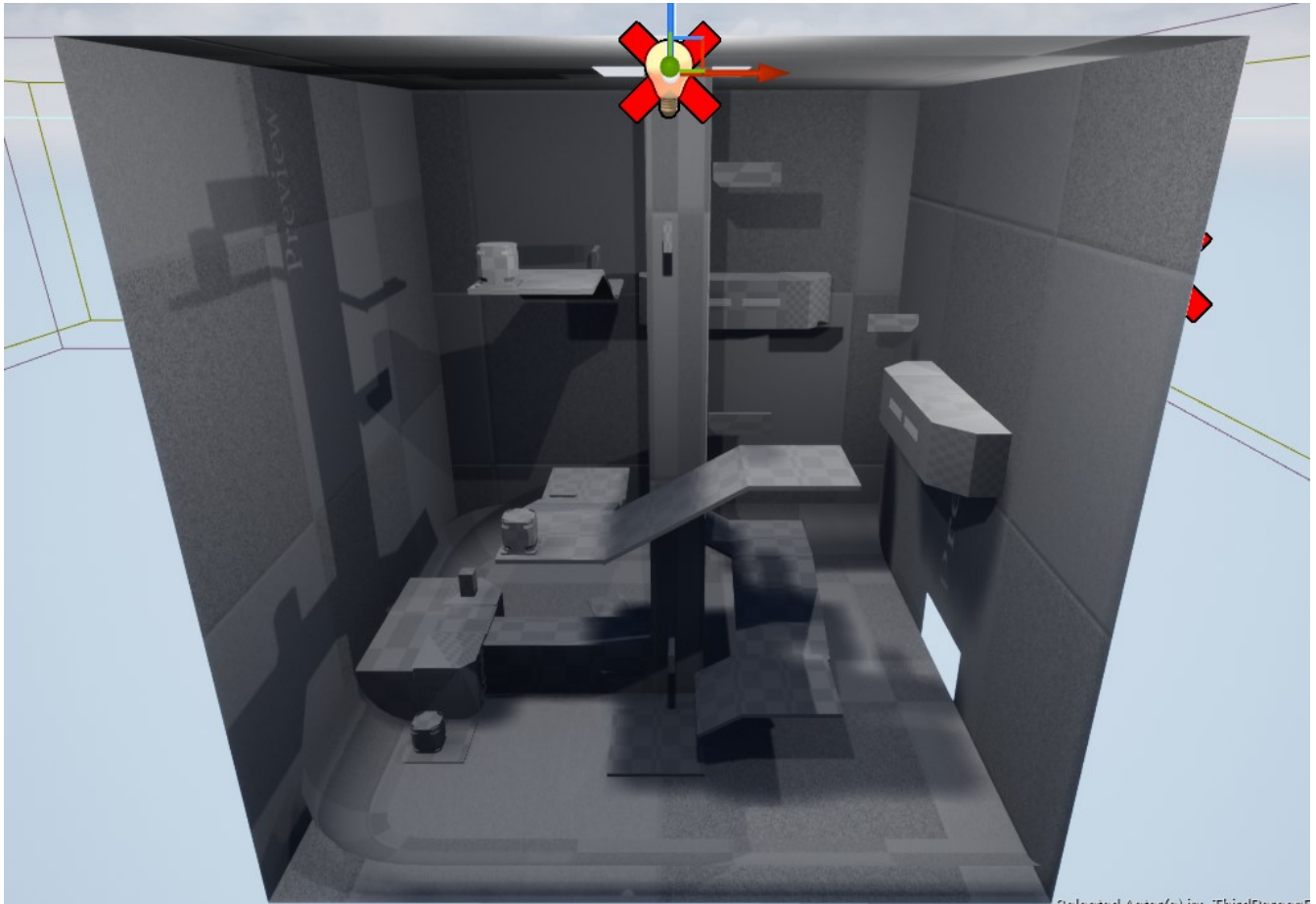
Level: ThirdPersonExampleMap (Persistent)







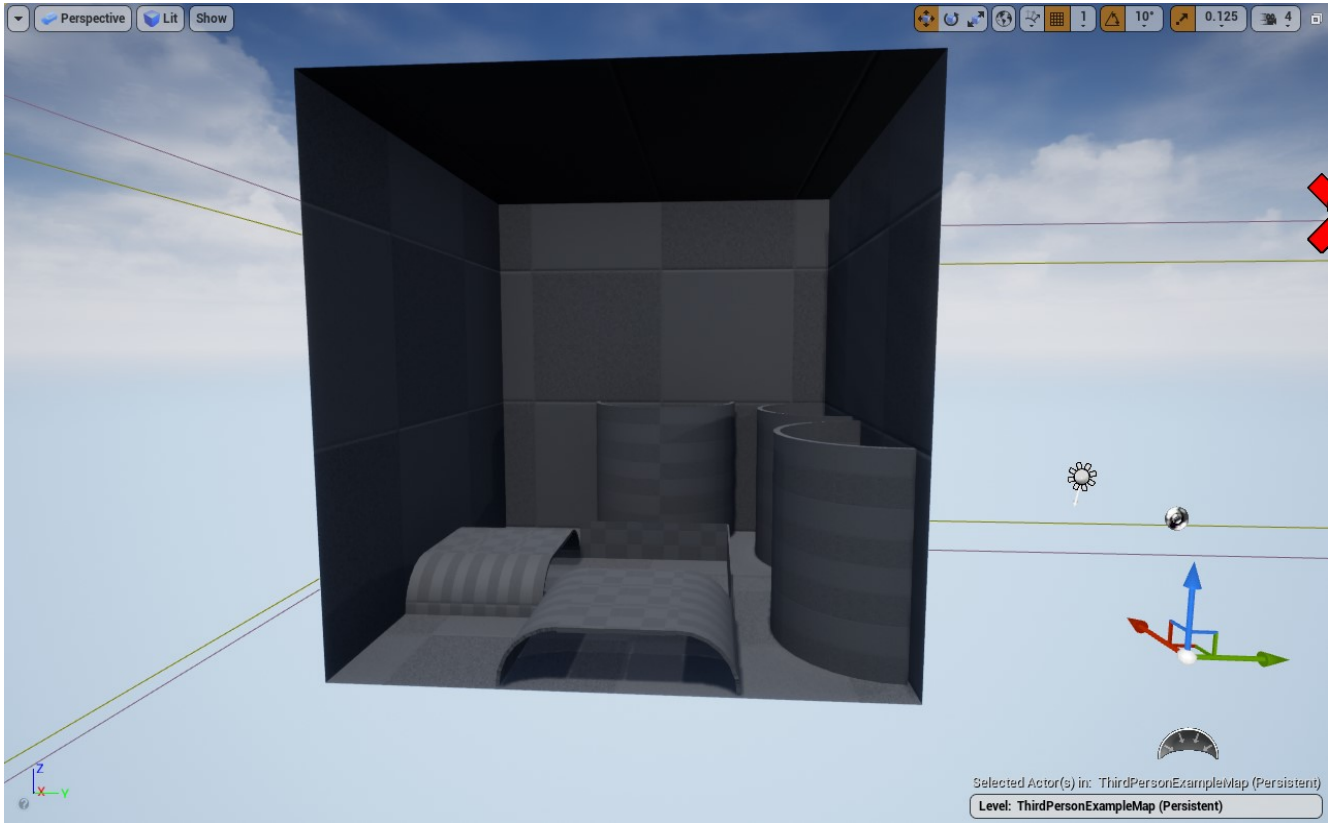




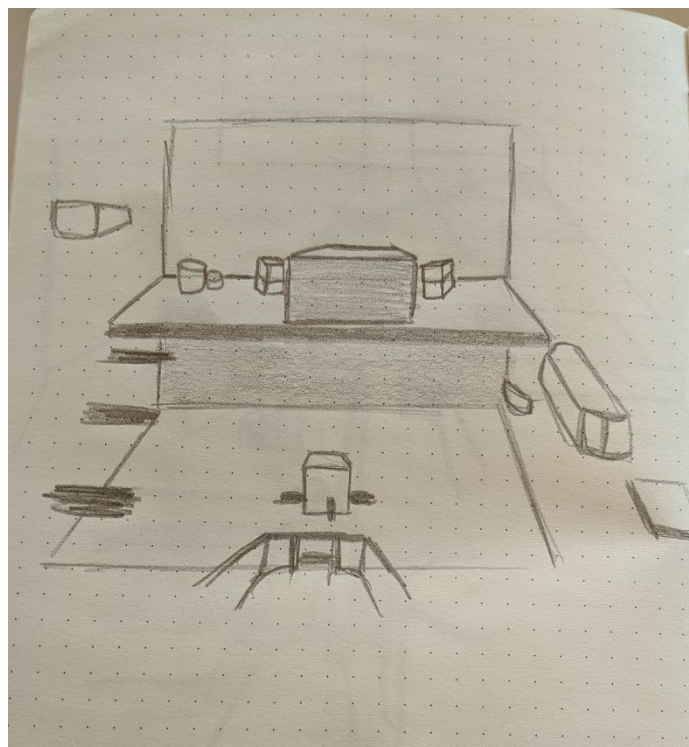
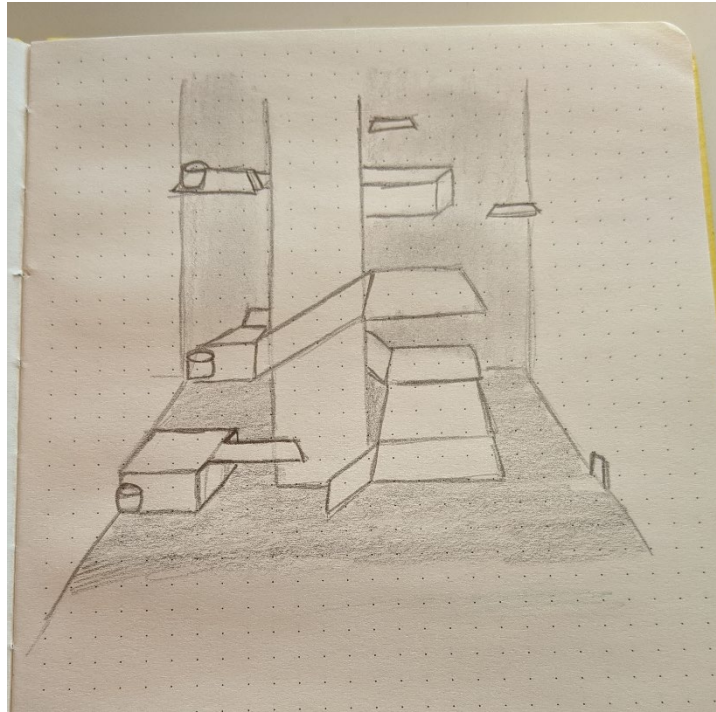
Selected Actor(s) in: ThirdPersonE  
Level: ThirdPersonExampleMap / D

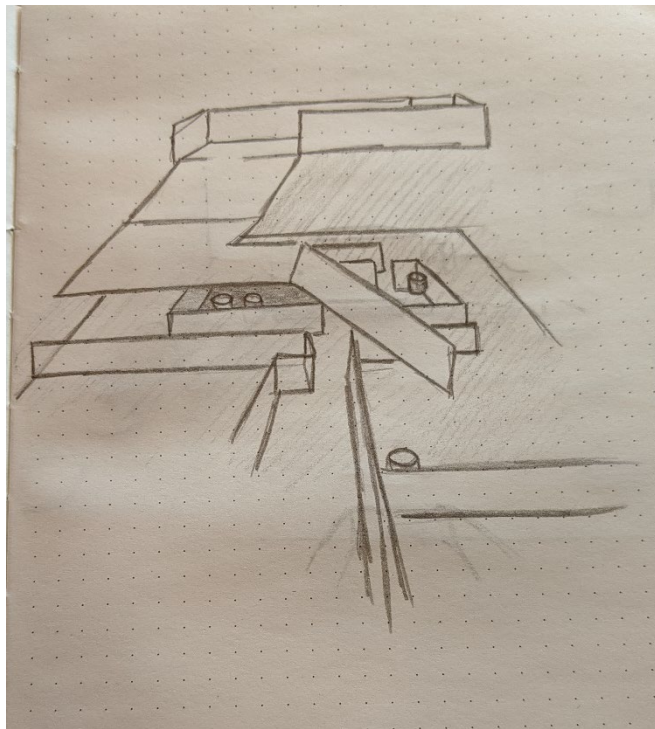
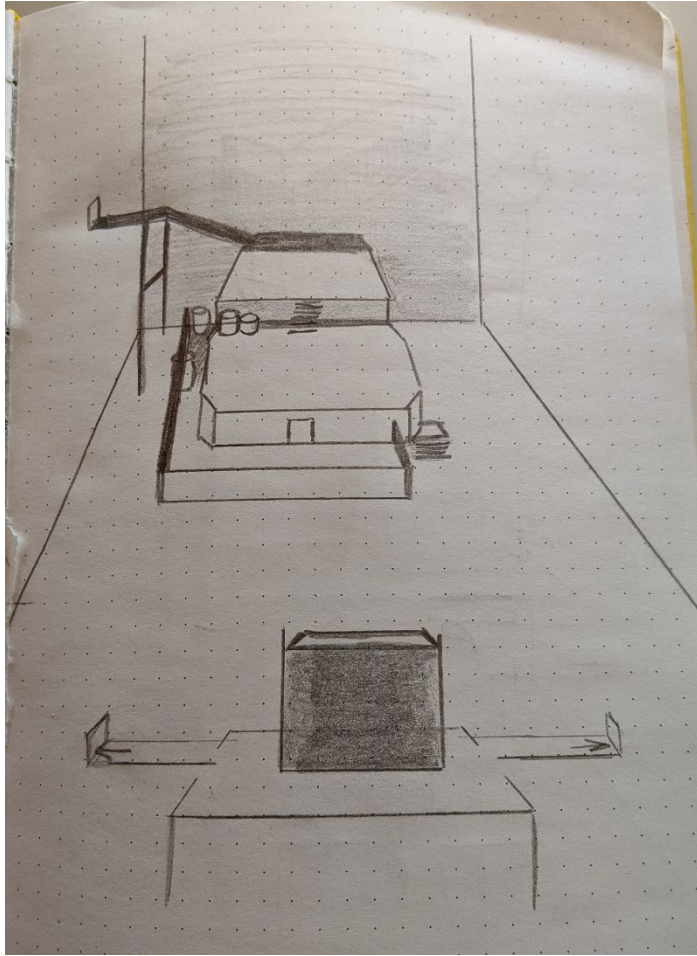
Show

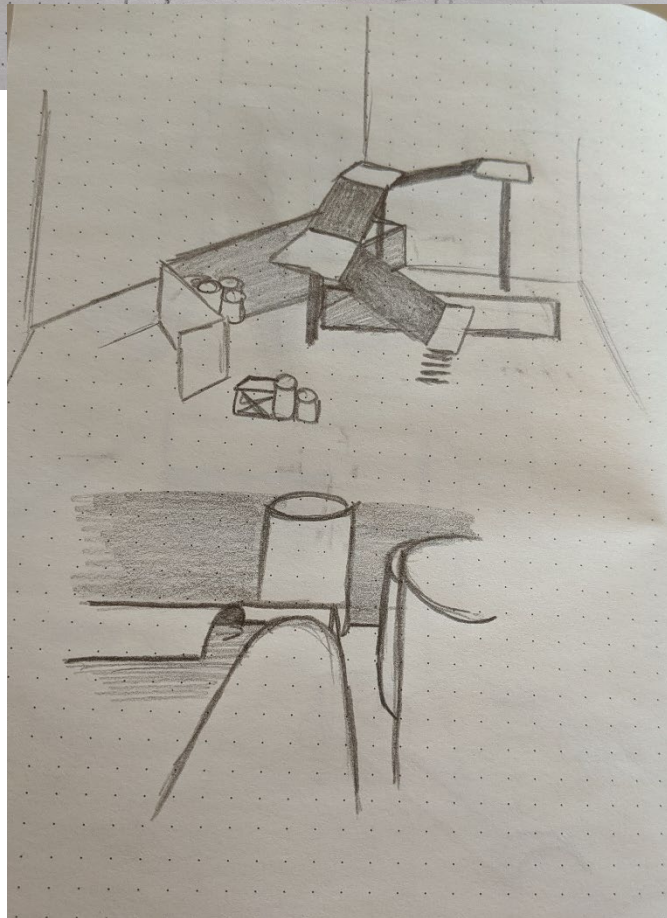
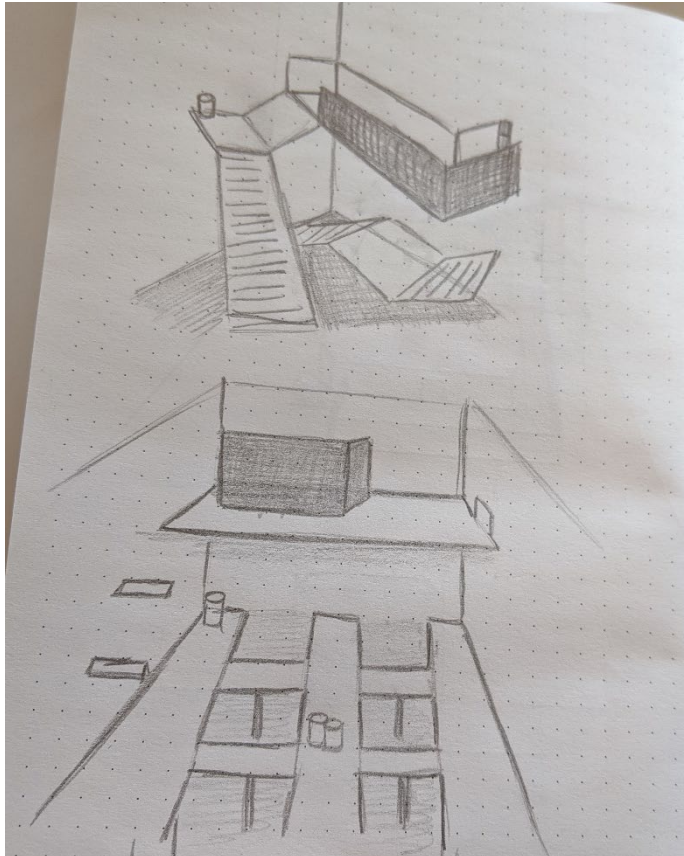


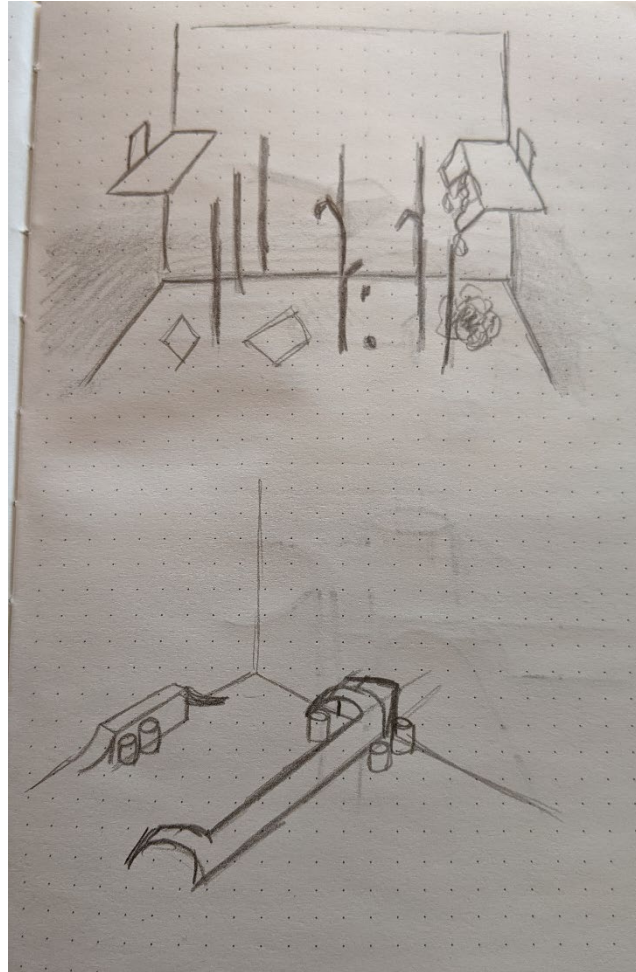


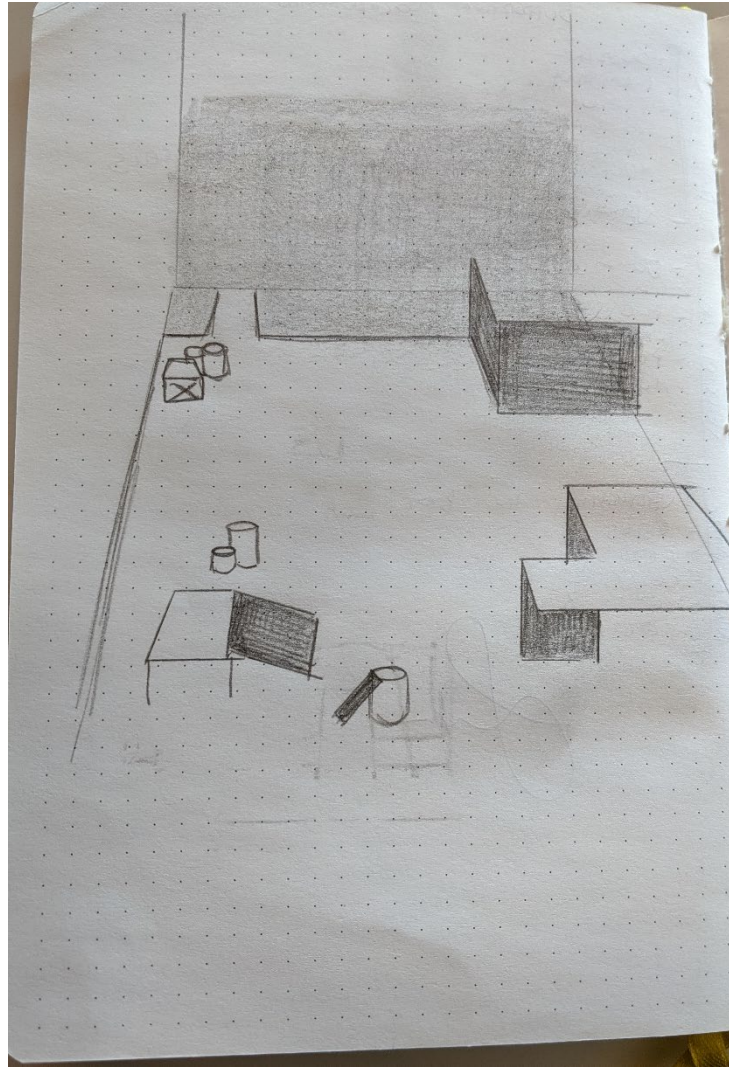
Developing cells V3



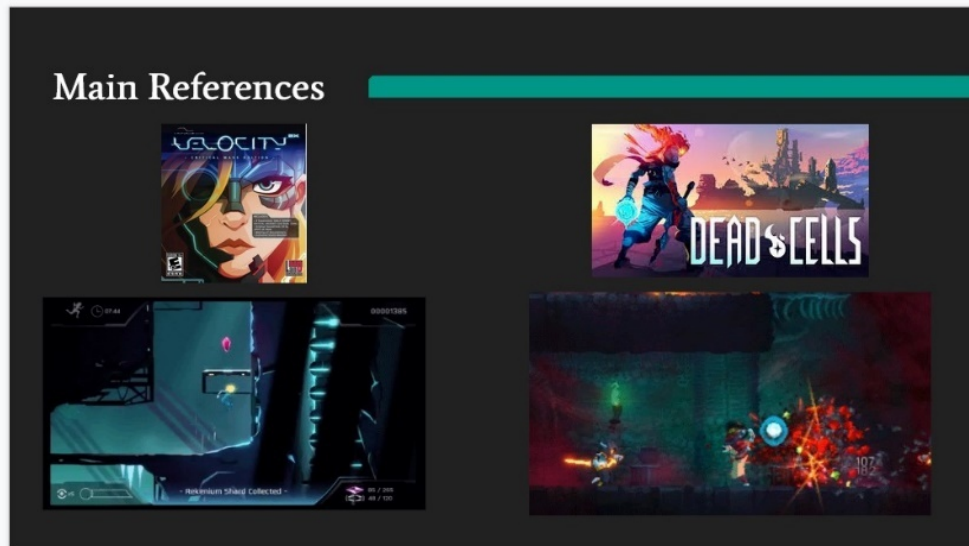








## Peer presentation



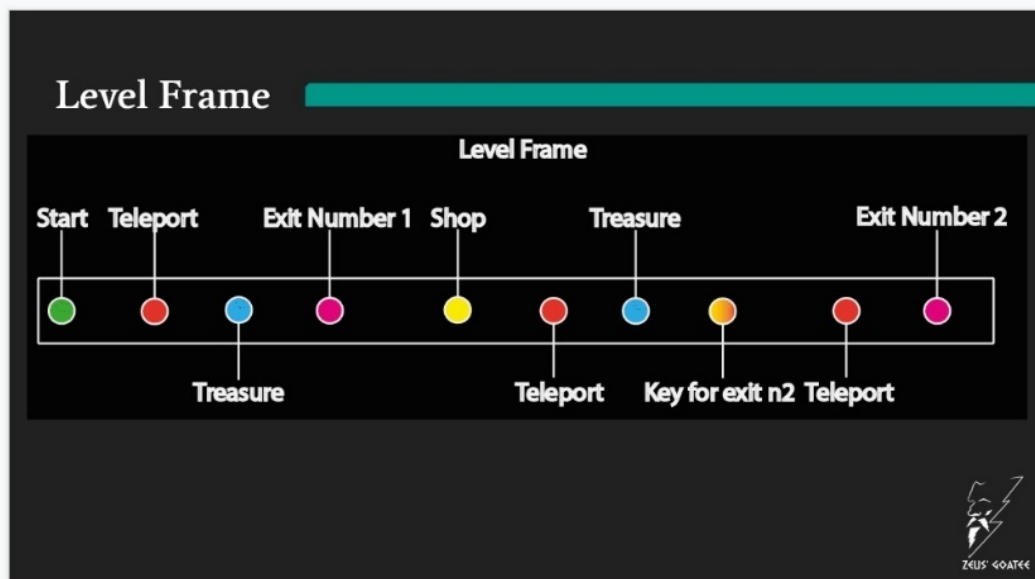
As we said before, the client wants dead cells, but in 3D and mixed with velocity 2x.

To refresh memory, dead cells is a 2D rogue-vania with heavy emphasis on fast traversal of the map and an even higher fast paced combat. Levels are procedurally generated with a castlevania influence to it.

So before start working on anything, we had to play dead cells to get a good grasp of how dead cells felt and how we could translate the progression, flow and the constant feeling of high octane gameplay in the third dimension.

Another reference for the flow is Velocity 2X. Velocity is one of the previous games from the client, with fast paced gameplay and flow quite similar to the one found in dead cells.

With these two reference, I analyzed them to get a good grasp of what the client wants and how we can deliver it at our best ability.

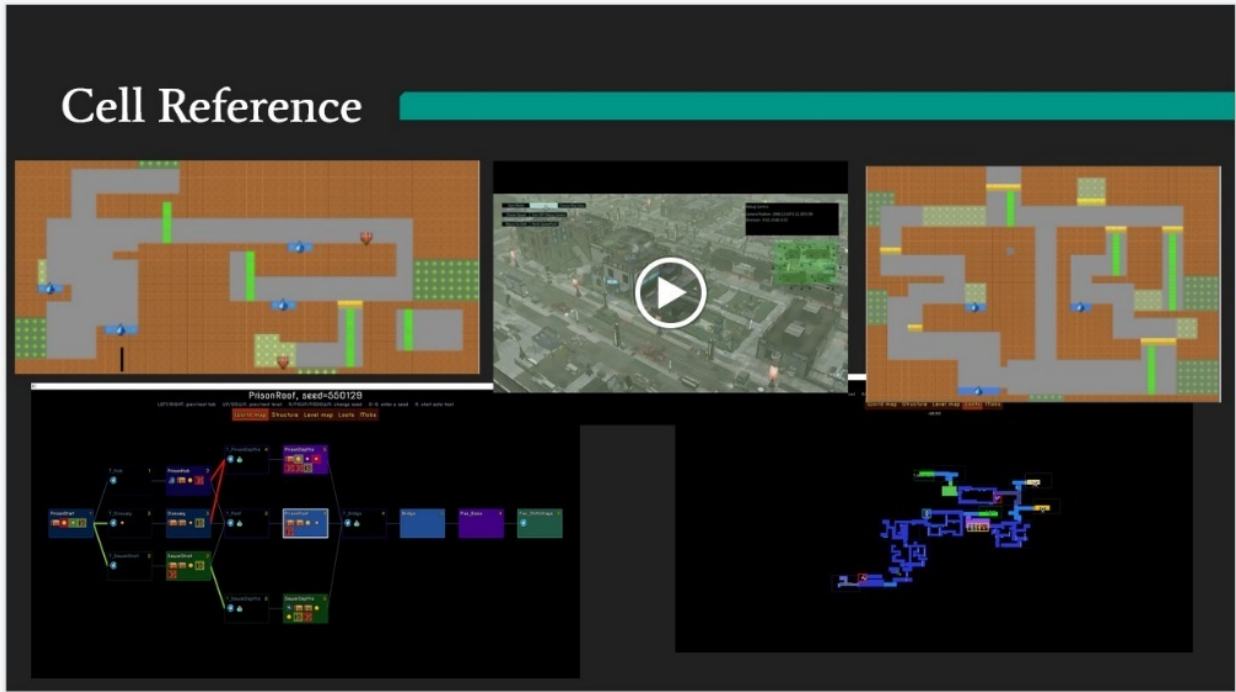


Ok, so this is the level frame. it's a visual representation of how levels are going to flow and the progression that the player is expected to make. Every level frame is made out of various elements such as shops, teleporter, etc...

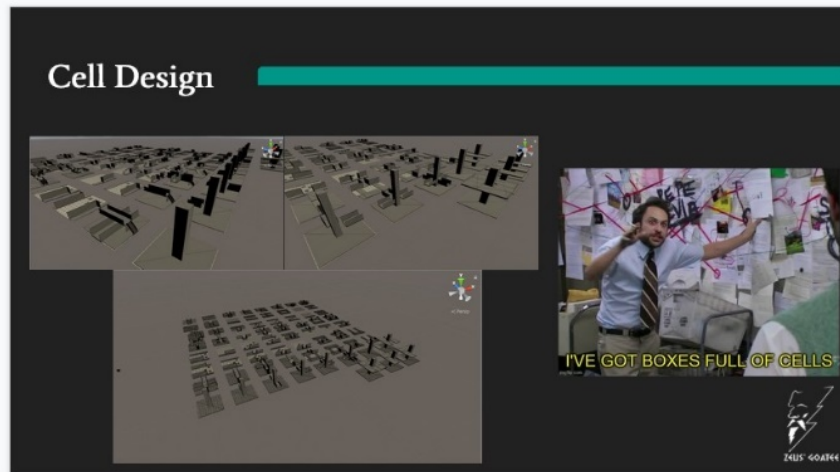
Each element is a rule for the algorithm, which is going to dictate how to make the procedural level.

Thanks to this modular approach, swapping elements around will change the player experience. For example, in a hard levels, the shop can be placed at the end while in easier levels shops can be scattered around the level to ease the player's journey

Thanks to these various elements, the player will experience each level without being fatigued by the repetition, and each level will play and feel different, engaging the player and saving both time and resources



Generation of level is very similar to the one used in dead cells.  
 A good inspiration for cells design are xcom 2, dead cells and phoenix point.



One of the main factors that we must keep in mind is that the length and extension of the level is unknown/ may varies during development time, so the biggest question to answer is how big and how extensive these cells/chunks are going to be. The dimension of these chunks is important as they establish the scale of the level, so deciding blindly without giving any thoughts to it is inconsiderate. We came to the conclusion that due to the fact that the gameplay is fast paced, the cells are going to be prevalently small and squared. There are going to be 3 type of cells, 1x1, 4x4 and 8x8.

The first series of cells are going to develop only on two dimensions not taking into account verticality. The element of verticality will be implemented in the second and third series. The second series will concentrate in merging the designs of the first series of cells with verticality, without being the only focus (the player will note the verticality and maybe uses it, but it won't be the main mechanic that the player will utilize in that space). The third series will concentrate on verticality and the relationship between the player and the environment.

Key abilities that they mentioned they wanted in the game (from example given on discord or brief):

- Wall running
- Slides
- Gravity thingy that pulls the player as a slingshot, utilizing the momentum to get player in places he otherwise

**First slide:** As we said before, the client wants dead cells, but in 3D and mixed with velocity 2x.

To refresh memory, dead cells is a 2D rogue-Vania with heavy emphasis on fast traversal of the map and an even higher fast paced combat. Levels are procedurally generated with a Castlevania influence to it.

So before start working on anything, we had to play dead cells to get a good grasp of how dead cells felt and how we could translate the progression, flow and the constant feeling of high-octane gameplay in the third dimension.

Another reference for the flow is Velocity 2X. [REDACTED], with fast paced gameplay and flow quite similar to the one found in dead cells.

With these two references, I analysed them to get a good grasp of what the client wants and how we can deliver it at our best ability.

**Second slide:** Ok, so this is the level frame. it's a visual representation of how levels are going to flow and the progression that the player is expected to make. Every level frame is made out of various elements such as shops, teleporter, etc...

Each element is a rule for the algorithm, which is going to dictate how to make the procedural level.

Thanks to this modular approach, swapping elements around will change the player experience. For example, in a hard level, the shop can be placed at the end while in easier levels shops can be scattered around the level to ease the player's journey

Thanks to these various elements, the player will experience each level without being fatigued by the repetition, and each level will play and feel different, engaging the player and saving both time and resources

**Third slide:** Generation of level is very similar to the one used in dead cells.

A good inspiration for cells design are XCOM 2, dead cells and phoenix point.

**Fourth Slide:** One of the main factors that we must keep in mind is that the length and extension of the level is unknown/ may varies during development time, so the biggest question to answer is how big and how extensive these cells/chunks are going to be. The dimension of these chunks is important as they establish the scale of the level, so deciding blindly without giving any thoughts to it is inconsiderate. We came to the conclusion that due to the fact that the gameplay is fast paced, the cells are going to be prevalently small and squared. There are going to be 3 type of cells, 1x1, 4x4 and 8x8.

The first series of cells are going to develop only on two dimensions not taking into account verticality. The element of verticality will be implemented in the second and third series. The second series will concentrate in merging the designs of the first series of cells with verticality, without being the only focus (the player will note the verticality and maybe uses it, but it won't be the main mechanic that the player will utilize in that space). The third series will concentrate on verticality and the relationship between the player and the environment.

Key abilities that they mentioned they wanted in the game (from example given on discord or brief):

- Wall running
- Slides

- Gravity thingy that pulls the player as a slingshot, utilizing the momentum to get player in places he otherwise

## Documentation

I did a part of the GDD, here's the part I did:

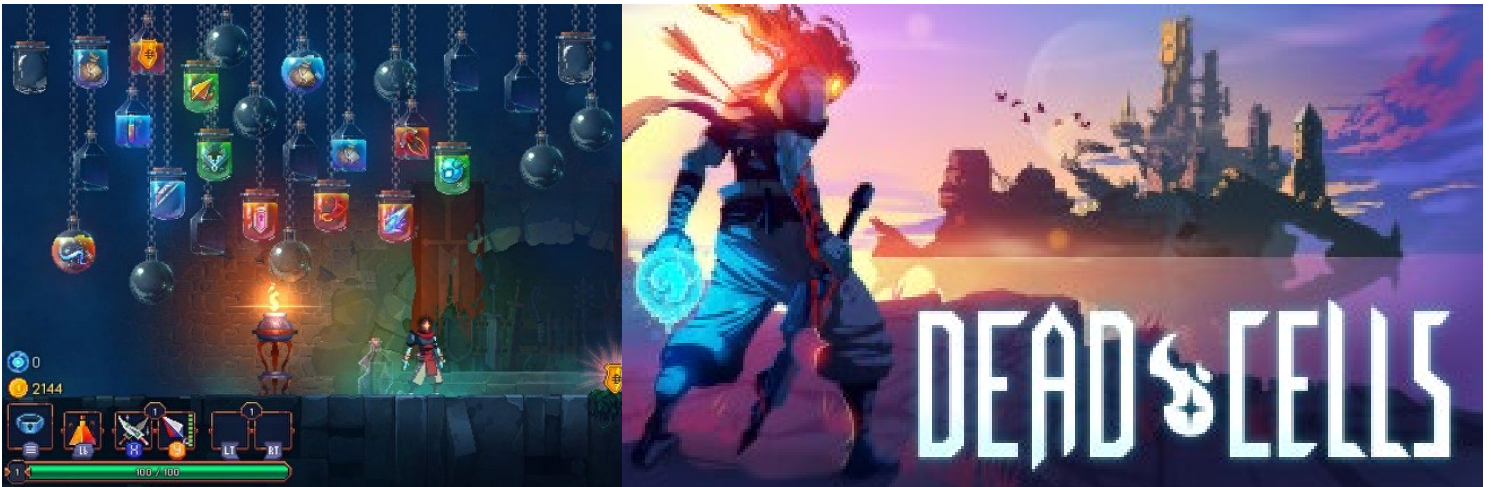
### High Concept

Simply put Dead Cells but in 3D and on Mars.

A 3D Metroidvania style game that rewards verticality and is created using a procedural system tool that allows level designers to input certain variables in an easy to use tool that will then generate a playable level environment, each procedural level will be attached to a seed.

### Core Statement

Tool similar to Dead Cells level creator in function. The tool will allow the level designers to create fresh looking level each time without making them visually and mechanically similar, stunning the player with the results. The tool should be easy to use and explained and understood in little to no

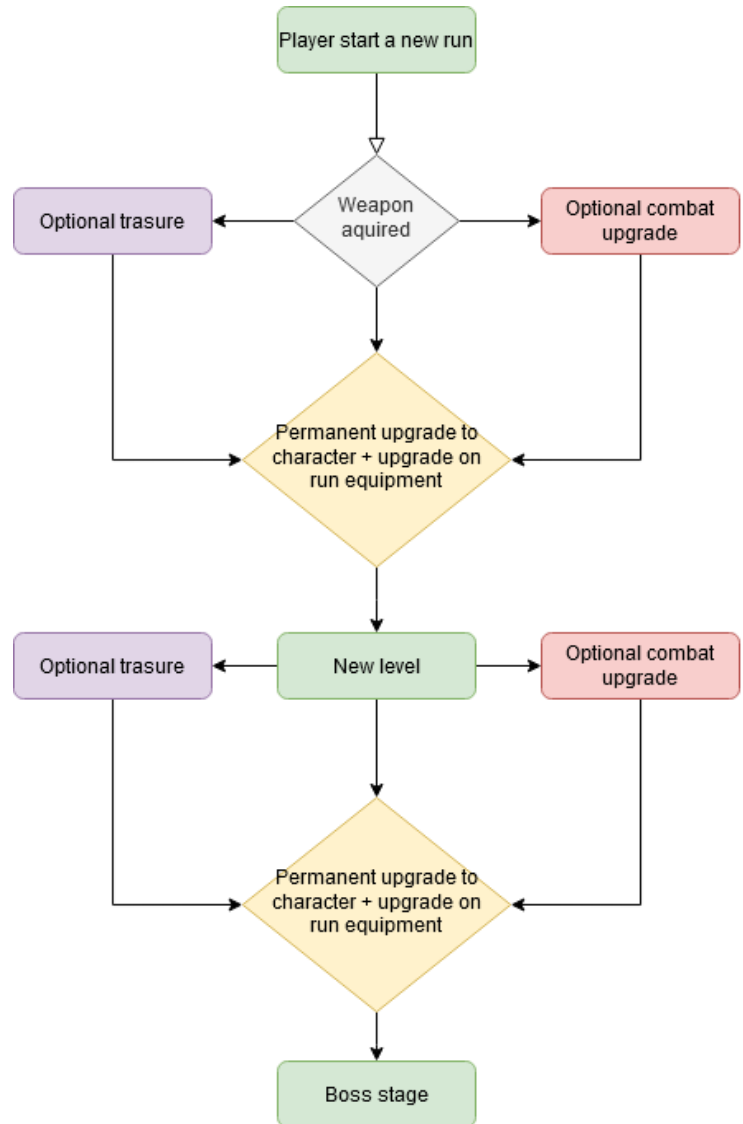


time.

# Feature Set & Descriptions

## Game Progression

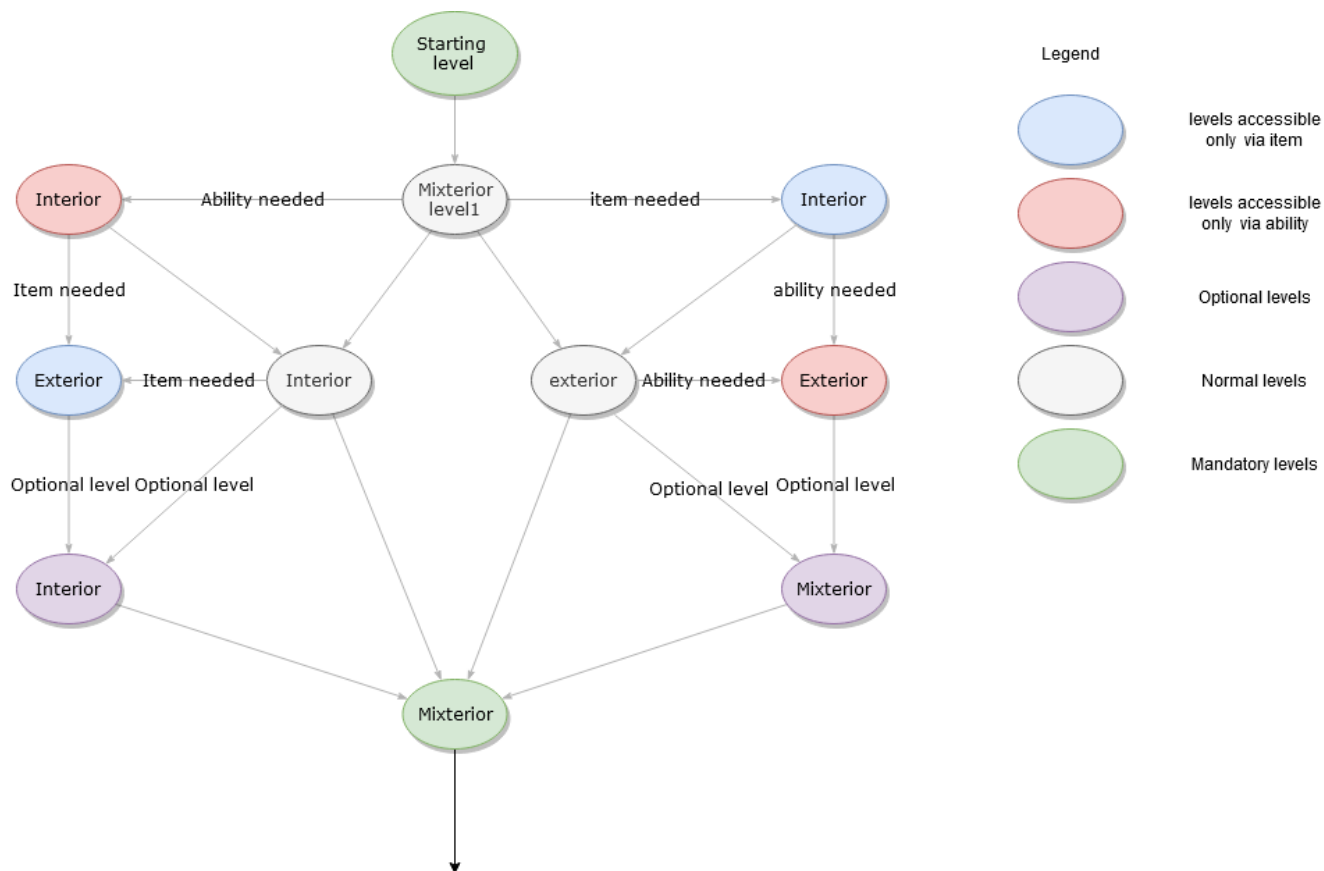
There are two progressions that the player has to undertake. The first one is the progression of the map with the character in the present, with its weapons, upgrades and items. The other is the progression that the player makes upgrading permanently the player character, boosting or acquiring an ability, unlocking a starting item etc... The graph on the right shows the expected game progression.



# Mission Structure

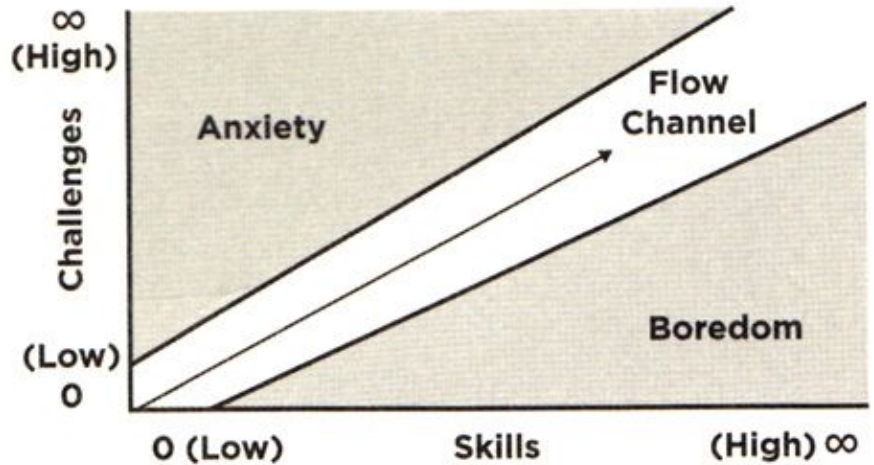
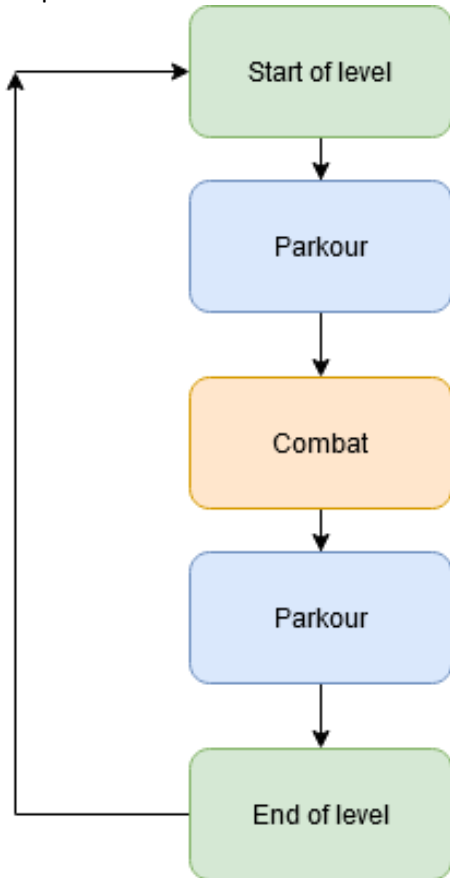
The mission structure is quite similar to the one found in Dead Cell, taking the player through a pre-determined series of biomes with fixed routes. Not all biomes are accessible to the player: there are biomes that require a specific ability to enter, or that require a specific item/upgrade to enter.

A visual representation of the maps in the game can be observed in this graph.



# Play Flow

The desired flow for the game keeps the interest of the player giving it incessant input from the game, challenging it in various ways, from traversal puzzles to combat sections. In order to keep the player in the zone, the player has to be subjected to new input from the game about every five second. This is looked more in depth in the document “Supporting documentation gameplay cells”. The graph on the right illustrate how engaged the player should feel when playing the game, while the graph on the left represent the flow simplified to its barest.



## Game Mechanics

### Movement (basic + abilities)

Movement can be divided in two section, basic movement and movement abilities.

The basic movement is the standard movement set of third person game, such as

- Walking
- Running
- Jumping

Movement abilities are those abilities that influences the way the player traverse the levels, those include:

- Wall running

- Sliding
- Gravity objects
- Gravity sling

### Progression (upgrades)

Each time the player traverses a level, it can find an upgrade for its loadout or find a permanent upgrade for the statistic of the player character. This progression is continuous, letting the player growing stronger and stronger, reinforcing the positive feedback and looping until the player feels the game is mechanical beaten, leaving the cathartic experience to traverse it as fast as possible with all the abilities.

### Portals and fast travel stations

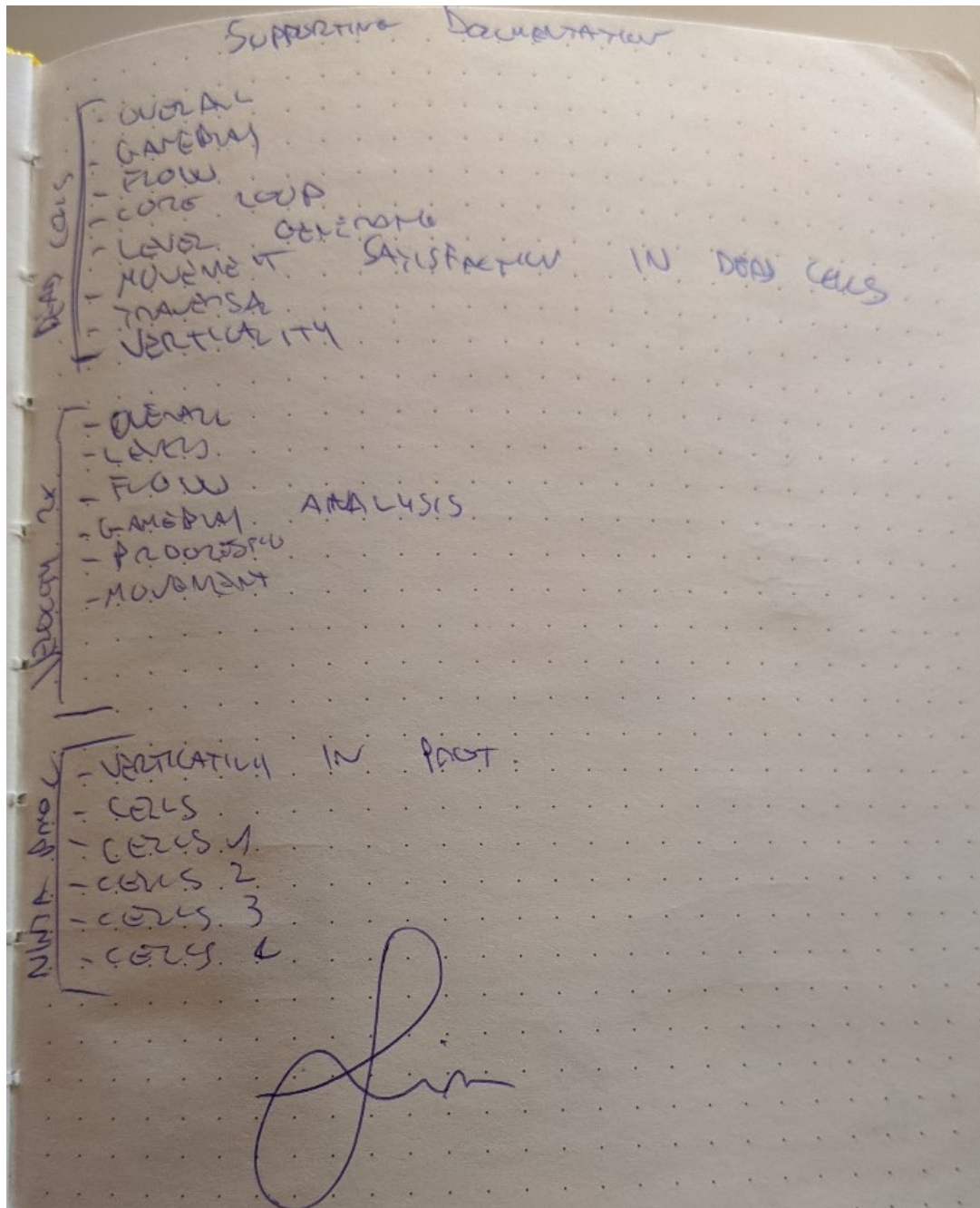
Inside a level, the player can use multiple ways to fast travel. These can be visually different, but mechanically are equal.

### Combat

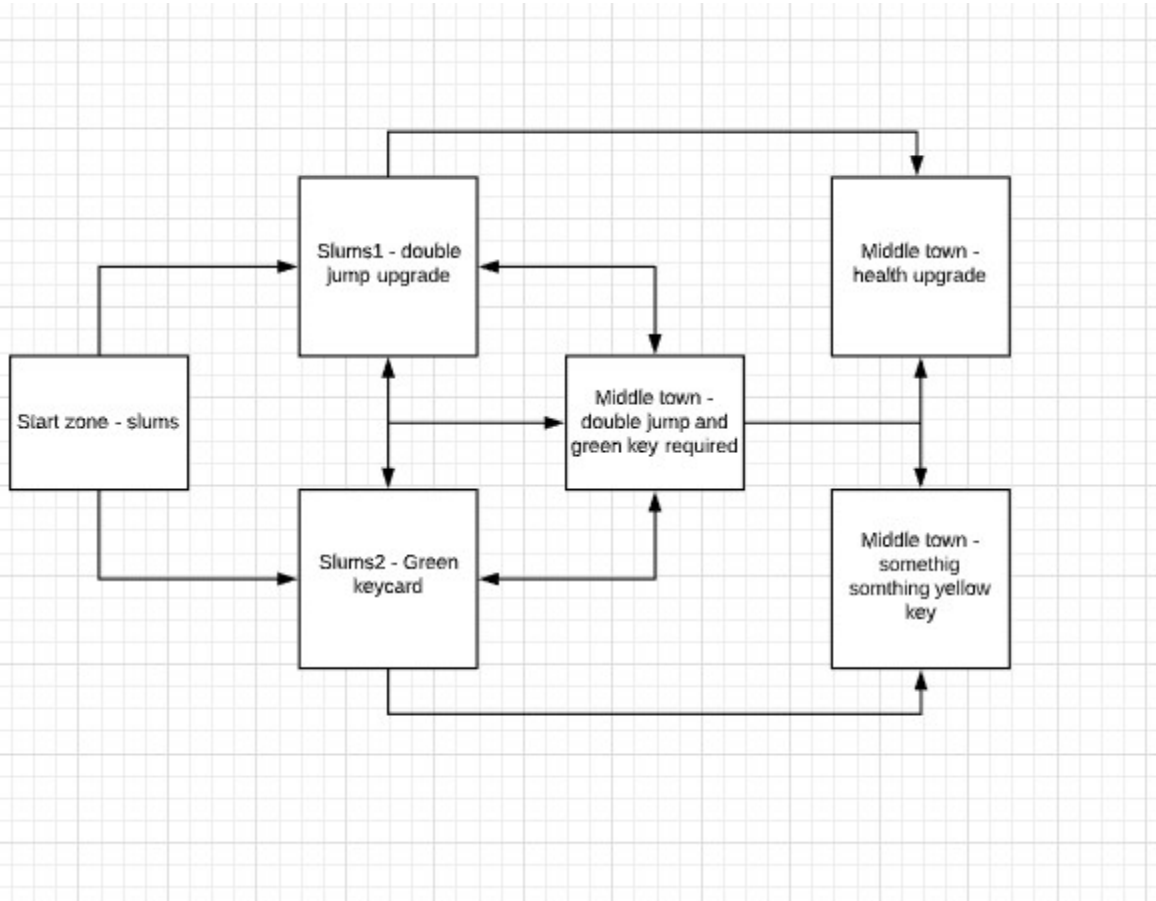
The player character circles around an enemy using the gravity boots looking for vantage points that the player can strike. To perform a combo there is a display on the screen that the player must execute with a list of buttons. The player character darts from enemy to enemy swirling around them as they dodge attacks and tries to angle itself properly to attack.

## Documentation 2

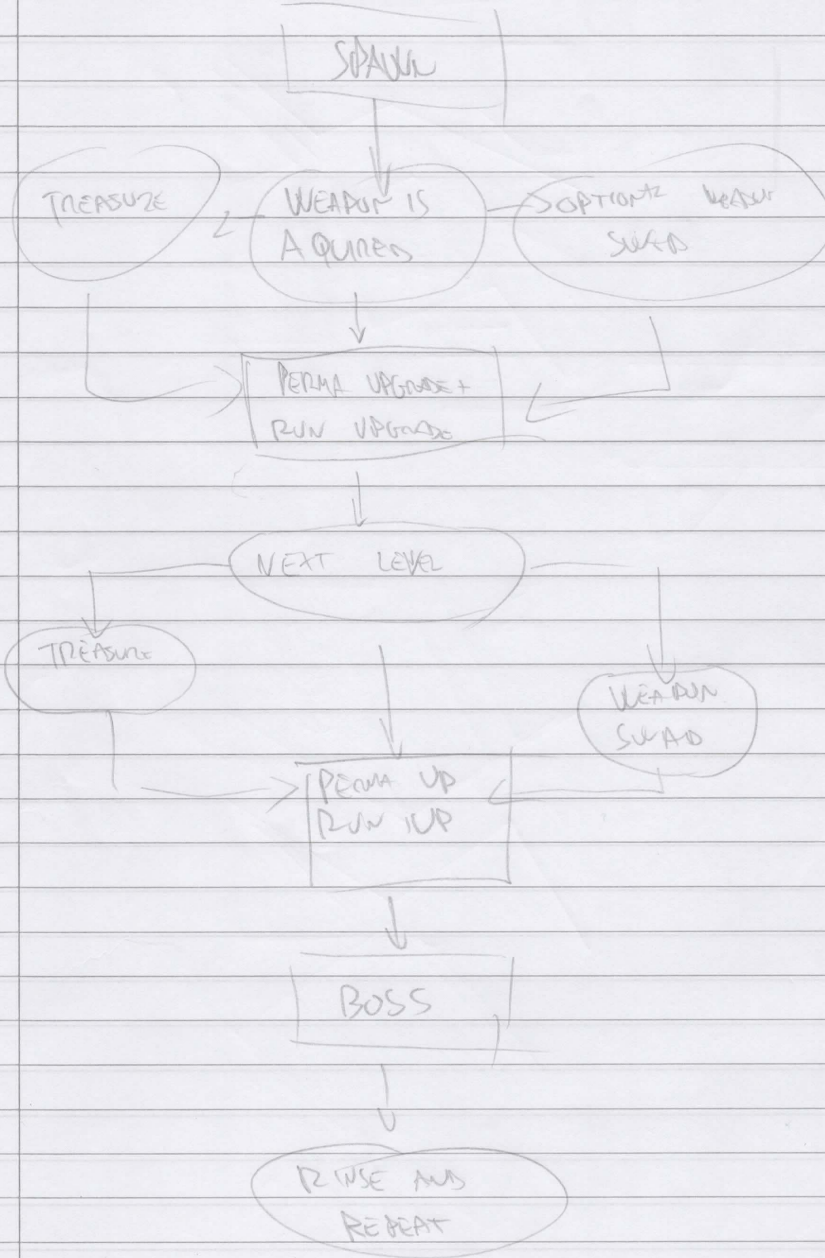
The other documentation I did is the file "Supporting documentation gameplay cells". It's around 42 pages long and around 6000 words. In the document I talk about extensively about Dead Cells and Velocity, analysing the flow, gameplay and pacing. After that, I talk about the gameplay cells for this project and the philosophy behind it. This file is included in the submission. The image below is the outline of the document.



Miscellaneous notes



FACE ABOVE EVERYTHING ELSE



224 Back **Stats** Menu

**TIME PLAYED**

**PLAY TIME** 05:24:26

FLIGHT TIME 02:59:38  
FOOT TIME 02:24:16

BOOST TIME 00:49:29  
RUN TIME 00:27:45

**PROGRESS**

**GAME OVERS** 4

LEVEL WINS 71  
TOP TENS 0

TIME EXPIRED 0  
BOSSSES BEATEN 290

**TELEPORTING**

SFT COUNT 1992  
TELEDASH COUNT 2572

LFT COUNT 110  
PFT COUNT 137

**MAP CHECKS** 118

**PICKUPS**

SURVIVORS 9728  
REKENIUM 10126

HEALTH COLLECTED 2005  
SURVIVORS LOST 360

**PICKUPS COLLECTED** 223

**TOTAL POINTS** 344725

**PROGRESS**

Upgrades	Total number of games	10
Weapons	Total number of games finished	0 (0%)
Skills	Most Gold obtained at once	28474
Mutations	Total gold acquired	245744
<b>General</b>	Total gold spent	173800
Exploration	Most Cells obtained at once	44
Biomes	Cells acquired	939
Monsters	Cells spent	820
Deaths	Teleportations used	61
	Health flasks used	44
	Curses survived	1

RT R Scroll B Back

NAKAGIN TOWER | KISHO KUROKAWA

↓

Humans and plants → modular design

Smaller person don't need much space.  
light spaces for light trees.  
Initial design, more houses and  
the glass, but design failed to take  
in mind that.

MODULAR DESIGN → NARROW SPACES

CITIZEN KANE 50 LVL

MOVEMENT



FLAIR

PIERCE SOLD &  
ALWAYS HAVE  
ABOUT INSPIRE



SAVO

PIERCE INSTRUCTION TO  
VARIOUS CONCEPT



A PIERCE

EACH MISSION HAS A  
TARGET

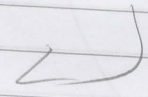


NAVG

GA

SOON MAP GET  
KNOWLEDGE ON  
NEW ELEMENTS

NO RULER  
CRISP



FAST PIERCE COMBAT. NOT UNFAIR.

RATING SYSTEM

CHARACTER UNLOCK  
PAC-MAN CANNON  
RIFLE  
PROJECTILES  
- TP  
TELEPOD

SHIP UNLOCK  
BOMBS  
JUMP/TP

LEVEL THING  
SPEED LINES

- LASER  
- TELEPOD

INITIAL  
CORE LOOP

TRAVERSAL

↓  
BUTTON

↓  
ENEMY

↓  
END LEVEL

MID LEVEL  
CORE LOOP

TRAVERSAL

↓  
TELEPOD

↓  
ENEMY

↓  
BUTTON

↓  
ON FOOT  
SEGMENT

↓  
END LEVEL

---

Overall goal is reach end of level  
Apart from initial stages and special levels, all  
levels are in both space and on foot  
Mechanics are usually the same in space as  
on foot, with slight change as per the level

DEMO	LIVELLO	NAVIC	MECHANIC
Ice		"	TELEPORT +
TEL		"	TELEPORT
4+1		FOOT	TRAP
5		FOOT	TELEPORT
6		FOOT	SPARK
7		FOOT + NAME	SPARK CASED
8		NAME + FOOT	ORDERED CASE SWITCH
9		NAME + DASH	SPECIAL WEAPON
10		NAME + PIEDI	TURRETS

LEVEL GAMEPLAY NEW MECHANIC

CRITICAL URGENCY PACK 6

DUAL CORE PACK 10

DAILY SPIRIT PACK RANDOMLY GENERATED EACH DAY

16 # DICE MAP + RANDOMLY GENERATED EACH DAY

## Development work

Sadly, due to technical constrains and time mismanagement from the code department, my work was not implemented in the final build. The design philosophy and guidelines for the cells are according to my guidelines.

I had done 3 versions of the cells, pouring my time into each one of them, trying to design them at the best of my capacity, but due to the above mentioned continuous technical problems with the code team, I kept doing different cells until the time was running out and the art team did cells without giving me anything to do or way to help them.

## References

80.lv. 2020. *80 LEVEL*. [online] Available at: <<https://80.lv/>> [Accessed 14 May 2020].

2020. *Dead Cells*. Twinmotion.

2020. *Velocity 2X*. FuturLab.

Gamasutra.com. 2020. *Gamasutra - The Art & Business Of Making Games*. [online] Available at: <<https://gamasutra.com/>> [Accessed 14 May 2020].

GamesIndustry.biz. n.d. *Home | Gamesindustry.Biz*. [online] Available at: <<https://www.gamesindustry.biz/>> [Accessed 14 May 2020].

Gdcvault.com. 2018. [online] Available at: <<https://gdcvault.com/>> [Accessed 14 May 2020].

KREMERS, R., 2018. *LEVEL DESIGN*. [Place of publication not identified]: CRC Press.