



SUPPORTING DOCUMENTATION FOR LEVEL DESIGN



FILIPPO CASOLA 1702870
ABERTAY UNIVERSITY A/Y 2019-2020

Table of content

Glossary	2
Why the need for this document?	3
Analysing Dead Cells	4
Gameplay and progression in dead cell	5
Flow analysis	6
Core loop in dead cells	7
Level generations in Dead Cells	8
Movement satisfaction in dead cells and analysis	9
Traversal of Dead Cell	11
Verticality in Dead Cell	13
Velocity 2X	15
Levels in the game	16
Flow analysis	19
Progression in Velocity 2X	20
Movement and navigation	21
Movement in space	21
Movement on foot	21
Verticality in [REDACTED]	22
Cells in [REDACTED]	23
Cells v1	24
Cells V2	27
Cells V3	31
Cells V4	43

Glossary

Before starting any conceptualizing and designing, we must state and define each term that will be used in this document. There must be cohesion in terminology, to avoid misunderstanding between the author and whoever reads the document and with other team members, while also ensuring consistency throughout the documentation.

This list is a work in progress, due to continuous changes in direction and the union between the design and the algorithm. If during the development the term for something has changed, it will be listed as (old/name) shown in the example down below

Example

- (Old/Weapon) **Item**= object that boosts the damage output of the player character. It can be found inside **levels** or bought inside specialized **shops**

Glossary:

- **Map**= The sum of all the levels. For example, the map is located on Mars, while single levels can be located on the surface, inside lava tunnels or inside a colony. Synonym "**world**"
- **Level**= Space where the player has to get from point A to point B. Once point B has been reached, the next level is loaded and played through. Also called "**stage**"
- **Layout**=Disposition of assets, or in other cases, the geological layout of the map.
- **Chunk**=Chunks are small pre-approved gameplay sections with a standard dimension. Also called **cells**
- **Biome**= the defined environment and visual theme of the level. An example would be **mixterior**, **external** or **interior**
- **Shop**=place where the player can acquire upgrades. They can be synth upgrades or more mechanical upgrades.
- **External**= refers to the levels that are located on the outer surface of Mars.
- **Interior**= refers to the levels that take place in a colony city/HUB.
- **Mixterior**= refers to the levels that take place in the dried lava tunnels.

Why the need for this document?

The client asked for a tool that would create randomly the levels for the game. This tool should give the player a sense of wonder every time the player sees a level, being different every time, yet coherent and fun to navigate and play through it.

This goal is near impossible to achieve without designing an algorithm able to create something that could be considered comprehensible and that is satisfactory enough to the client.

The goal of this document is to narrow down the infinite possibilities and decide from where and how to get where the client wants. For example: analysing dead cells we can see how the chunks bit helped cutting down the development time without compromising the quality of levels and biomes.

Analysing Dead Cells



Over the last few weeks (from week 4), [REDACTED] stressed multiple times on different occasions on how they wanted the same gameplay flow and “progression” present in Dead Cell. In a simpler way one could say that they wanted Dead Cells but in three-dimension space (and set on Mars). To gain more knowledge and a deeper understating of what the client wanted, I first had to research what makes the Dead Cells flow so greatly and how it can become so highly addictive. I also had to figure out how to translate it in three-dimension space, without losing the same sense of movement and combat.

Before diving into Dead Cells and each component; I would like to establish, what Dead Cells is, the key concepts that define it as a game, what components of it make it stand out compared to the other games of the same genre. This introduction is going to help comprehend the analysis and flows paragraph if reader is unfamiliar with Dead Cells.

Dead Cells is a 2D action-combat centric game developed by the French studio Twinmotion. On the game’s website, they use four keywords to describe it:

- **Rogue Vania:** “intense 2D action with the adrenaline pumping thread of permadeath in a castle full of cuddly creatures.”
- **Souls-like combat:** “Pattern-based bosses and minions, weapons and spells with unique gameplay. Roll roll roll your boat gently down the stream”
- **Non-Linear progression:** Unlock new levels with every death, take a new path. Tired of the stinking sewers? Why not take the ramparts?
- **Exploration:** Secret rooms, hidden passages, charming landscapes. A fine place for a holiday.

Roguevania is a term coined by Twinmotion to define their game. They used it because the core loop of Dead Cells is the same loop as found in others Rogue-like games, mixed with elements from the MetroVania genre.

Souls like combat means that both the enemies and bosses are pattern bases. So the player can learn their pattern and how to counter them. Enabling the player to optimize the combat by learning when to roll away when the combat gets too hard or unmanageable.

The non-linear progression of Dead Cells is interesting. The player has two “games” in one. The first is with the current character with its stats and objects, while the other is the future game that will be influenced by the upgrades obtained by the player in the “current” run. This helps to make the player feel motivated to continue to play the game. It makes the player feel as if it’s getting stronger. Considering also the fact that the player is learning the behaviour and patterns of the enemies, it can create a highly addictive core loop. The progression in the map is at the player’s discretion and tied to the player’s experience and items.

Exploration The player is encouraged to explore every single room to avoid losing any secrets or much needed upgrades or new weapons. Or even better, cells used to increase permanently the stats.

Gameplay and progression in dead Cells

Gameplay in Dead Cells is extremely fast paced. The player character can only take about 4 to 5 hits at best before dying and resetting his current run progress.

Platforming is a huge component and it is refined to perfection. The movement will be analysed more in depth in its section.

As we are going to discuss in the flow analysis, and later in the movement satisfaction section, the jump is a central component in the game, establishing flow, serving both traversal and gameplay mechanics. For example; it could be used for simple traversal, jump to get to a higher platform, or used for gameplay, dodge an attack, gain momentum for a leap attack, or just jump to aggregate more enemies.

Dive is used to inflict damage while also having the opportunity to evade incoming damage or avoid sticky situation. If the player dives and attacks while facing away from the enemy, the game automatically inverts the direction of the player character to connect the outgoing damage.

No commitment while giving out damage > lowers skill but also gives the opportunity to the player to avoid more consistently the incoming damage.

The combat is standard, with invincibility frames when rolling. The combat is shown to the player as a puzzle that he has to solve with the weapons and build that he has, also taking into account the level and the surroundings. When initiating an attack, there is no commitment to it, so the player can stop it anytime. This mechanic lowers the skill ceiling but helps with the movement satisfaction.

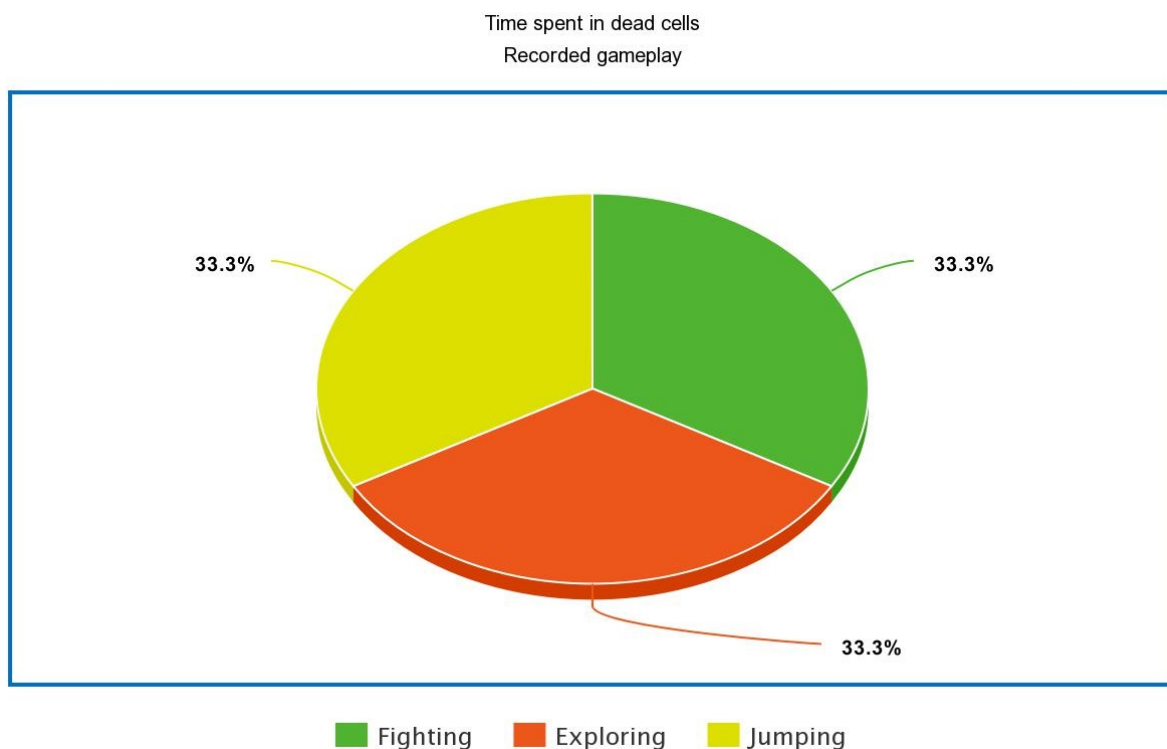
Overall, the mechanics in dead cells are fine tuned to make the player earn every little accomplishment, challenging but never until it become frustrating. It combines the outstanding movement with a deep combat system, crating one of the most compelling gameplay loop of the genre.

Flow analysis

Firstly, I played through Dead Cells until I familiarized enough to comprehend the feeling and the flow of the game. Once I achieved that, I recorded my screen while starting a new run and after a little more than 20 minutes, I examined the gameplay to comprehend and analyse the flow. Once one gets into the spectator seat, the game feels smooth and fast, but chaotic enough to surprise you even if you played that game yourself.

After carefully analysing twenty minutes of my recorded gameplay, the conclusions were quite perplexing at first. The main chunks of action that the player is actively doing are pretty much three, without counting the time spent in shops and in the areas between levels. A third of the time was spent exploring (simply running around between doors and enemy encounters), another third of the time was spent fighting with enemies and avoiding/triggering traps, while the last third of the time was spent jumping around the map and enemies. In the end, every 3 to 5 second, the player is expected to be stimulated by enemies, platforming or traps.

The graph below illustrates these percentages.

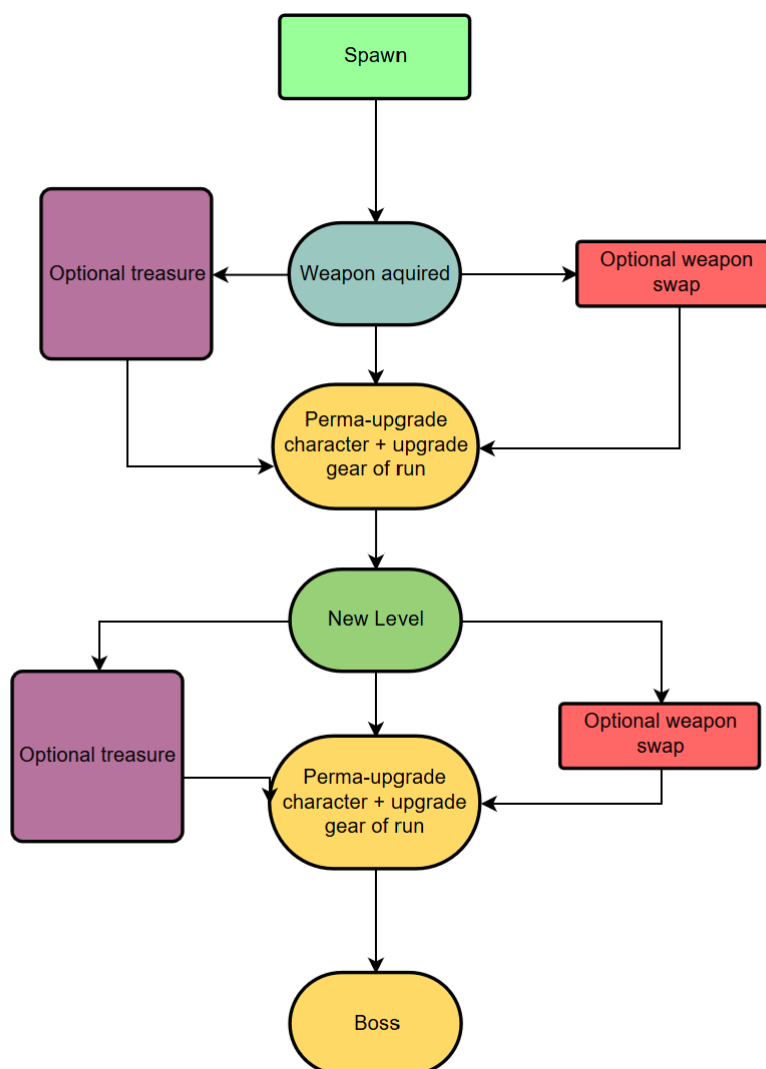


Core loop in Dead Cells

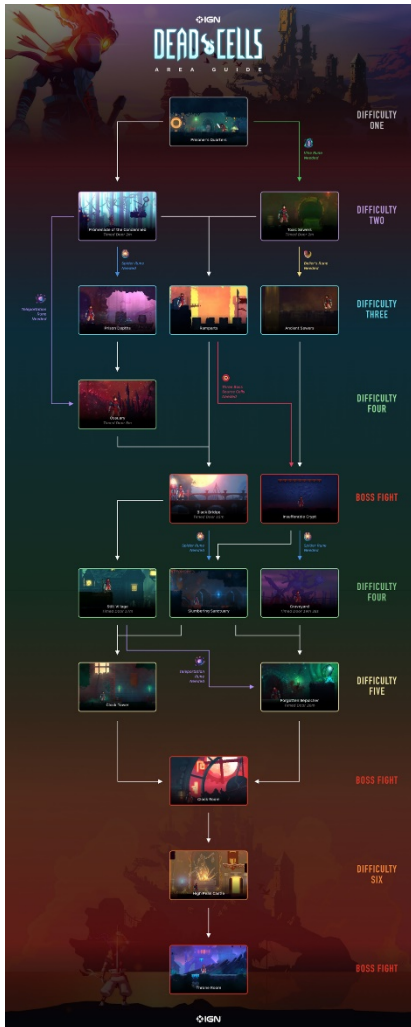
After a couple of runs, the core loop that emerged from the game is quite simple yet ingenious.

The player can see that each time he plays, the game rewards him with some help using either weapons or treasure. This programmed “luck” helps to make the player feel that he has beaten the odds. Upgrading proves to the player that he is getting better at the game, and his gears are increasing in damage or stats as well; while also, his permanent stats are increasing a little each time he reaches a safe zone. This reinforces the feeling of satisfaction and fulfilment in the player, giving him another reason to get back in the game and keep playing even after a successful run.

Below is a flowchart representing the core loop at its bare minimum

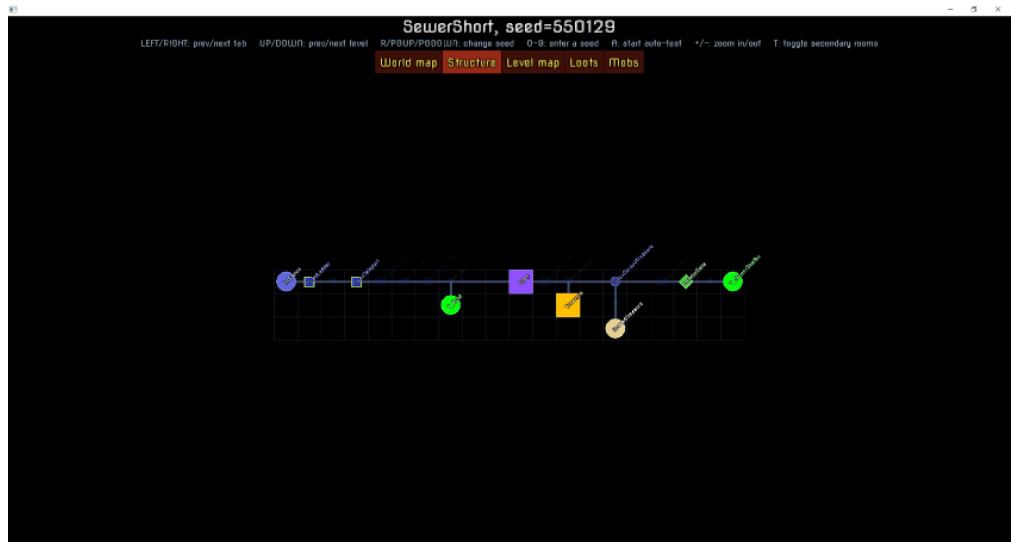


Level generations in Dead Cells



The image on the left is the complete map of Dead Cells. It shows all the biomes, and how the player can progress from one into another. These biomes are pre-defined, while the levels in this map are procedurally generated. In the next paragraph we are going to see how they generate a single level in Dead Cells.

A level (or biome as they are called by TwinMotion) is generated using a line to set the rooms, the structure and the order they are placed is the same as shown in the image below.



Once the line is generated with the right parameters, the algorithm takes pre-approved chunks (shown in the image down below) and places them together, keeping in mind the other rooms specified above. These chunks are designed with modularity in mind, and most of them are not specific just to one single biome but can be reused for multiple biomes. This ensures that players never encounter the same level, and therefore prevents the development resources from getting wasted on a high volume of carefully crafted levels that would be predictable after a couple of runs.



Once the algorithm places the chunks in place and ascertains that the player can reach the end of the level without any Rune (to ensure that no matter the level of experience of the player, the player can traverse the level), the level is ready and the player can navigate through it, without any problems, as shown below.



Movement satisfaction in dead cells and analysis

One of the main reasons Dead Cells is so addicting and highly replayable is the fact that moving inside the game feels “good” and effortless, as if the player’s inputs were perfectly translated from the player to the game. This feeling is achieved by an impressive input interpretation and by a set of tricks that the player doesn’t notice but help with the ease of navigation. Developer Twinmotion said that the amount of tricks they used in Dead Cells are around 50, and they are used not only for the movement but for every aspect of the game. These are the ones that I found regarding the movement satisfaction while playing:

1. **Just in time jump** = The game helps the player getting around by not requiring split second precision jumps, but there’s a fairly generous big leeway to manoeuvre after having missed the perfect beat or jump. This is roughly a quarter second. Motion Twins calls this leeway “just in time jump”.
2. **Just in time climb** = When the player has just executed a jump and is approaching a ledge, the game helps the player climb it even if it’s too far for the player character.
3. **Auto Climb** = when the player is near a small step in elevation and moving towards it, the game automatically jumps over it, letting the player focus on other things.
4. **Auto Jump** = just like the Auto Climb function, in certain situations, the player character jumps without the input of the player.
5. **Smashing door** = if you roll into the door, they break and stun the enemy behind the door. It is also used to gain momentum.

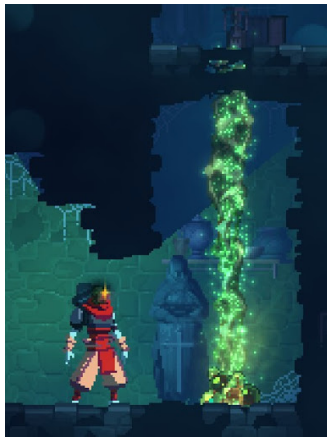
6. **Dive Direction** = the player can dive from an elevated platform to a lower one. When doing so, if an enemy is on the lower one, the player character will automatically land facing it. When it is more than one enemy, the game will choose the enemy that is the nearest.

Traversal of Dead Cells

The movement in Dead Cells is extremely fast, with the standard set of ability, such as jump, roll, plunge, etc. But to further the complexity and higher the skill ceiling of the movement and traversal of the map, Dead Cells uses Runes to give the player new ways to traverse the map. These Runes are found scattered around the map and are permanent upgrades tied to the single save file. Each Rune is guarded by a boss (usually it is a mob specific to the biome where the Rune is founded, but with more health and a higher damage output).

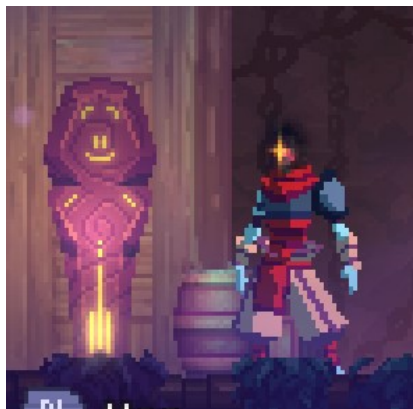
Runes that affect the movement:

- Vine Rune. This Rune gives to the player the ability to grow vines from green sprout. This ability helps the player explore the map when the player is beginning his run.



The vine extended after using the Vine Rune

- Teleportation Run. This Rune gives the player the ability to teleport via electric sarcophagi. This helps the player explore closed off sections of the map that are accessible only through this method of transportation.



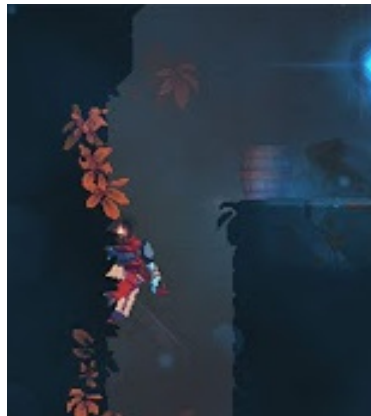
The electric sarcophagus that allows the player to teleport once in possession of the Teleportation Rune

- Ram Rune. This gives the player the ability to slam into cracked ground and break it. This helps the player navigate the parts of map blocked by the above-mentioned obstacle.



Player slamming into the ground thanks to the Ram Rune

- Spider Rune. This Rune gives the player the ability to climb wall and wall-jumping. This rune increases dramatically the amount of freedom the player has in exploring the map, more than any other Rune. To balance the progression of the game, this rune is found in a specific level which is mechanically difficult, so it has to be gained after a lot of hours of experience and training.



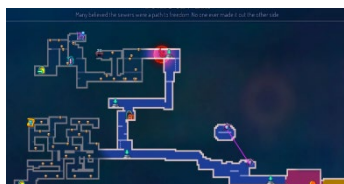
Wall climbing in action thanks to the Spider Rune

- Homunculus Rune. This rune allows the player to detach the head of the player character from the body. This allows the player to gain access to certain areas that were not accessible before.



Head detach in action thanks to the Homunculus Rune

- Explorer's Rune: This rune reveals the entire biome and goodies in the level. This activates once the player explores the 80% of a level.



The effect of the Explorer's Rune on the player's Map

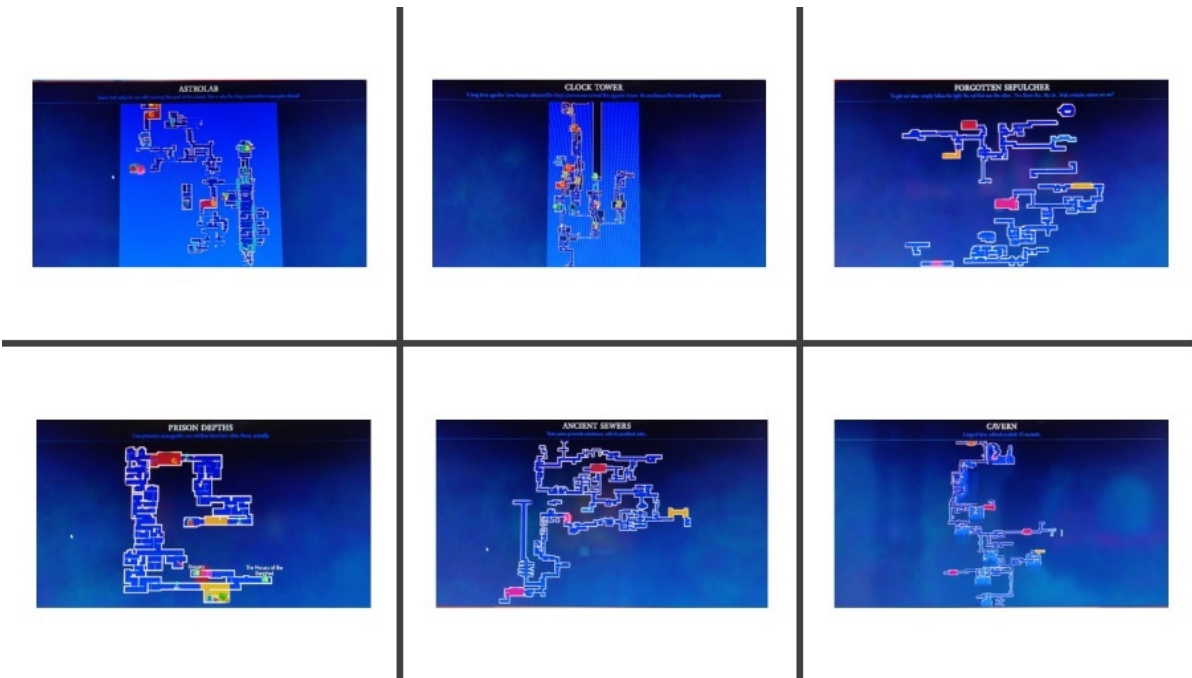
In conclusion, thanks to these runes and the basic movement of the player character, the traversal of the game is extremely pleasant and almost cathartic (if it wasn't for the enemies).

Verticality in Dead Cells

As we have seen in the previous paragraph, each level is generated procedurally in mere moments before the player gets the chance to traverse it. Due to this, it is impossible to get a map of how vertical levels are supposed to be, since chunks can differ in height and the way it works negates any way to know for sure. For this document, I played a couple of runs of Dead Cells and photographed the “most average” level of each biome. Below is the image showing them.



Each biome has a gimmick. This gimmick can range from a safe space mechanics (the player has to reach light range in 15 second after leaving it, otherwise he dies) to toxic pools, etc... There are only 6 six levels that use verticality in a substantial way (as shown in the picture below).



Verticality is present in almost every level in Dead Cells, exception made for boss stages. Even if it is present in virtually every level, it is not the core mechanic nor a relevant component, but a mere result of the chunks method of assembling the levels. Only in a few handfuls of levels the verticality is the real focus of the biomes. One example would be the astral tower, where the player has to navigate a maze of elevators and towers. In this case, verticality is quite important, but it is always in the background, leaving the pacing dictated by the combat.

In conclusion, we can say that verticality in Dead Cells is present but it is not a core element of the experience. It is used to deepen the combat and to give some levels a different theme.

Velocity 2X



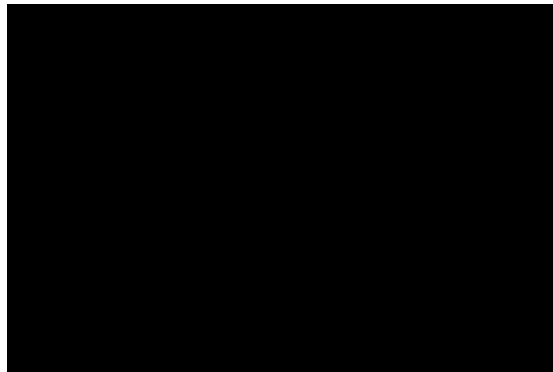
The same day the team had the client pitch, the client was stressing once again on the importance of Dead Cells as an inspiration, [REDACTED]

[REDACTED] After watching some videos on YouTube, I decided to play the game to get a better understanding on the requirement of the client. It must be noted that in doing so the client encouraged us to watch at Speed Runners while also looking at Velocity videos during the post presentation meeting, implying that one of the main elements to take away from the game was the swiftness and speed of the game.

Velocity 2x is both a shoot 'em up with light puzzle elements and a side scrolling platforming. These two genres coexist in the same game thanks to a change of perspective, piloting the spaceship in space with the shoot em up segments, while the platforming is with an on foot sideview.

The main goal for the player is to reach the end of the level, while also completing one out of four fixed objectives. This fixed objectives can be gathering of crystal, refugees, the destruction of enemies, defeat the level boss or reaching the end of the stage as fast as possible. All of these objectives are usually announced before the level starts so the player can prepare mentally for the task that lies in front of him.

Here's an example of gameplay from Velocity. (if video doesn't open link is <https://youtu.be/n5etCXadk0>)



One of the main characteristics that stands out in this game is the dual gameplay section, that almost in every map intertwinds one with the other to create the fast-paced traversal and combat.

One, sees the player piloting a spaceship in a shoot 'em up gameplay, with emphasis on speed and reflexes. Combat and avoiding obstacles are the main mechanics.

The other is on foot, with a 2d scroller gameplay, with emphasis on movement, shooting and collecting crystal

Levels in the game

There are four different ways to play levels:

1. **Fifty levels in the base game.**
2. **Ten new levels from the Dual Core pack.**
3. **Six new levels from the Critical urgency pack.**
4. **Theoretically infinite levels from the Daily sprint pack.**

1) **The fifty levels in the base game:** are built upon a simple core loop, to introduce the player to a single mechanic, let the player familiarize with it, mix it with other mechanics, test the knowledge and skill of the player, then extend the depth of the mechanic a little further. Rinse and repeat.

2) **Dual core pack:** As for every downloadable content, this pack has a specific focus and theme: let the player control two spaceships instead of one. The controls remain the same, but every input will influence both the ships. The player will navigate through these ten levels solving medium/hard levels.

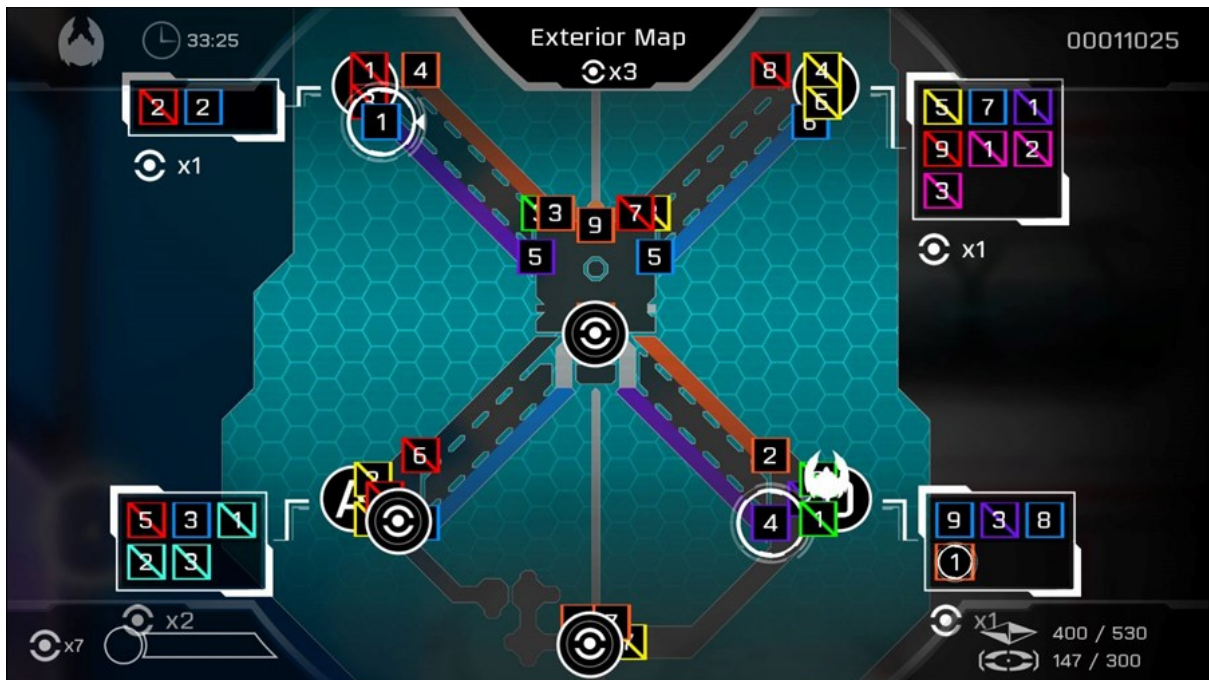
3) **Critical Urgency Pack:** the theme for this DLC is the speed of traversal of the level, or simply said, the velocity of completion of the levels. In these six levels the player will fight against time to reach the end of the level. The difficulty is pretty high, as time given by the game is just enough to commit a single error in each level.

4) **Daily sprint pack:** This pack has a theoretical infinite number of levels within, each day the level is different, being randomly generated each day. In these levels the player will try to speed run the level while also acquiring all the crystals to improve its score.

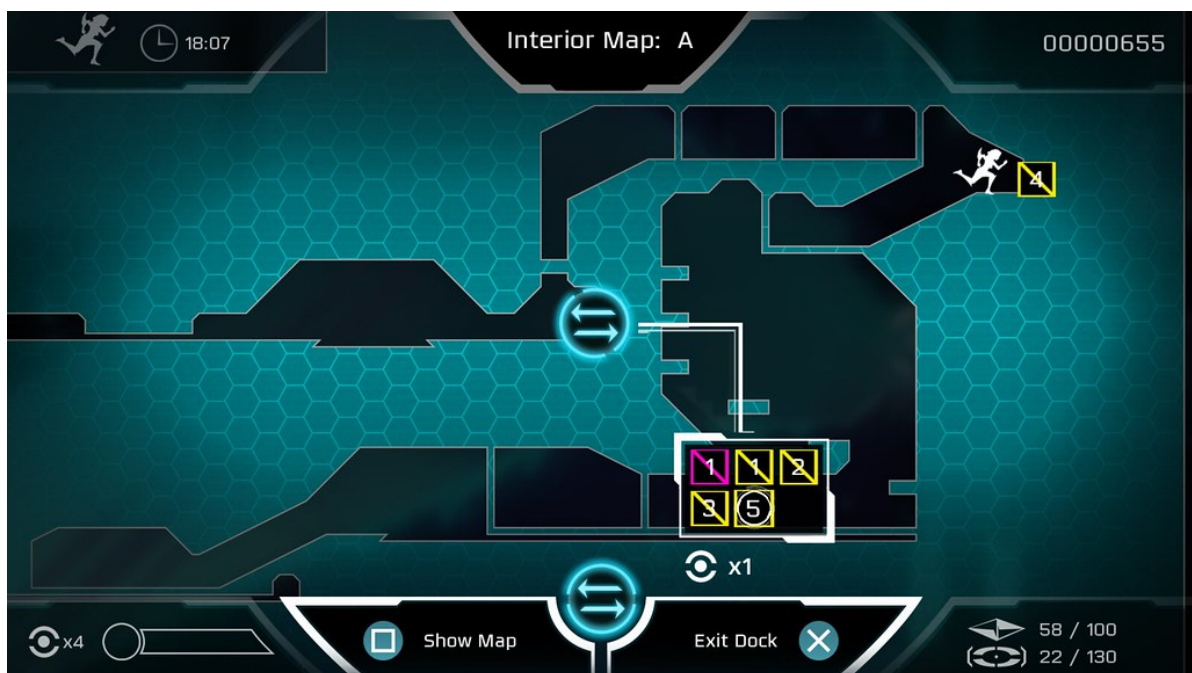
Side note: The aforementioned DLCs are included in the PC edition of the game, the same thing is for the newly released edition for the Nintendo Switch.

Supporting documentation regarding the gameplay cells

As we previously discussed, the maps in Velocity are crafted by hand, so their design is always tight and superb. Due to the gameplay in the space segment, most of them expand in length rather than in width.



Interior maps usually serve the purpose of containing a switch that the player must deactivate to advance with the spaceship. There are usually around 2/3 levels in elevation, with the design winding back to the exit. There could be turrets or Vokh guard, so it can serve as traversal, combat arena or both. Usually there are also crystals to collect inside, but to get them, you need to shoot them to get them to separate from the vein.



One of the unique characteristics of the game is the fact that every mechanics are explored to its full extent trying to expand it in every possible way, but implementing them gradually, giving the player time to adjust. For example, in the first level in the game, the player is taught how to move inside the level, how to steer and move the ships, the fact that it can die when the spaceship gets out of the camera. But it is taught in a safe and slow environment, so the player can learn the 'how to'.

In the next level the player is introduced to the spaceship teleporter. Once again this is brought in a safe and easy environment, to maximise the player learning of the mechanics. The third level is a test for the players, where they have to test how well they understood the mechanics and how good they are at executing them.

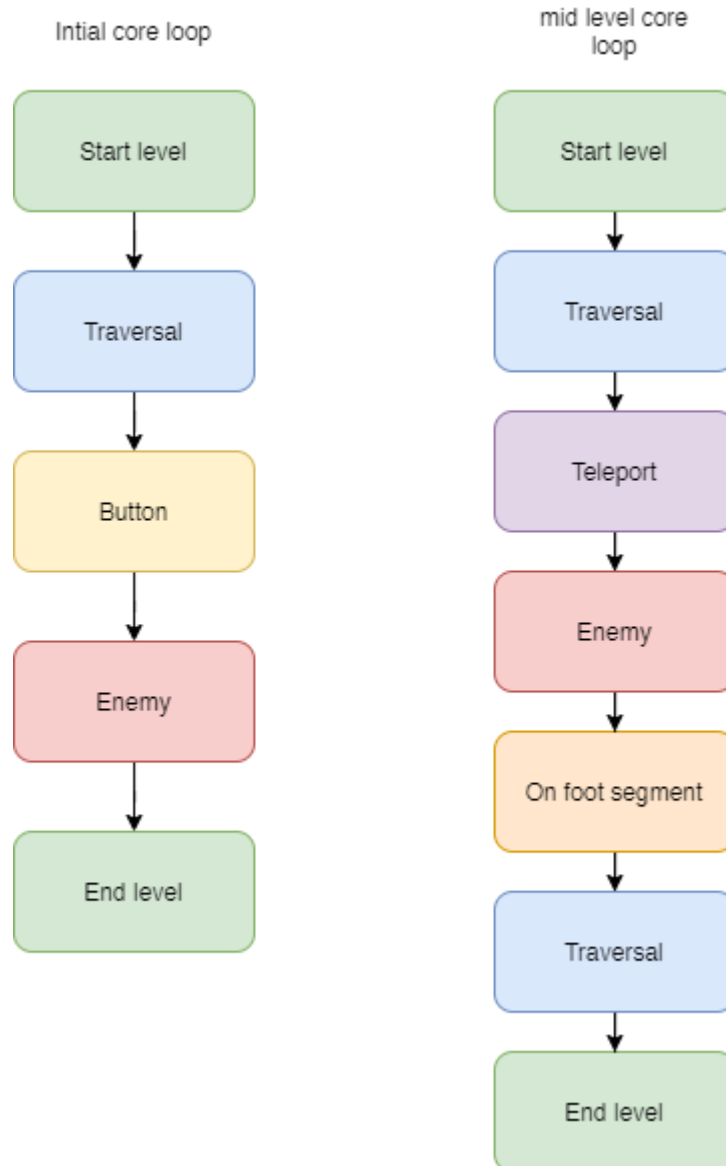
This table analyses the first ten levels of the base game

Level:	Gameplay segment:	New mechanics
1	Spaceship	Movement with spaceship and traversal of map
2	Spaceship	Spaceship teleport
3	Spaceship	No new mechanics- Instead, this level is a test of player's skill and comprehension of mechanics
4	Foot	Movement on foot and traversal of map
5	Foot	Teledash on foot
6	Foot	Shooting crystal
7	Foot+Spaceship	Laser field
8	Spaceship+Foot	Ordered laser switch
9	Spaceship+Foot	Special Weapon
10	Spaceship+Foot	Turrets

The gameplay is carefully crafted, just like the whole experience. There are no different playstyles to adopt, the game is made with one playstyle in mind, and the whole game is beautifully crafted to support that vision. The experience of Velocity 2X is virtually the same for each player.

Flow analysis

The graph below represents the barest flow loop for an initial level and a middle level.



Progression in Velocity 2X

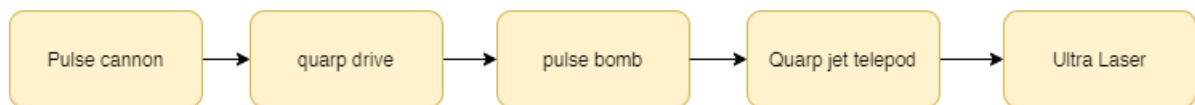
As we mentioned in the above paragraph, progression in gameplay term is constant, as if the whole game was the tutorial for the game, and once the game is beaten, the player can replay with his knowledge and so, beat his previous records or optimize his previous time. The power creep in the mechanic is never felt significantly.

The gameplay keeps changing slightly thanks to the numerous power ups and numerous ways old mechanics are brought back with newer functions or are combined in new situations.

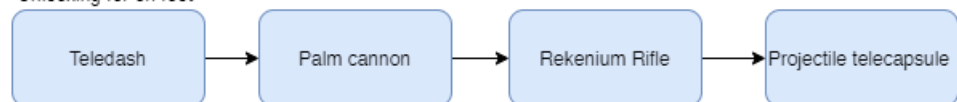
The progression of unlocking new abilities (which can refer to usable objects) is on quite similar from both sections, receiving similar upgrades almost at the same time. It creates a constant feeling of increasing power, letting the player enter a virtuous circle.

The graph below illustrate the upgrade path for the space and on foot segment.

Unlocking for spaceship



Unlocking for on foot



Movement and navigation

Movement in space

The Quarp ship it's easily controllable, fast and reactive to the player input. The feel of the spaceship is instrumental to the gameplay, since the majority of gameplay time is spent piloting it. Due to the high degree of manoeuvrability, the movement and navigation of the levels are perfect for the high intensity and speed of the game. The main mechanics/abilities that control or influence the movement in space are these:

- Pulse cannon. Projectile that is shot only forward. Used to defeat enemy turrets and ships. It can also be used to shoot the buttons.
- Quarp Drive. It's an aimed teleport that the player can perform instantly with no cooldown. Used to traverse the map and essential to traverse each level.
- Pulse bomb. Quad-directional bomb that can be used to defeat enemy turrets, ships, and to deactivate buttons.
- Quarp jet telepod. It's a capsule that the player drops at a bifurcation, used to quickly teleport back to it, usually after having solved the puzzle on one side which was necessary to advance once in the other side of that same bifurcation.
- Ultra-laser. Upgrade to the Pulse Cannon.

Movement on foot

Controlling Lt. Kai Tana is fun, the controls are very responsive and the fact that she can shoot while performing other actions just feels right. Controls are tight, and combo with the objective to traverse more advanced levels are quite satisfying to accomplish, without being too hard to pull off. The capacity of movement of the character is great, thanks to the various way it can fly through the most hostile environment. These elements are:

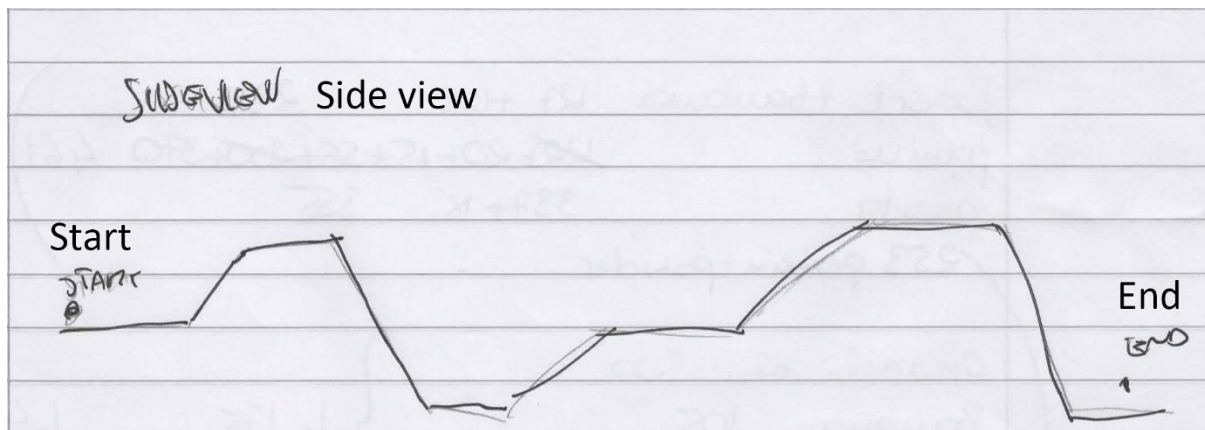
- Teledash. The character can dash through wall or any obstacle.
- Jump. Simple jump used for traversal.
- Sprint. Sprint used to gain momentum and perform longer jumps.
- Palm cannon. Used to shoot wall turrets, crystals and deactivate buttons. Can be directed at any angle.
- Rekenium rifle. Weapon that can only fire directly in front of the player character. Mainly used to kill the Vokh guards after teledashing through them to deactivate their shield.
- Projectile telecapsule. Once the capsule is thrown, the player can teleport instantly to its location. Used to traverse narrow spaces or places that are too high to reach.

Parallels can be drawn from one section to the other, in fact almost all the abilities from the space segment section can be directly compared to the on-foot section; for example, the quad directional bomb has gameplay similitudes to the Rekenium rifle. This gameplay fact simile helps the player with the ease of learning, letting him transfer his knowledge and skill from one section to the others, making the player feel accomplished and powerful.

Verticality in [REDACTED]

As mentioned before, one of the main concerns of the client is the implementation of verticality. [REDACTED], he told me that the median verticality that he expects in a level is about 3-4 floors, while levels that will concentrate mainly/only on verticality will be around 9-10.

Below is a fast sketch of verticality expected by the client as we discussed it and how he described it to me.



From the team came out an idea about a gravity pillar letting the player levitate to higher or lower level. This gravity pillar will be the main mode that the player will use to switch floors, while also use stairs and platforms when needed.

While thinking about the design for the cells, we must keep in mind that verticality was extremely important to the project and to the client. It was essential to the client due to the fact that the gameplay was supposed to be fast paced and the unique selling point (as mentioned in the brief itself) is the traversal at high speed; which has to have verticality to allow for interesting gameplay feature and varied navigation of levels, otherwise the feeling of “certified badass” is not going to be present. It is not fun to run at high speed in a labyrinth of corridors, but it is fun to casually stroll from rooftop to rooftop, taking detour into smaller hallways.

Mentally, what I had in mind while thinking about the design of the cells, was that the gameplay would be similar to mash up between the movement and traversal of Mirror’s Edge and Vanquish with the RogueVania gameplay of Dead Cells.

Cells in [REDACTED]

One of the main factors that we must keep in mind is that the length and extension of the map/level is unknown/ may varies during the development time, so the biggest question to answer is how big and how extensive these cells/chunks were going to be. The dimension of these chunks is also important because they establish the scale of the level, so deciding blindly without giving any thoughts to it was inconsiderate.

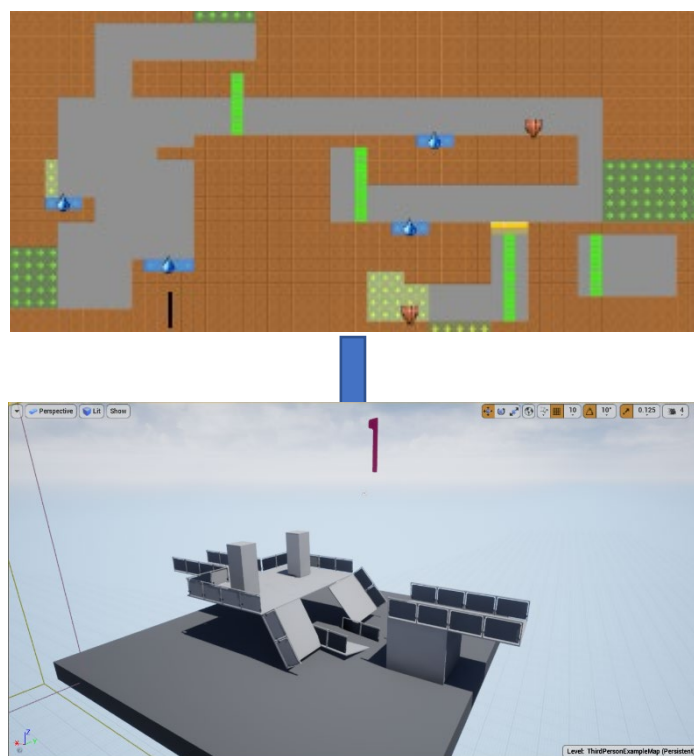
After a couple of tries with the scale and dimension of these chunks, I came to the conclusion, (after numerous try and playtests) that the dimensions were going to be 10 by 10 and the first series of chunks were going to develop only on two dimensions not taking into account verticality. The element 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 chunks with the sense of verticality, without being the only focus (the player will note the verticality and use it, but it won't be the main mechanic that the player will utilize in that space). The third series will utilize mainly the verticality as a way to get around the level and the navigation as the puzzle to solve.

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

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

All these gameplay mechanics are established around the player and the space that he occupies, so the gameplay that these mechanics allows are tailored to the relationship between the player and the space.

At the end the result we want is the same core chunks as in Dead Cells but in three-dimension.



Cells v1

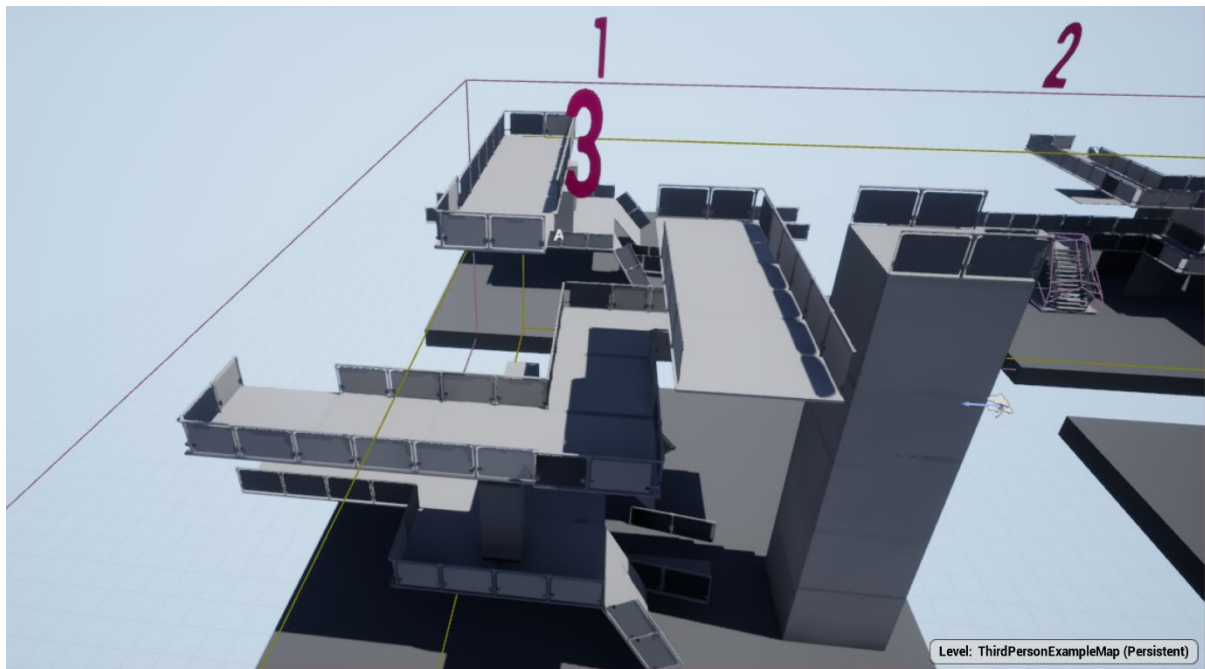
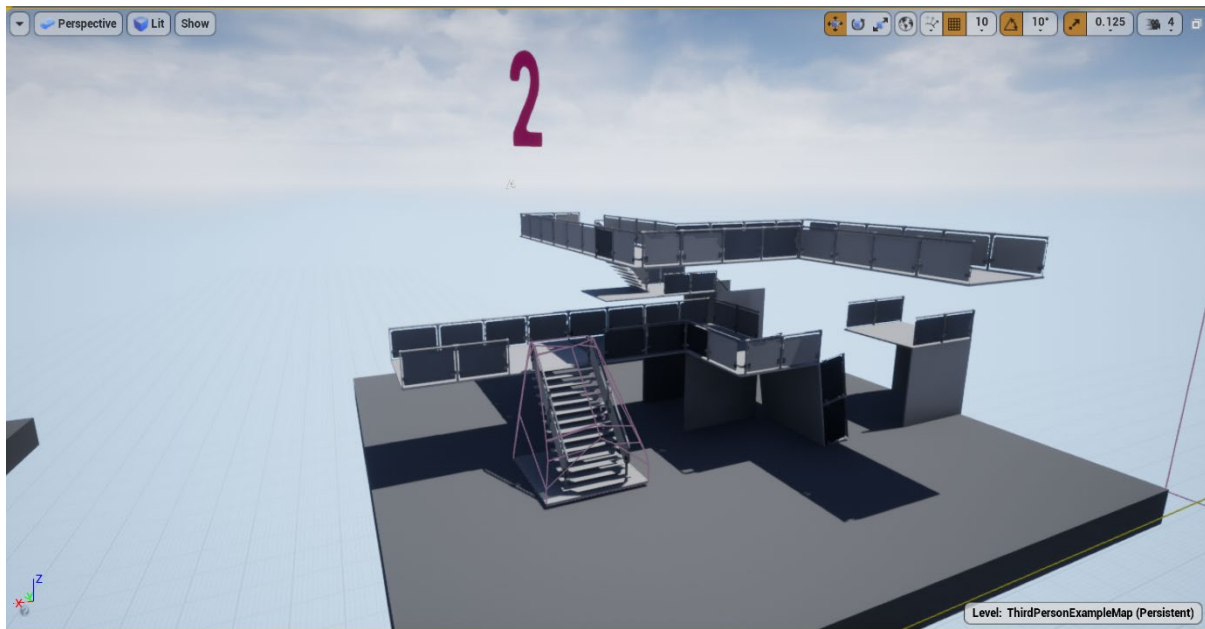
The first batch of cells was made in Unreal Engine. They were experimental cells which I designed them before meeting the client and before we decided to switch from a block basis to a cell-based focus. The block basis was how we approached the building of the level before switching to the approach of 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.

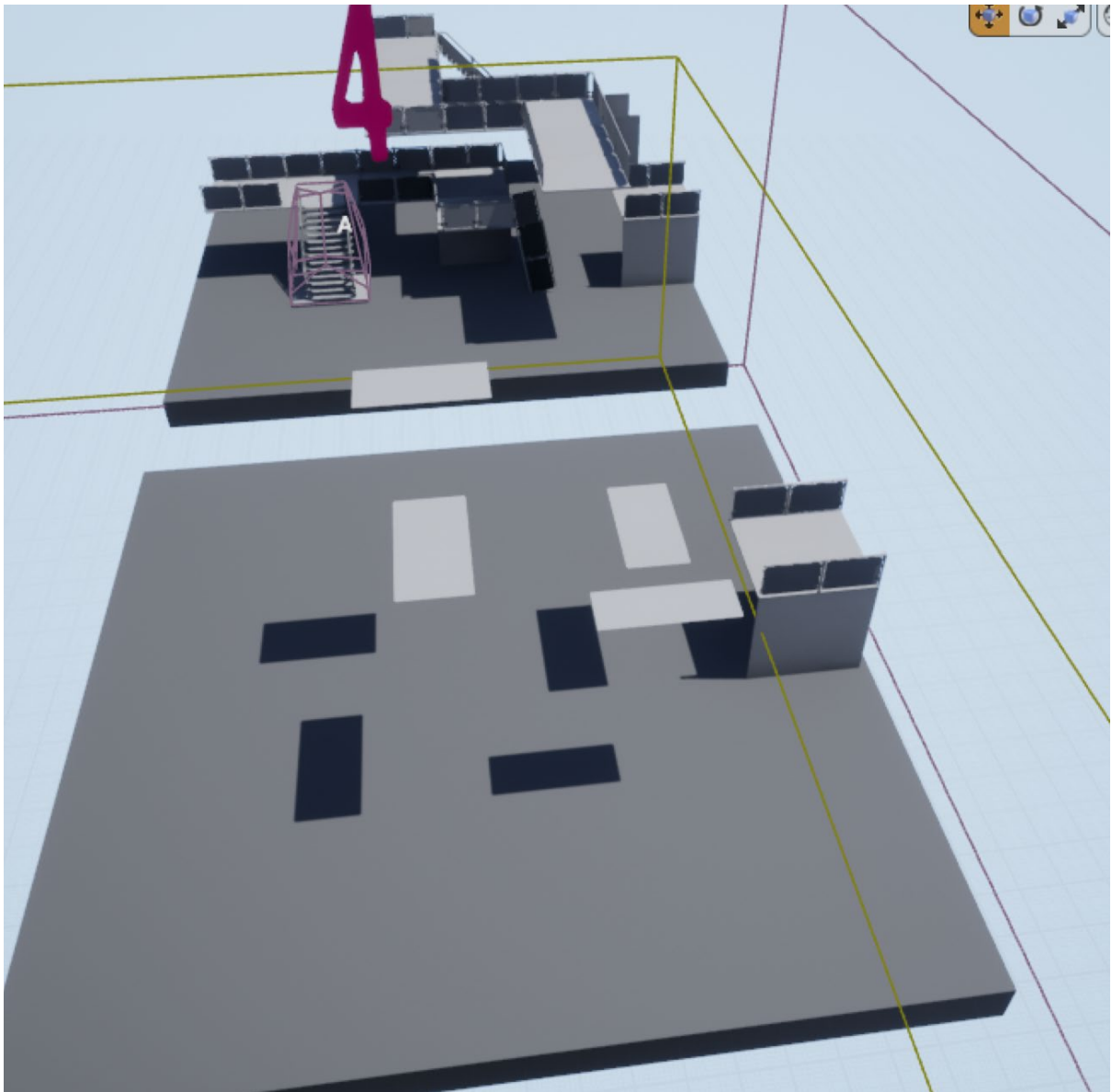
Every cell is centred around the same mechanic: the jump. This is because at this stage I wanted to design only a handful of cells to get a good pre-vis for the scale of the cells and how they would interconnect. At this stage we did not have a clear vision from the client of what kind of gameplay they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



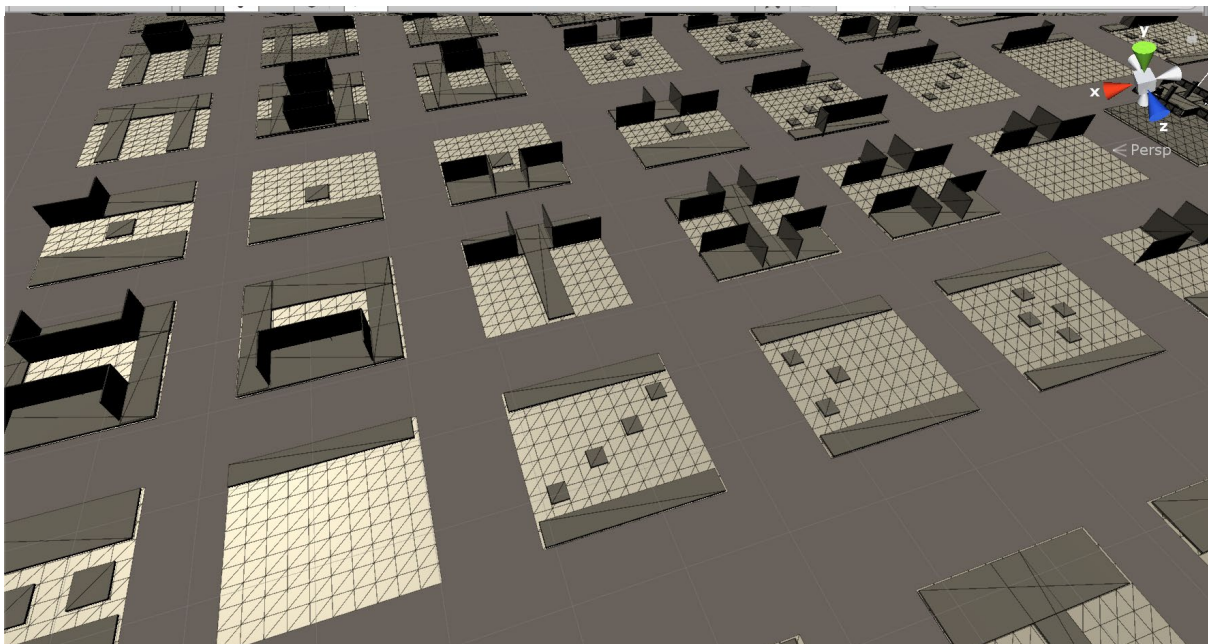
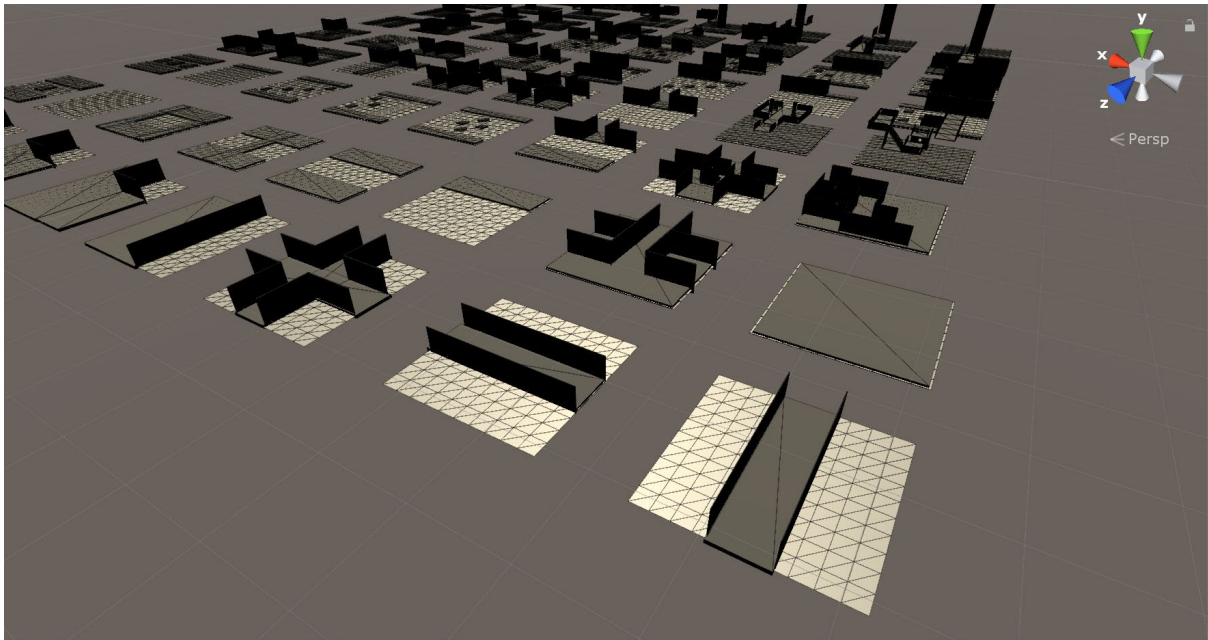
Supporting documentation regarding the gameplay cells

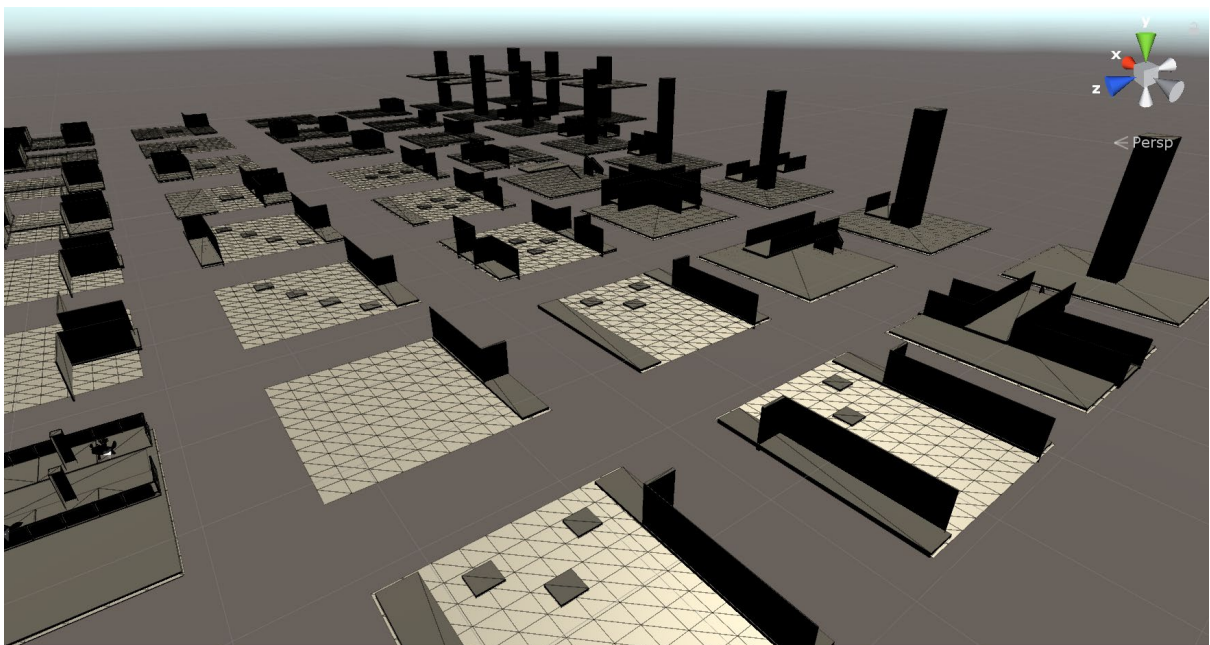
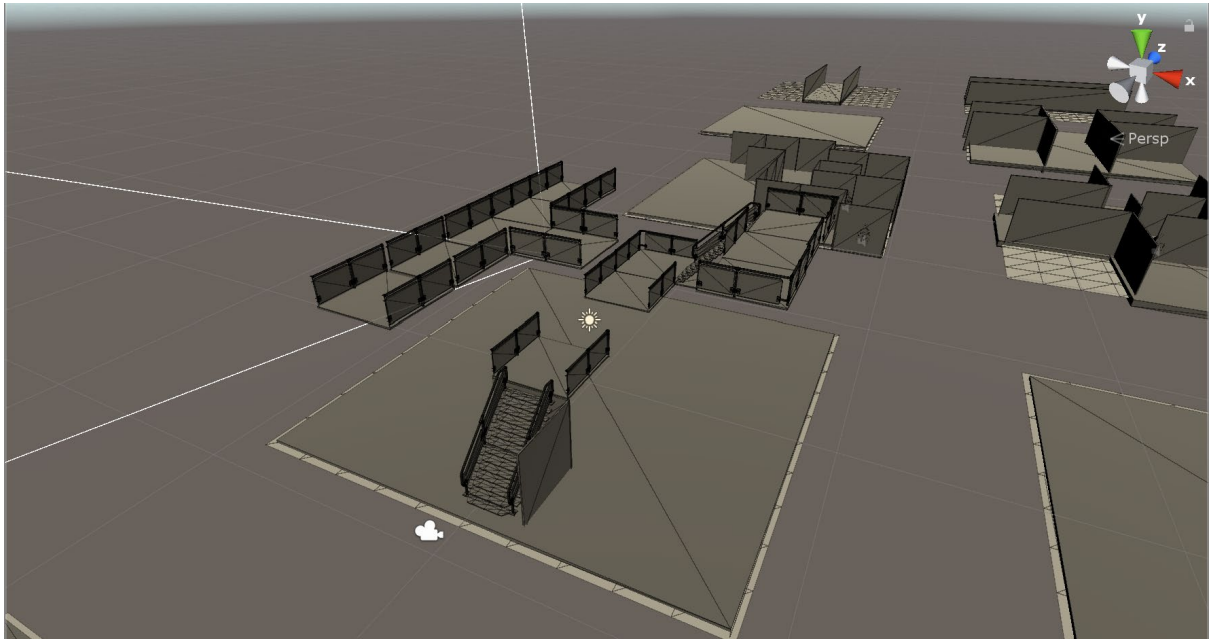


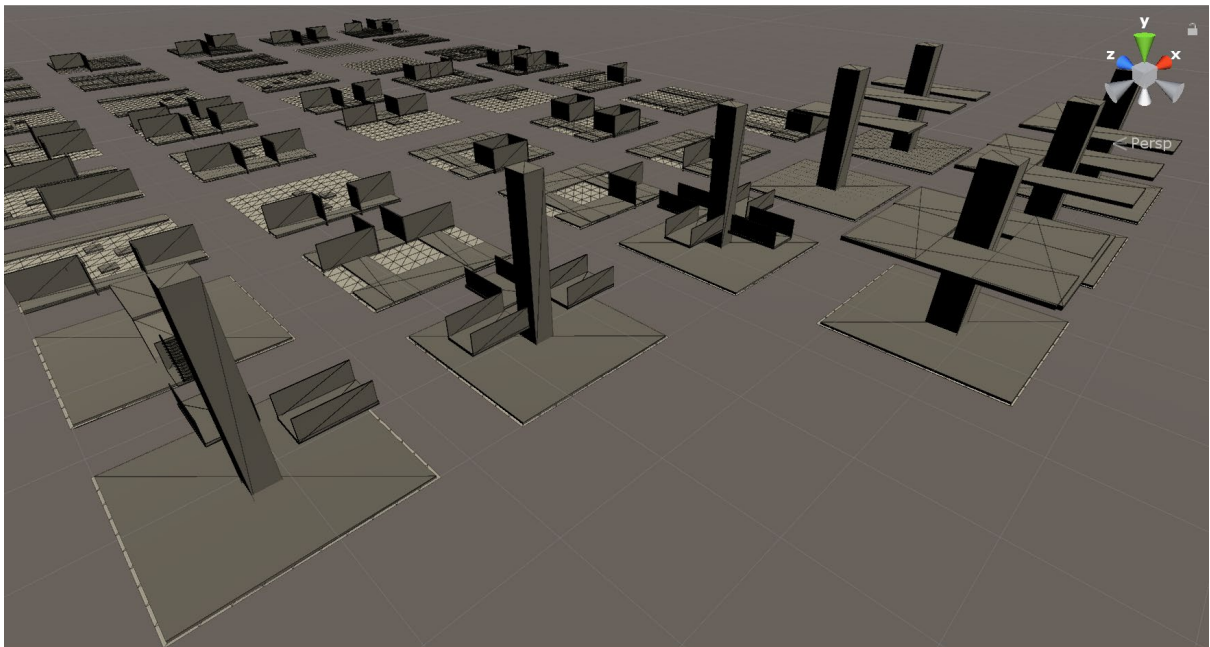
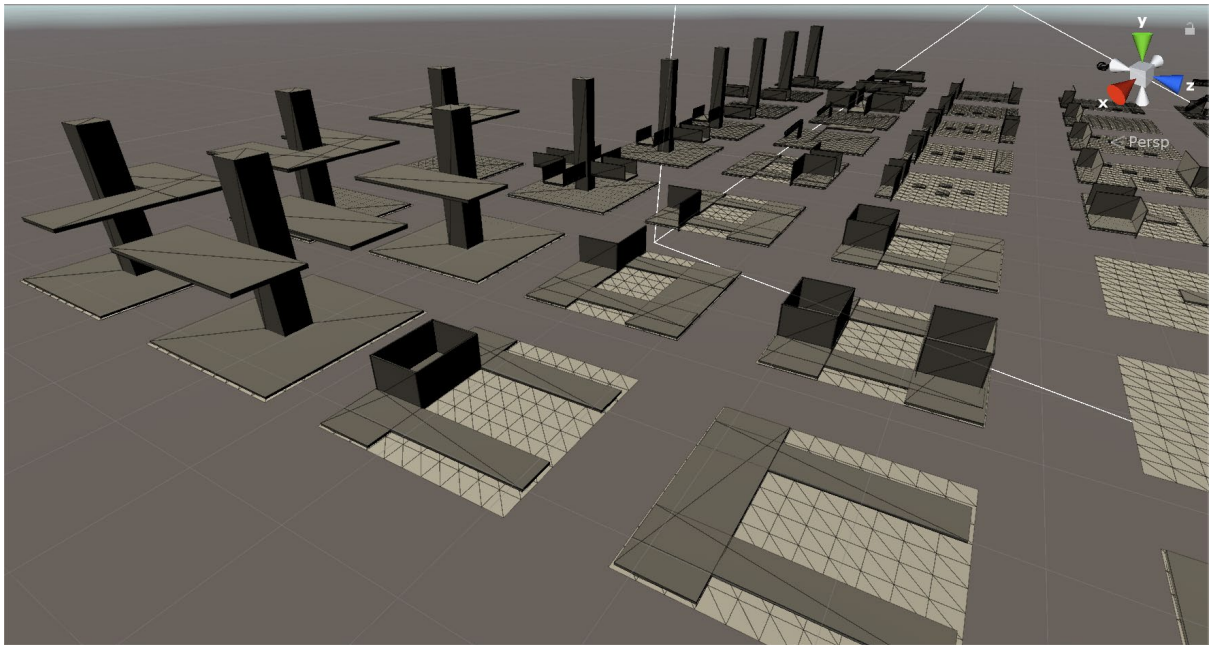


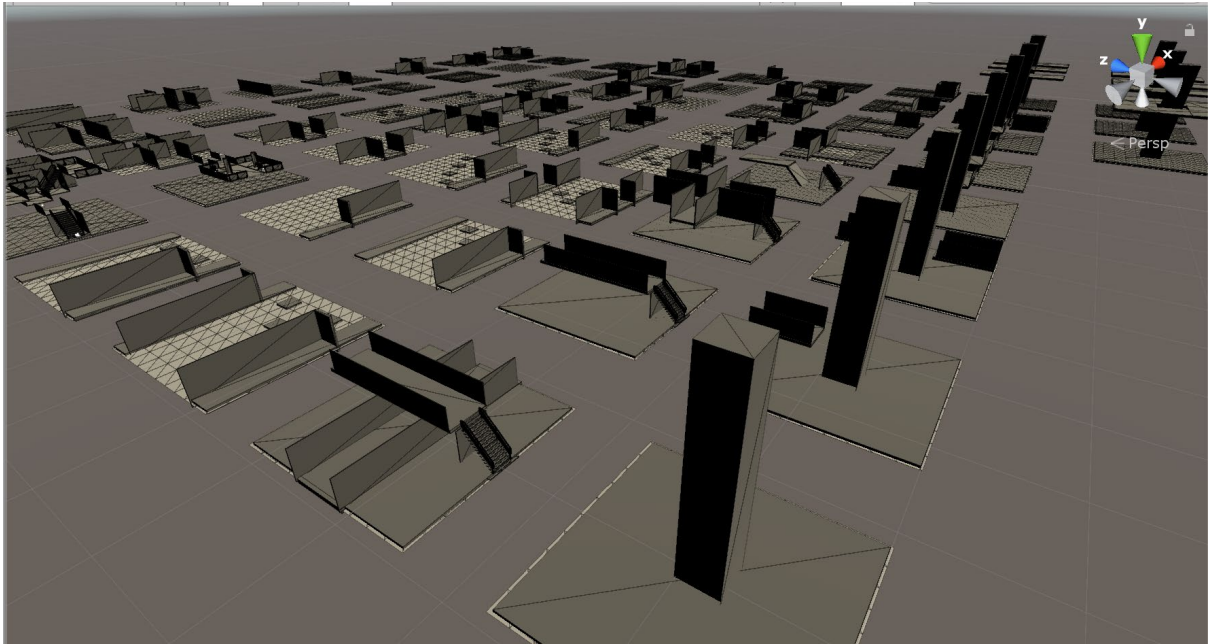
Cells V2

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.









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)

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

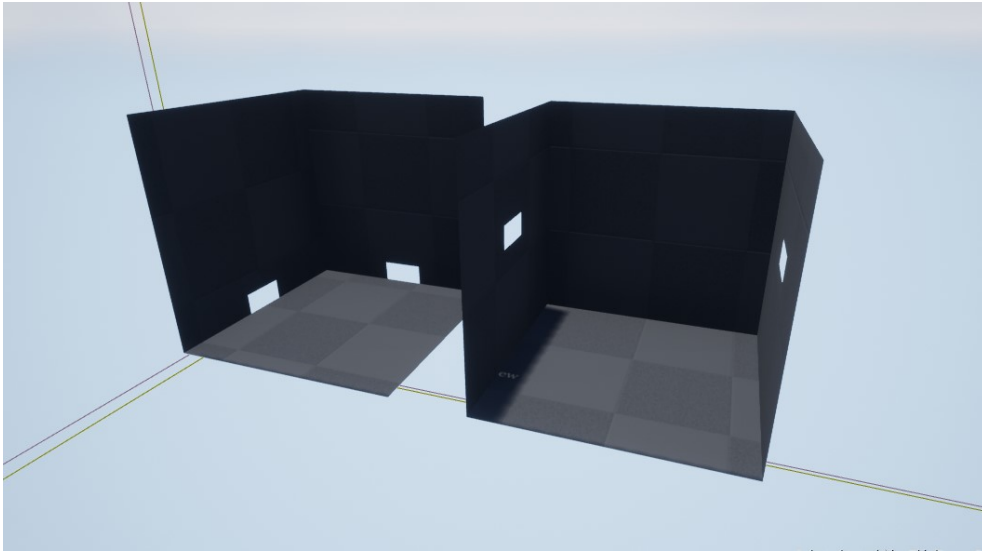
Supporting documentation regarding the gameplay cells

- Concrete pillars: 0.8 x 0.8 x 3

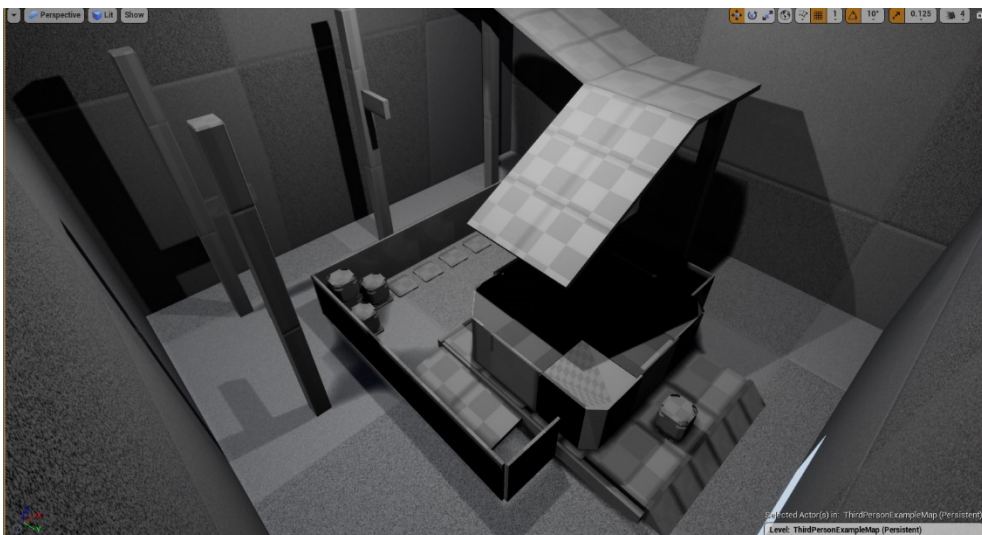
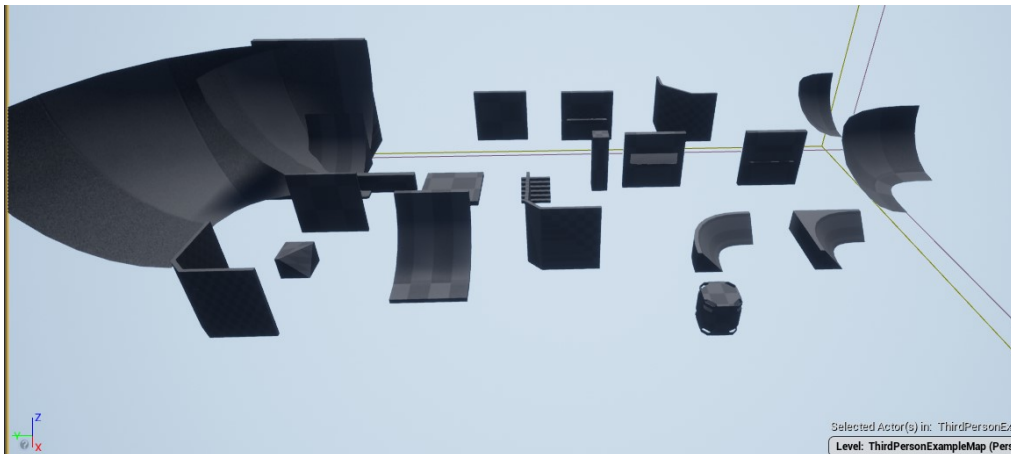
Supporting documentation regarding the gameplay cells

Each final cell is made of two elements:

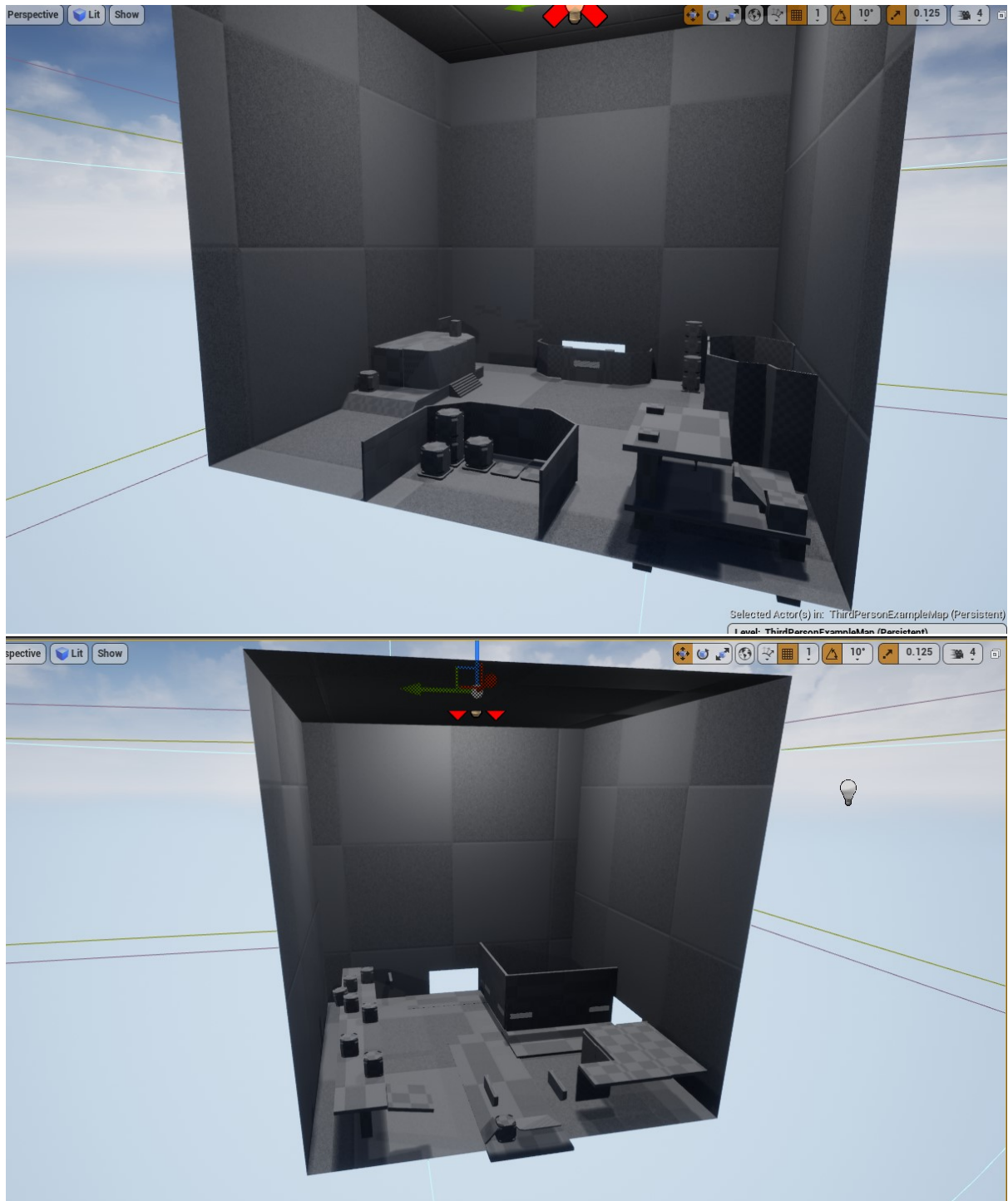
- Core cell layout. This core layout is a cube with two holes that are the entry/exit



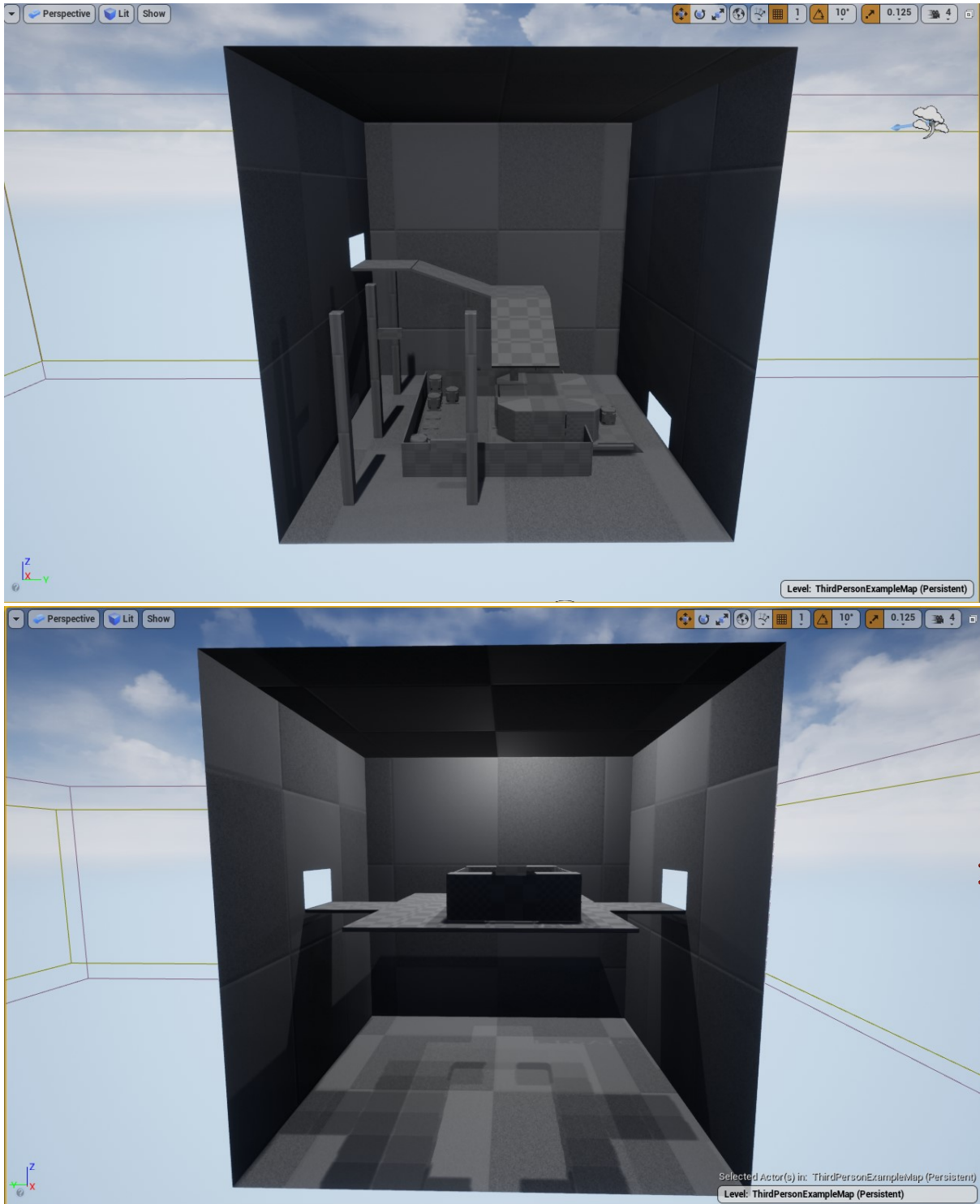
- Props. Assets made by the art team



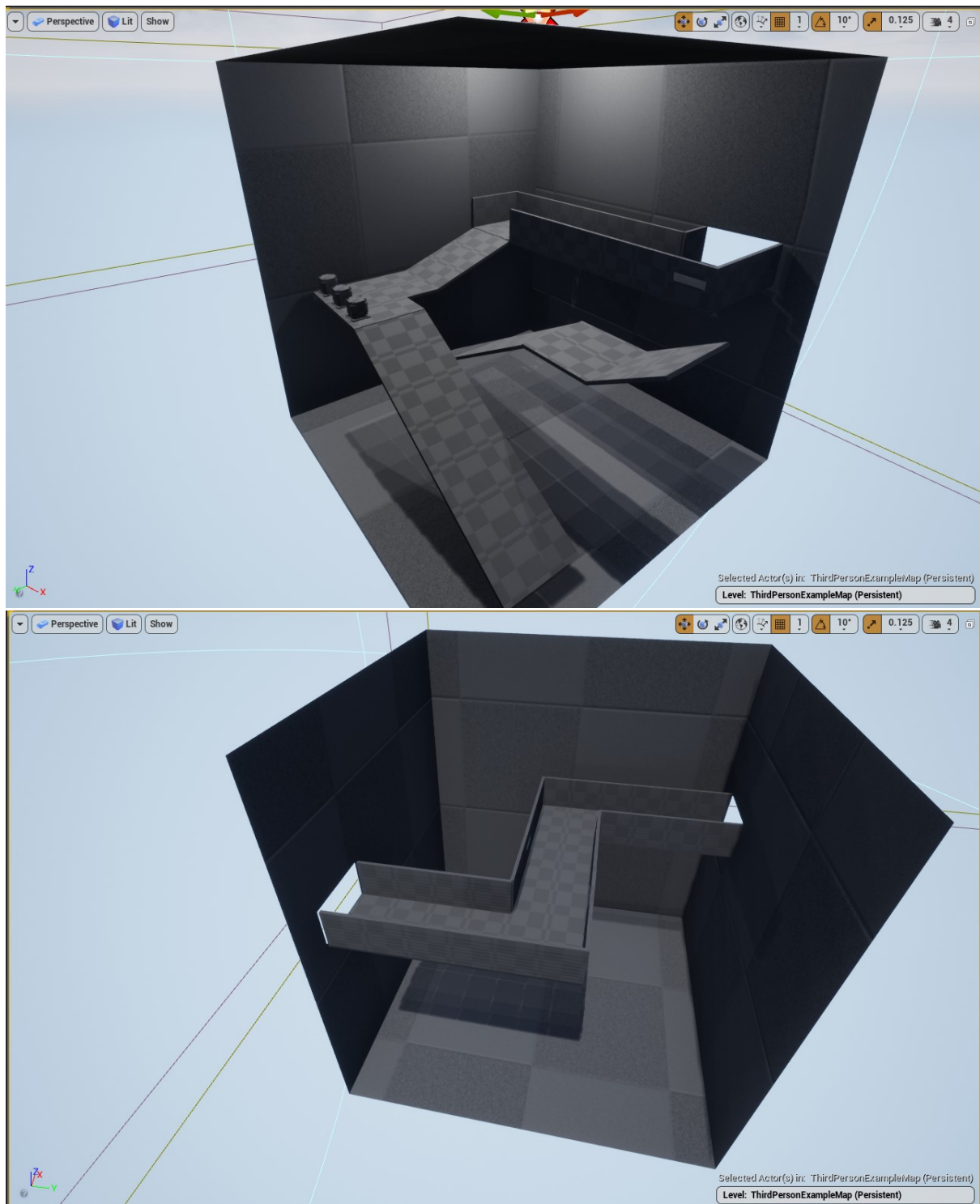
Here's a couple of designs for the cells (v3)



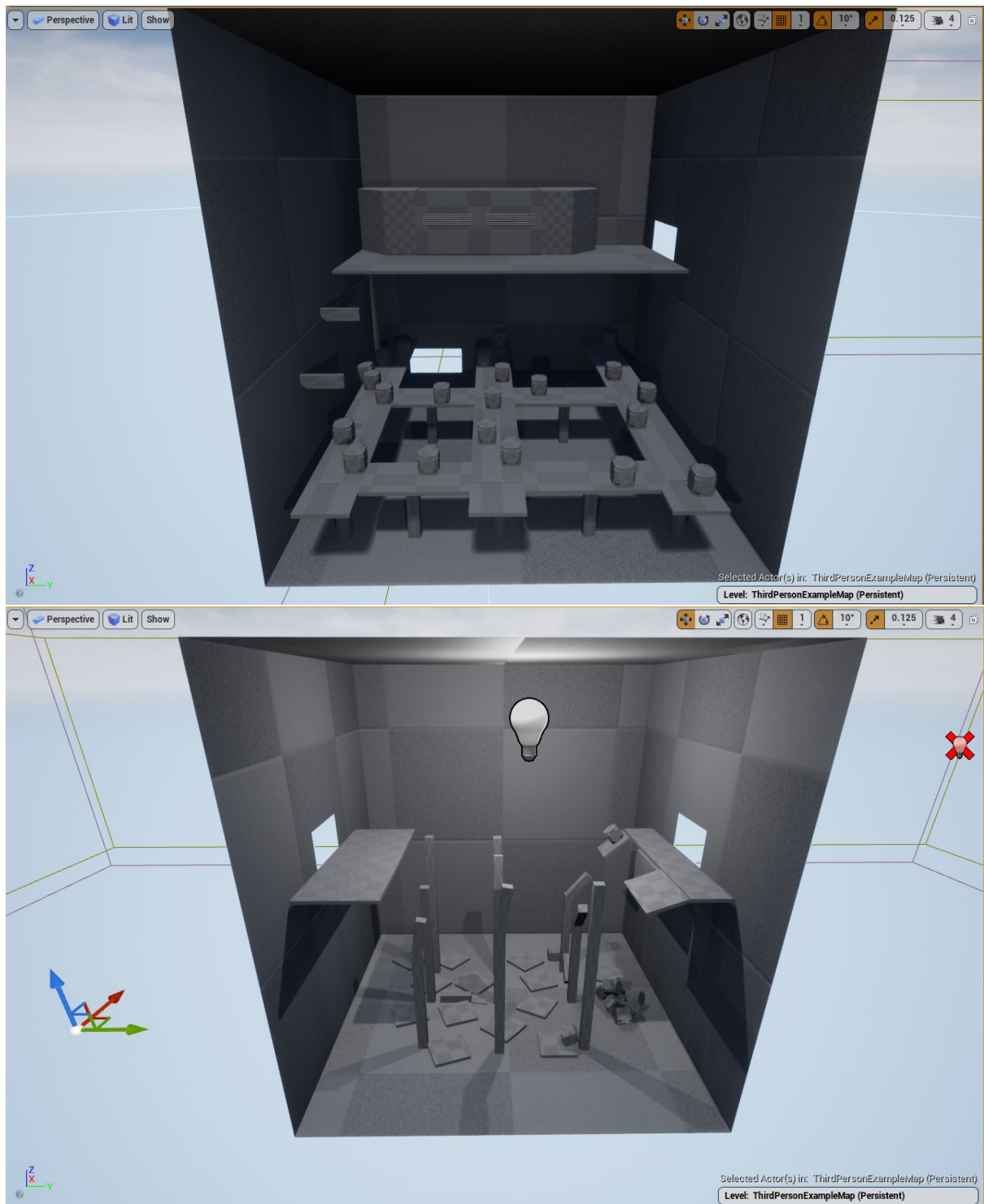
Supporting documentation regarding the gameplay cells

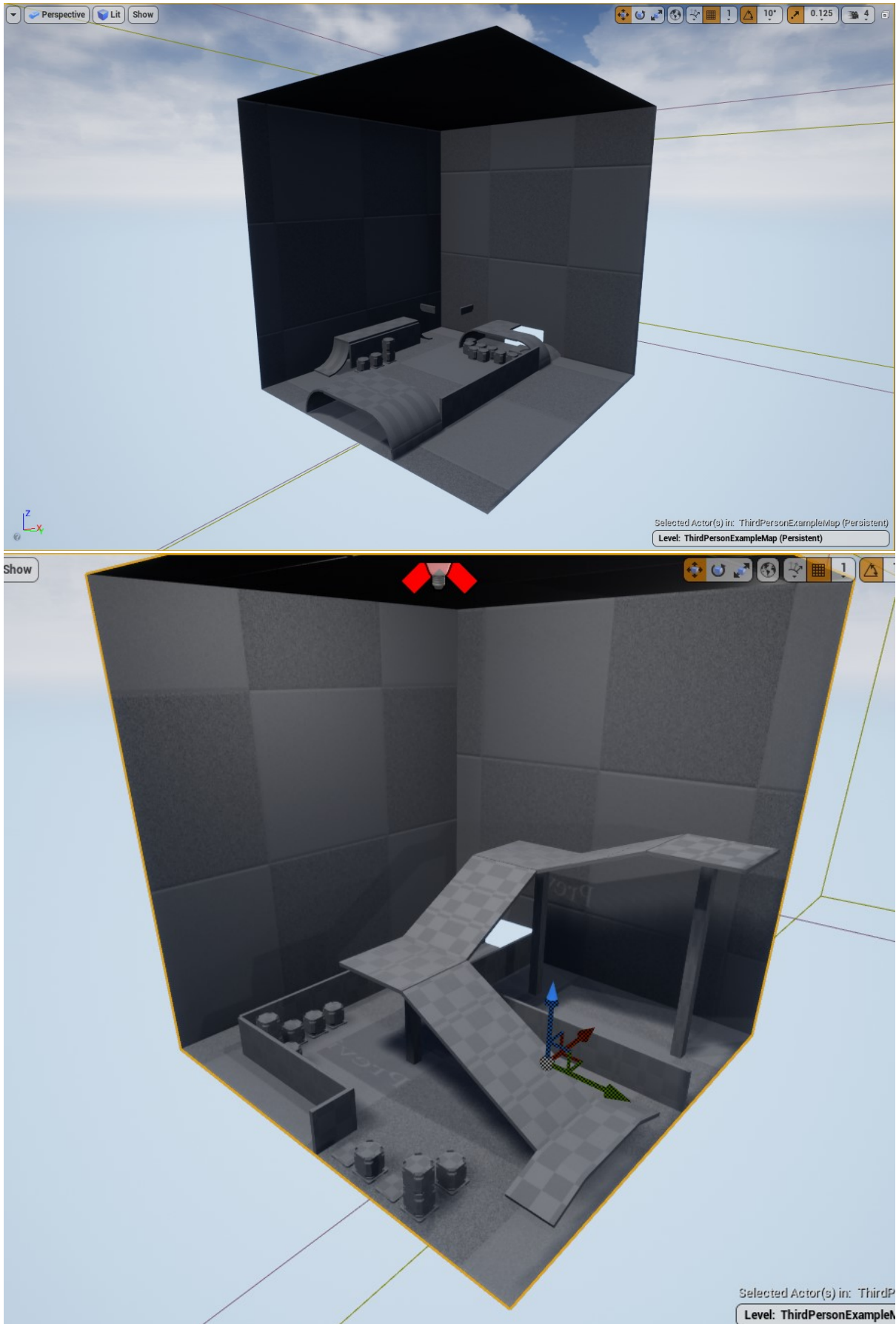


Supporting documentation regarding the gameplay cells



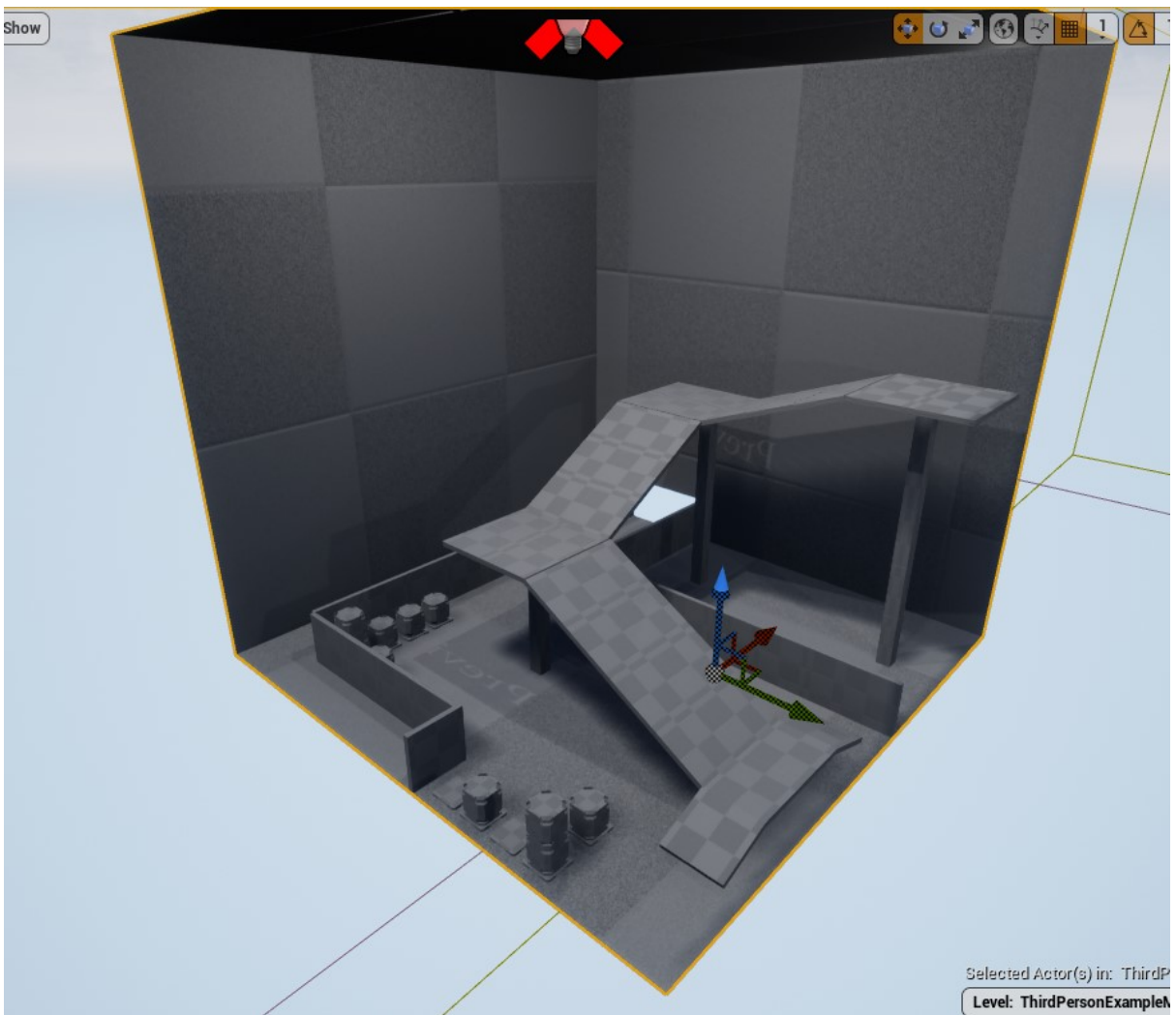
Supporting documentation regarding the gameplay cells

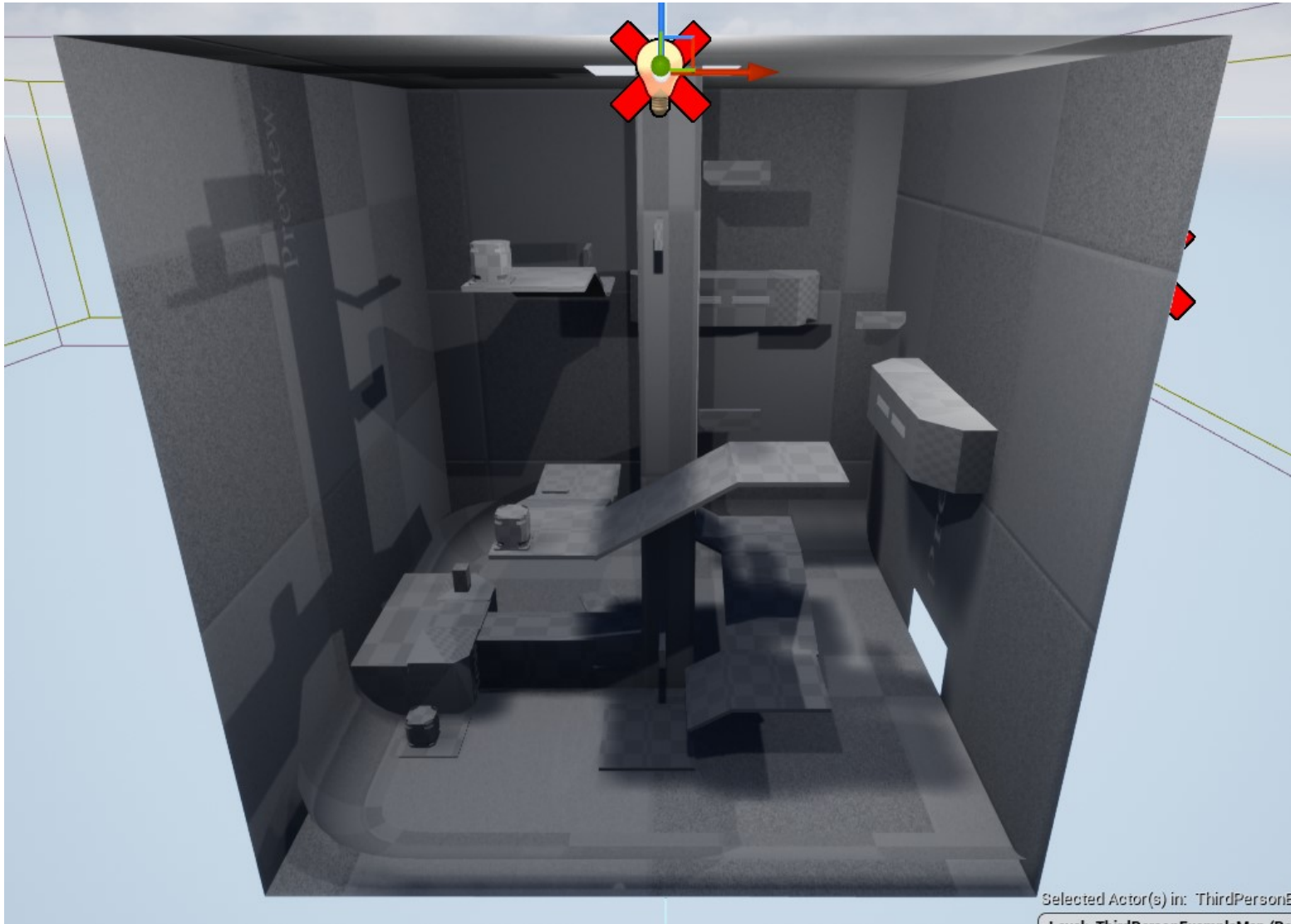






Supporting documentation regarding the gameplay cells





Selected Actor(s) in: ThirdPersonE
Level: ThirdPersonExampleMap (0/2)

This is a cell designed for vertical movement. It has only one entrance on the right wall, and the exit is placed at the top. It is possible to exit the cell only through verticality.



Selected Actor(s) in:

Level: ThirdPersonE

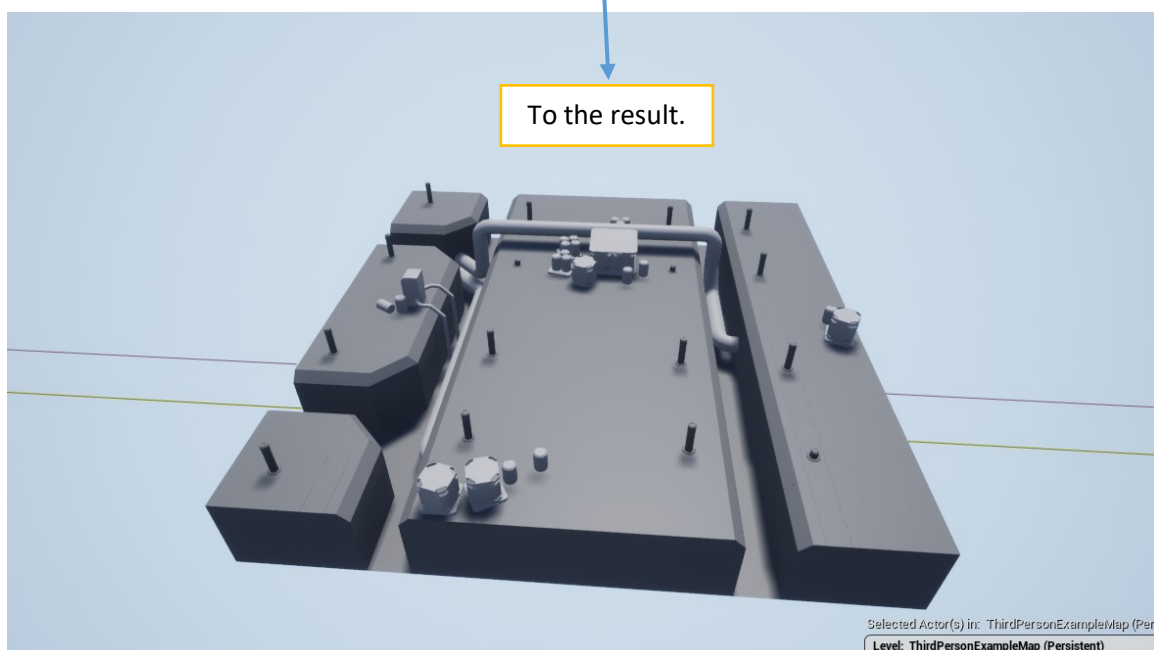
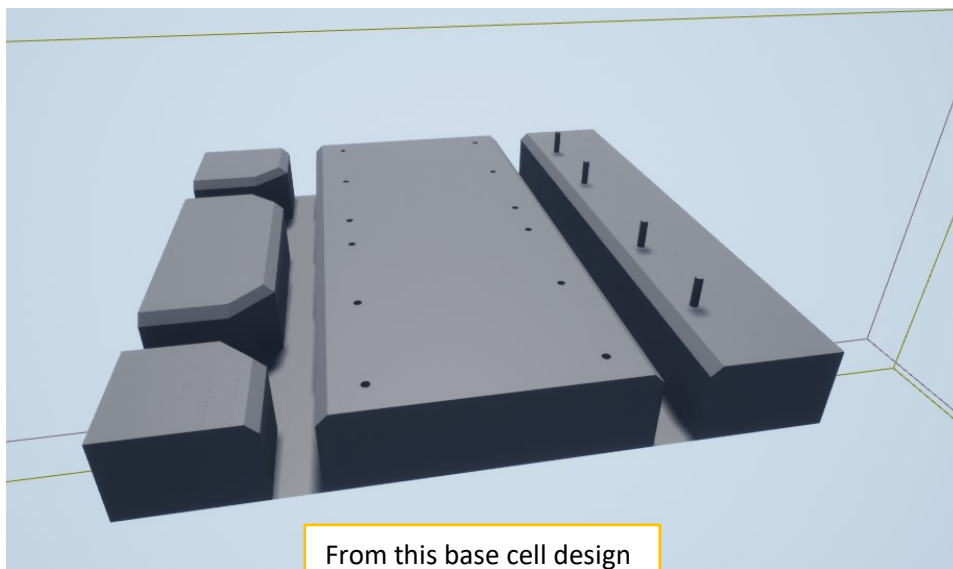
This is a cell designed for vertical movement. It has only one entrance on the floor connecting to the other vertical cell, and the exit is placed in a wall. It is possible to exit the cell only through verticality.

Cells V4

Due to more technical constraints, the design and base of the cells were changed once again.

The design idea was to have a pre-defined cell that would be adorned with other props made by the artists. This can be seen from the two images below, the first one shows the base cell, while the other one shows the same cell adorned with various props to give some personality and avoid a sense of repetition and visual fatigue to the player. The positions of the props are not only cosmetic, but they serve also for mechanic, dictating pacing and giving the player a visual aid as where to go.

Example: This cell above is the base cell. It must be adorned with props to make the traversal interesting, while also keeping the cell visually interesting. The assets placed in these cells are designed with modularity in mind. This is to help with the construction of cells and to cut down on “construction” time.



Here's a few examples of finished cells made by the artists following the designer guidelines.

