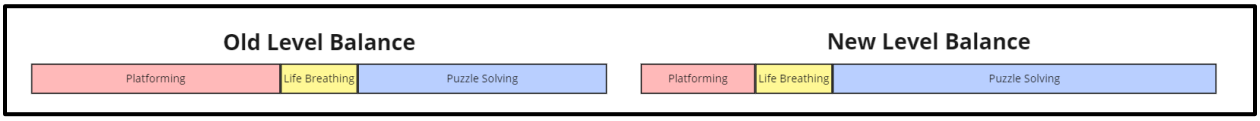
Creating Player, Level, and Camera Metrics that Accommodate Puzzle Gameplay

Joel Bergen

Introduction

Our game loop began with an even blend of puzzles and platforming in its levels. During development, we found that most of our mechanics supported puzzle gameplay, but didn't fit directly with platforming.

This left us with two choices: we could rescope to introduce new mechanics that supported our platforming gameplay, or we could de-emphasize platforming and focus on supporting puzzle gameplay. Given the short amount of development time our team had left, we decided to pivot to level design primarily focused on puzzles.



Design Problem

This change introduced a new problem to our game. Our game metrics were designed around platforming, which meant that the player had a very agile jump. This created issues with puzzle-focused issues, as for an area to be inaccessible, it had to extend almost entirely off-screen. This made it tough to include information that was visible to the player but out of reach.

The above diagrams show the player's old jump capabilities (green) alongside the distance required to make a jump visibly impossible (blue). Note how in both examples the blue distance occupies most of the screen.

Any change to metrics would require all existing levels to be redesigned, making it important to solve this problem as soon as possible. Because of this, I made it my goal to solve the following design problem:

What player, level, and camera metrics best accommodate our shift toward puzzle gameplay?

Method

Approaching the Problem

Before I began solving the problem, I first identified what variables were causing the original problem. I found that a mixture of player, level, and camera metrics was the primary cause.

Player Metrics	Level Metrics	Camera Metrics
How fast does the player move? How far/high can they jump?	*What's the size of each level? How long/short should jumps be? Are any jump distances/heights not allowed?*	*How far away is the camera from the player?*

After adjusting one of these factors, I made sure to re-evaluate the state of the other two. This is because all three variables are intertwined, and a positive change to one could harm another.

Brainstorming Solutions

Once I identified the three factors I would edit, I created a test level and adjusted the player and camera metrics. After some tweaking, I was satisfied with the following changes:

- Reduced the player's jump height from 2.8m to 1.2m.
- Zoomed the camera out by 10%.

I then used these new player and camera metrics to establish level metrics. This was important to do as our game has 3 separate level designers working on it. If we left level metrics to whatever "felt right", some levels may have jumps that are 1.9m long, and some may have jumps that are 2.1m long. This would leave our final game with an inconsistent and messy feeling set of levels.

To do this, I tested the capabilities and limitations of the player controller and camera and made min/max values for important level features. This includes:

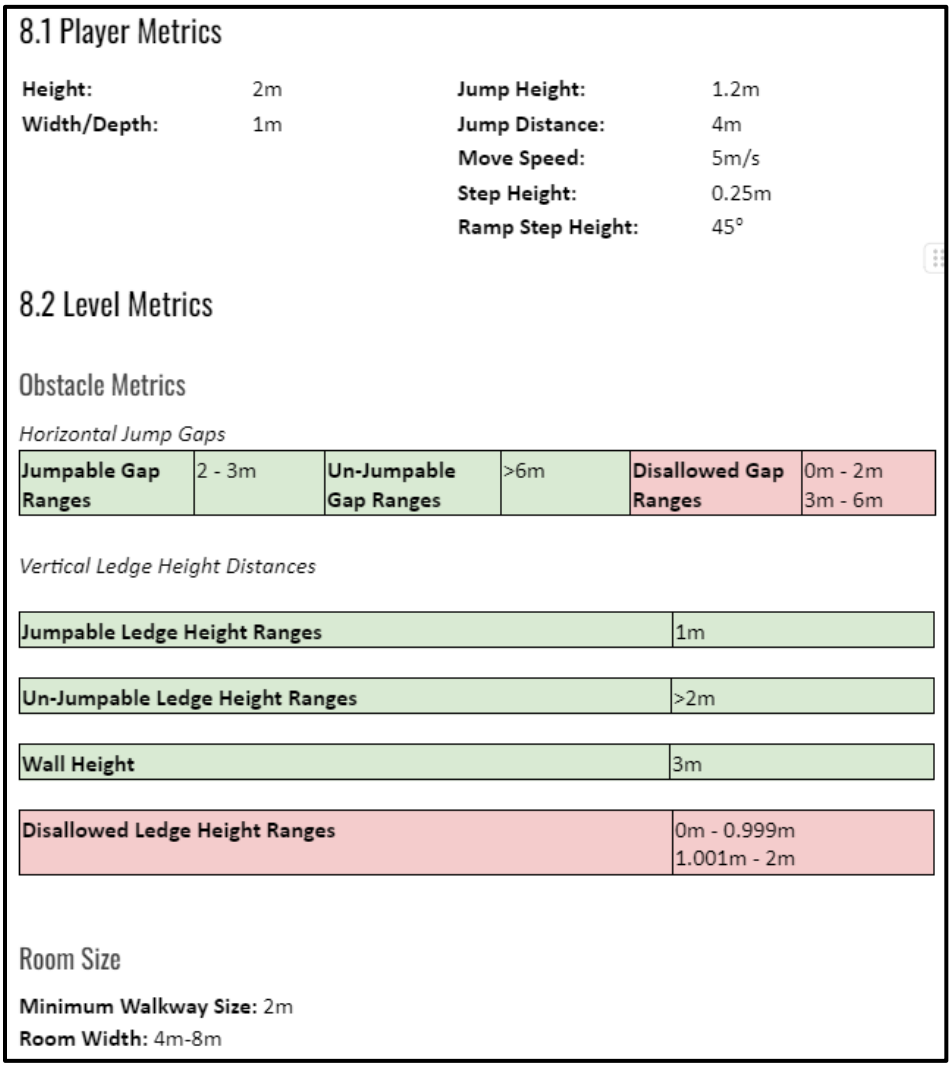
- Min/Max jump distances
- Min/Max jump heights
- Min/Max room widths

I also established dead zones to improve player comprehension of levels. For example, the player has a 4m jumping distance, and therefore 3m-6m is established as a jump dead zone. This means no jump in the game will be within that range so that the player will never be confused by a jump that "looks" possible but isn't.

Implementing the Solution

Establishing metrics is helpful, but it's just as important to make them easily accessible to the team. To solve this, I documented our metrics through our GDD, a Metrics Gym, and a Modular Greybox Kit.

In our GDD, I wrote down what movement capabilities the player has, as well as what jump distances/heights are allowed/disallowed.



8.1 Player Metrics

Height:	2m	**Jump Height:**	1.2m
Width/Depth:	1m	**Jump Distance:**	4m
		Move Speed:	5m/s
		Step Height:	0.25m
		Ramp Step Height:	45°

8.2 Level Metrics

Obstacle Metrics

Horizontal Jump Gaps

Jumpable Gap Ranges	2 - 3m	**Un-Jumpable Gap Ranges**	>6m	**Disallowed Gap Ranges**	0m - 2m 3m - 6m

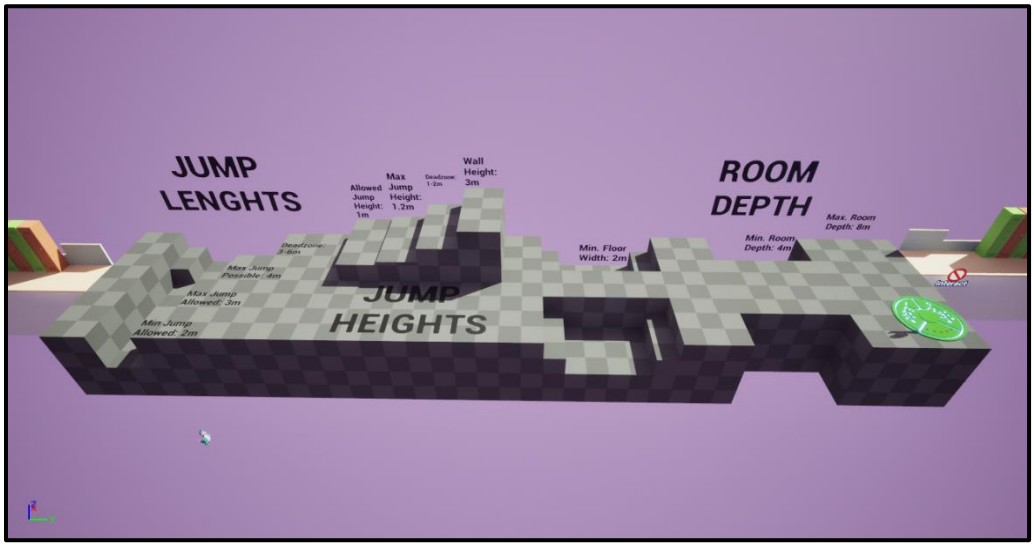
Vertical Ledge Height Distances

Jumpable Ledge Height Ranges	1m
Un-Jumpable Ledge Height Ranges	>2m
Wall Height	3m
Disallowed Ledge Height Ranges	0m - 0.999m 1.001m - 2m

Room Size

Minimum Walkway Size: 2m
Room Width: 4m-8m

I also created a *Metrics Gym* inside of the Unreal project itself, which displayed all level metrics within a playable map. This way level designers could quickly reference level metrics while working without having to open the GDD.



Lastly, I created a modular Greybox kit that abided by the newly established level metrics. This way level designers can use ready-made pieces to build their level rather than having to measure out geometry each time. This will rapidly increase the speed of our level creation while also increasing the consistency of our created levels.

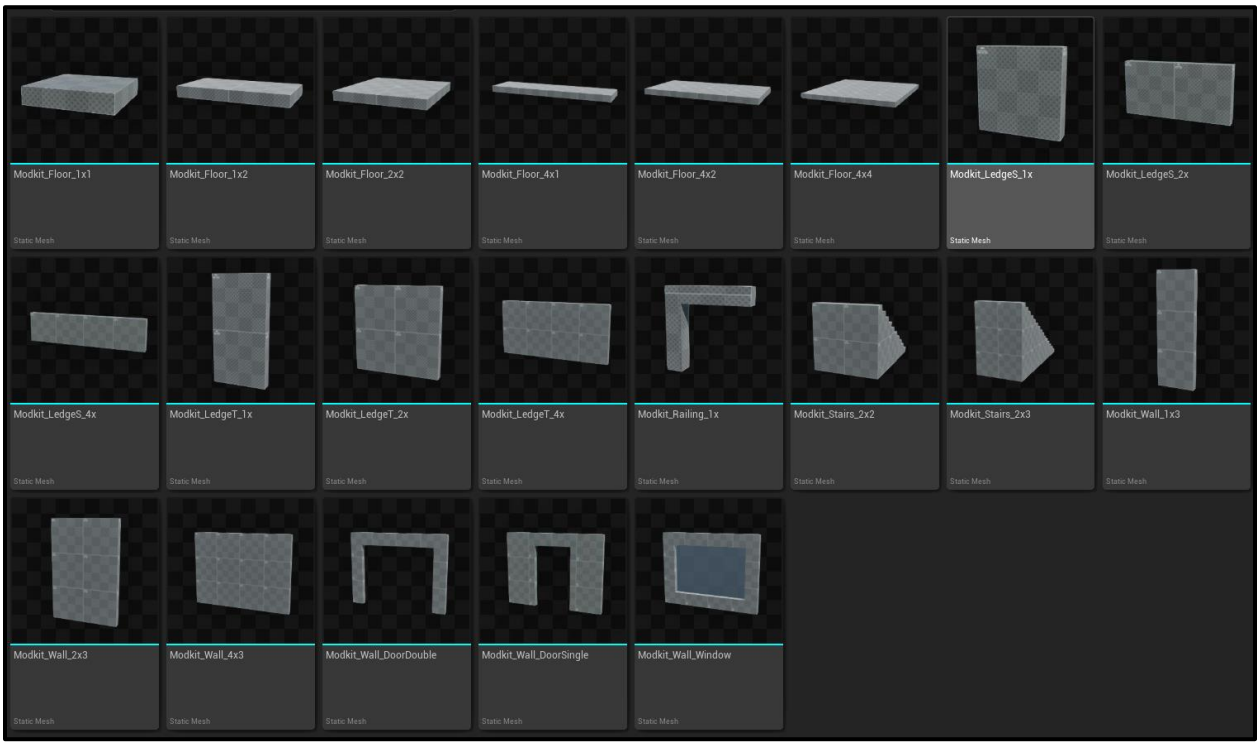
To do this I first created a list of needed level pieces along with their size. This list was then approved by the project's other two level designers, ensuring the kit would give all of us the tools we needed.

Modular Kit Pieces

Piece	Size (LxWxH) (meters)	Notes	Created?
Modkit_Floor_1x1	1 x 1 x 0.2		✅
Modkit_Floor_2x1	2 x 1 x 0.2		✅
Modkit_Floor_2x2	2 x 2 x 0.2		✅
Modkit_Floor_4x1	4 x 1 x 0.2		✅
Modkit_Floor_4x2	4 x 2 x 0.2		✅
Modkit_Floor_4x4	4 x 4 x 0.2		✅
Modkit_Railing_1x	1 x 0.2 x 1		✅
Modkit_LedgeS_1x	1 x 0.2 x 1		✅
Modkit_LedgeS_2x	2 x 0.2 x 1		✅
Modkit_LedgeS_4x	4 x 0.2 x 1		✅
Modkit_LedgeT_1x	1 x 0.2 x 2		✅
Modkit_LedgeT_2x	2 x 0.2 x 2		✅
Modkit_LedgeT_4x	4 x 0.2 x 2		✅
Modkit_Wall_1x3	1 x 0.2 x 3		✅
Modkit_Wall_2x3	2 x 0.2 x 3		✅
Modkit_Wall_4x3	4 x 0.2 x 3		✅
Modkit_Wall_DoorSingle	4 x 0.2 x 3		✅
Modkit_Wall_DoorDouble	4 x 0.2 x 3		✅
Modkit_Wall_Window	4 x 0.2 x 3		✅
Modkit_Stairs_2x2	3x2x2		✅
Modkit_Stairs_2x3	4.5x2x3	Individual steps are 2x0.3x0.2	✅

I then created the level pieces in Maya and scaled their UVs so that a consistent 1mx1m development texture could be applied in-engine.

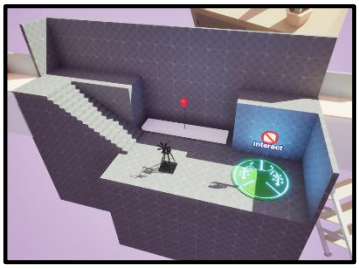
Finally, I imported the pieces into Unreal and set their pivots so that they could easily be snapped together on a grid.

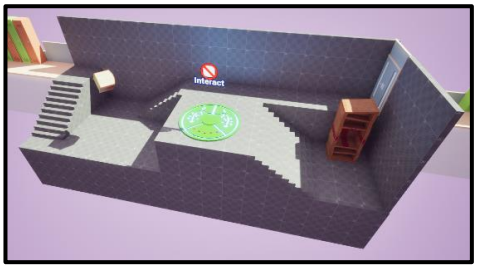


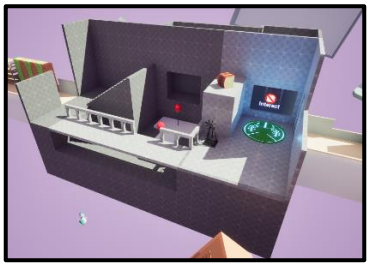
Testing the Solution

To ensure the new metrics could work on a real level, I used the modular kit to recreate the three rooms used in our MVP build. After doing so the levels kept all of their original functionality, while also better matching the puzzle gameplay of our game.

This implementation proved that the new metrics were functional and ready for future levels.







Conclusion

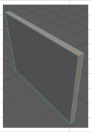
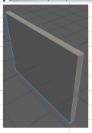
This design problem proved to me just how important consistent metrics are for a full game. On a solo project, you might be able to get by without them, but when multiple people are designing levels it's nearly impossible to build consistent levels off of "feel" alone.

I found the creation of a *Metrics Gym* to be an especially helpful addition, as metrics are only as useful as they are easy to follow. Creating an in-engine example of proper metrics makes it much easier for level designers to abide by them.

Future Steps

The level art kit we used for our MVP build does not currently match our new level metrics. Because of this, we will have to alter existing assets and create new ones to accommodate the changes.

I have already created a list of level art pieces we will need for the game alongside their metrics, and have begun creating some of the pieces. In future weeks I will create more pieces on this list while also creating new levels with the new metrics.

Asset Name	Metrics (L W H)m	Modular Kit	Asset Type	Priority	Animation	Implementation Status	Member	Comments	
floor_tilegrey_1x1	1x1x0.2	N/A	Base			In-Engine	Joel	If easier, one texture could be made, and tinting can be used to get colour variants.	
floor_tilegrey_2x1	2x1x0.2	N/A	Base			In-Engine	Joel		
floor_tilegrey_2x2	2x2x0.2	N/A	Base			In-Engine	Joel		
floor_tilegrey_4x1	4x1x0.2	N/A	Base			In-Engine	Joel		
floor_tilegrey_4x2	4x2x0.2	N/A	Base			In-Engine	Joel		
floor_tilegrey_4x4	4x4x0.2	N/A	Base			In-Engine	Joel		
ledge_drywall_1x1	1x0.2x1	N/A	Base			Not Started	Joel	Make white/grey, and we can tint to colours as needed. To avoid z-clipping, only leave the front and back faces on the geometry. This way the wall won't clip with a floor directly above it. (Delete the orange faces in the image to the right.)	
ledge_drywall_2x1	2x0.2x1	N/A	Base			Not Started	Joel		
ledge_drywall_4x1	4x0.2x1	N/A	Base			Not Started	Joel		
ledge_drywall_1x2	1x0.2x2	N/A	Base			Not Started	Joel		
ledge_drywall_2x2	2x0.2x2	N/A	Base			Not Started	Joel		
ledge_drywall_4x2	4x0.2x2	N/A	Base			Not Started	Joel		
wall_drywall_1x3	1x0.2x3	N/A	Base			Not Started	Joel		
wall_drywall_2x3	2x0.2x3	N/A	Base			Not Started	Joel		
wall_drywall_4x3	4x0.2x3	N/A	Base			Not Started	Joel		
wall_drywall_window_4x3	4x0.2x3	N/A	Base			Not Started	Joel	The window itself should be translucent and 3 x 2 x 0.1	
wall_drywall_rippedtop_1x1	1x0.2x1	N/A	Detailed			Not Started		The tops of walls won't have geometry on their own. Because of this, we'll need to manually add it on parts of the wall where this geo is exposed (border of level) with these kit pieces.	
wall_drywall_rippedtop_2x1	2x0.2x1	N/A	Detailed			Not Started			
wall_drywall_rippedtop_4x1	4x0.2x1	N/A	Detailed			Not Started			
ledge_brick_1x1	1x0.2x1	N/A	Base			Not Started		Make white/grey, and we can tint to colours as needed. To avoid z-clipping, only leave the front and back faces on the geometry. This way the wall won't clip with a floor directly above it. (Delete the orange faces in the image to the right.)	
ledge_brick_2x1	2x0.2x1	N/A	Base			Not Started			
ledge_brick_4x1	4x0.2x1	N/A	Base			Not Started			
ledge_brick_1x2	1x0.2x2	N/A	Base			Not Started			
ledge_brick_2x2	2x0.2x2	N/A	Base			Not Started			
ledge_brick_4x2	4x0.2x2	N/A	Base			Not Started			
wall_brick_1x3	1x0.2x3	N/A	Base			Not Started			
wall_brick_2x3	2x0.2x3	N/A	Base			Not Started			
wall_brick_4x3	4x0.2x3	N/A	Base			Not Started			
wall_brick_window_4x3	4x0.2x3	N/A	Base			Not Started		The window itself should be translucent and 3 x 2 x 0.1	
wall_brick_rippedtop_1x1	1x0.2x1	N/A	Detailed			Not Started		The tops of walls won't have geometry on their own. Because of this, we'll need to manually add it on parts of the wall where this geo is exposed (border of level) with these kit pieces.	
wall_brick_rippedtop_2x1	2x0.2x1	N/A	Detailed			Not Started			
wall_brick_rippedtop_4x1	4x0.2x1	N/A	Detailed			Not Started			
staircase_2x2	2x3x2	N/A	Base			Not Started			
staircase_2x3	2x4.5x3	N/A	Base			Not Started			
railing_1x	1x0.2x1	N/A	Detailed			Not Started			
railingpost_1x	0.2x0.2x1	N/A	Detailed			Not Started			
baseboard_1x	1x0.1x0.5	Trimsheet	Detailed			Not Started			
baseboard_2x	2x0.1x0.5	Trimsheet	Detailed			Not Started			
baseboard_4x	4x0.1x0.5	Trimsheet	Detailed			Not Started			
windowDetail_1x2	1x0.1x2	N/A	Detailed			Not Started		Opaque, cant see through, solid, no look	
windowDetail_2x2	2x0.1x2	N/A	Detailed			Not Started		Opaque, cant see through, solid, no look	
chair	1x1x1	N/A	Detailed			Not Started		metrics are rough ballpark	
desk	1x1x1	N/A	Detailed			Not Started		metrics are rough ballpark	
locker	1x0.75x3	N/A	Detailed			Low Poly		Make tinted material variants	