

Level Construction

Summary

Following are the principles followed to create the levels/environments of *Impostors*. In addition, certain decisions are explained so that the trial and error process can be skipped for later levels. At the end, common feedback or other noteworthy level-related comments will be listed.

- [Summary](#)
 - [Level Planning](#)
 - [Blocking Volumes](#)
 - [Placement of Interactables](#)
 - [Room Scale](#)
 - [Other notes](#)
 - [Level Setup](#)
 - [Level Decisions](#)
 - [Wall Height](#)
 - [Prop Placement](#)
 - [Optimizations](#)
 - [Modular Construction](#)
 - [Merging Meshes](#)
 - [Feedback](#)
-

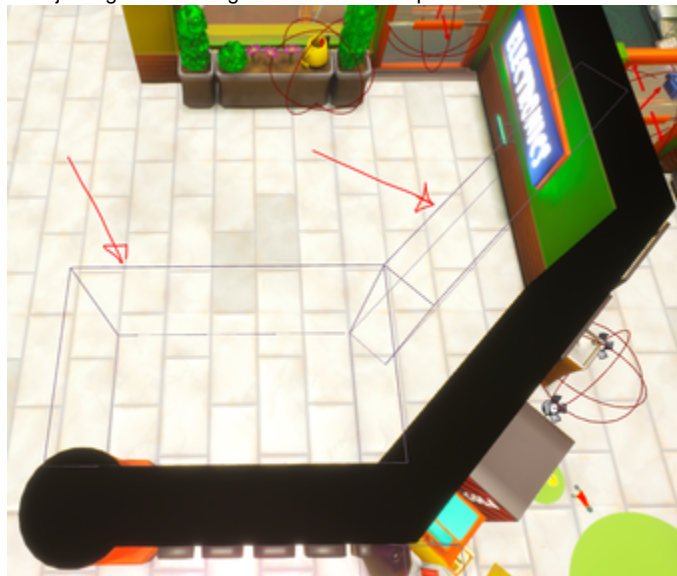
Level Planning

The general concepts of level planning. This details the intentions behind blocking volumes, placement of interactable, room scale, and various other macro level elements.

Blocking Volumes

The single largest impact on level creation is the requirement for blocking volumes. These invisible walls prevent players from hiding behind walls but also eat up considerable space in the level. Early levels (Supermall) were initially planned without this in mind, and have become heavily distorted and difficult to work with as a result. Ensure blocking volumes are planned early on.

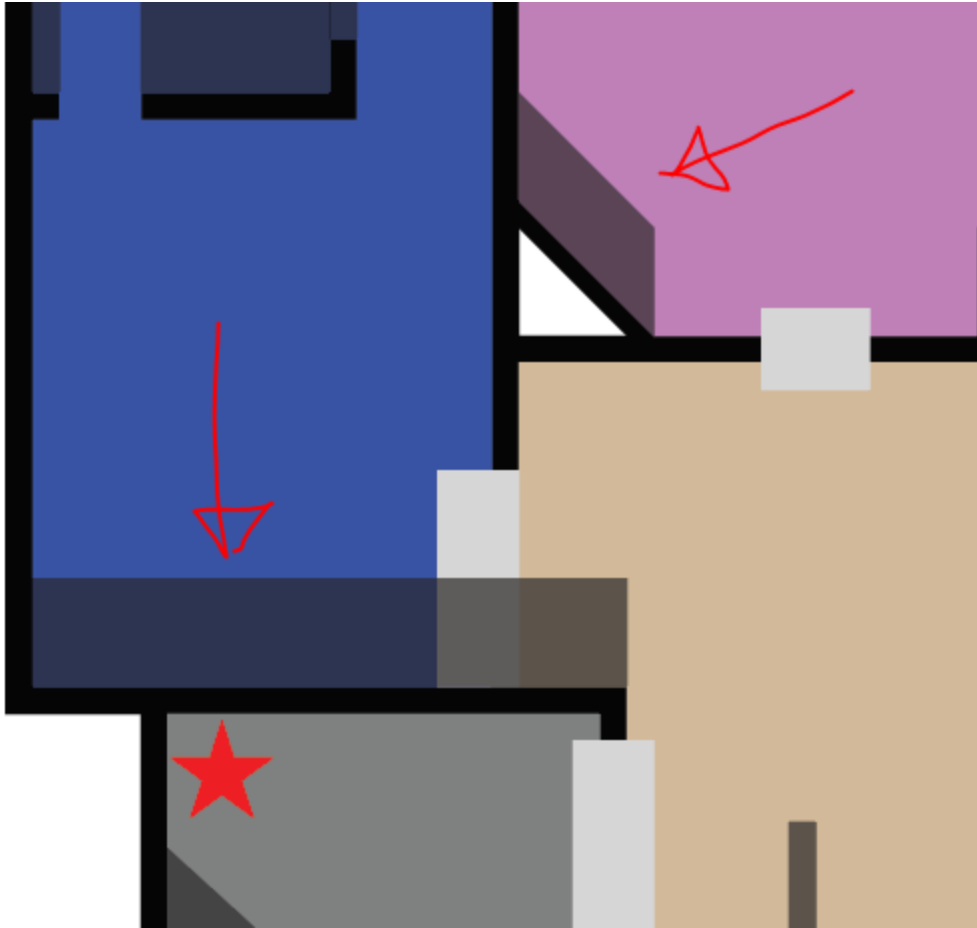
Whenever a wall is place (or over object that hides a significant portion of the player model like a shelf) a blocking volume should be placed behind it as indicated in the image below. This is an Unreal Engine tool found in the 'Place Actors' panel. Note: adjusting the scale through the 'Brush Settings' options is cleaner than adjusting scale through the 'Transform' options.



Blocking volumes should extend the entire (horizontal) width of the object, and should be the height of the walls; this prevents players from climbing on top of them. [Walls are 350 Units tall](#), at this height, the blocking volume should extend 300 Units behind the wall. For shorter objects, the volume should extend less - this is determined by feel, some object feel good with a 200 Unit blocking volumes, others should on be 50 Units - there isn't a mathematic equation to determine blocking volume depth.

Blocking volumes are not needed behind transparent walls, they exist exclusively to prevent players from being unseen behind walls and objects due to the camera angle.

When Creating sketches, it is best practice to add in the blocking volumes to best see how they will effect the dimensions of the level. See below for how this can be done in Adobe Illustrator. If the grid scale in Illustrator is consistent with the scale used for modular assets in the UE4 level, this should work itself out.



Placement of Interactables

Interactables include game objects such as impostor holes, sabotage fix locations, doors, the emergency meeting button, and innocent tasks. Following are considerations to be made for each interactable when designing a level. Most of this has been learned through feedback or are necessary to make a level that is balanced for both Impostors and Innocents so skill determines success for the players.

Innocent task placement, surprisingly, is the least important of the interactables. Players don't seem to notice or care when they move or are changed. That said, their placement should influence other aspects of gameplay like [prop placement](#) or the locations of other interactables.

- Tasks should be evenly distributed in the rooms they occupy.
- Tasks should have sightline to at least one other task (Unfortunately, this is difficult in levels like Supermall, as not enough minigames have been made at the time of writing to fill the space with tasks.)
- Tasks should be against walls.
 - Players don't like to be standing in the middle of rooms.
 - This also ensure the entire playspace is used and there isn't wasted area.

The emergency meeting button is used to gather everyone in a central location on the map and call a meeting. Players will use this to check for dead players without investigating the entire level, prevent a kill if they felt chased, or report sus activity.

- The emergency meeting button should be in a large, central location viewable from all angles.
 - A player should never feel like they died because the side of the level they were on was too far from the button.
 - The further the button is from the center, the less Innocents will use it. Even to their detriment.
- The emergency button should be near spawn.
 - Players don't want to travel to call follow-up meetings.
 - This makes the emergency meeting button's location obvious to new players immediately upon starting a game.
 - Interactables like the sabotage and task locations are indicated to the player, and new players will simply follow the arrow.
 - The emergency meeting's location is never indicated to the player through UI.

Doors can be used to slow down innocents or trap them in rooms with the Impostor.

- Most rooms should have doors.
 - rooms with no doors are inherently unbalanced towards the innocents.
- There should be no more than three doors between any one location and another.
 - Too many doors quickly becomes frustrating and abusable.

- Aim for two doors as the norm.

Sabotages exist in the game to break up player groups or move players to a specific location. In some circumstances, they can also act as a win condition for the Impostors.

- Sabotages are most effective on the outer reaches of the level.
 - This allows the imposter to reliably move players either towards themselves for a kill or away from a fresh kill to create confusion.
- Ensure sabotages are not so far that a player cannot move across the entire map and stop them with reasonable time.
 - Players should not lose for being in the wrong place at the wrong time.

Impostor holes exist to allow the Impostor to travel across the level quickly and unseen to get kills, avoid being spotted, and to sow confusion. The Impostor hole is the most important interactable in terms of placement and its impact on gameplay. It interacts well with all of the above features in unique and interesting ways that should be considered.

- There should be an Impostor hole near all sabotages, at least within the same room.
 - This allows Impostors to reliably get safe kills when using a sabotage.
- Impostor holes should not be seen from room entrances.
 - An Impostor should be able to see a room is clear, and enter. Being caught because a player you didn't see was just outside the room's entrance feels unfair.
- There should be no Impostor holes in central areas or near the emergency button.
 - It feels unfair as an innocent to be unable to call a meeting because the impostor popped up behind them.
 - Central areas are too busy for an Impostor to feel safe using the hole, and that hole becomes unusable.
- Some tasks should be visible from some holes, but not all holes should have tasks nearby.
 - This allows Impostors to justify proximity to holes sometimes, but it's not a guaranteed excuse.
 - This also allows Impostors to find Innocents who are preoccupied.
- Avoid Impostor holes that cut clear across a level - especially diagonally.
 - Players find this too powerful.
- Avoid Impostor holes that connect to each other in adjacent rooms.
 - Since being near a hole is sus, players will choose to simply walk to the next room instead of use the hole, since this looks more natural.
 - The holes will feel useless to the Impostor.

Room Scale

Rooms scale rules can also go hand-in-hand with prop placement. For example: if a room is large, vision-blocking props can be added to break up sight lines and make spaces feel less open.

Variety of room scale is the most important aspect of them. A variety of rooms scale allows players to have preferences and makes navigating the map feel dynamic. For example: some players may prefer larger rooms because they can see more while other will prefer smaller rooms because it allows them to have more complete information.

- Hallways should be about 5m wide.
 - This allows space for players to move but it's tight enough to allow Impostors to get kills while in transit between locations.
 - Any more narrow and halls become hard to see and navigate.
 - Any wider and they no longer feel like hallways and they become very difficult to encourage player interaction in them.
- Rooms should be larger than hallways, at least 10m squared.
- Consider player count when creating a level.
 - 5-10 and 11-15 are good group sizes to consider.
 - Maps should be created specifically for either large or small groups.
 - Ideally, there will be two less rooms in a level than the maximum group size, this guarantees that at least two player will share a room at any given point.

Other notes

- Some rooms should have hiding places.
 - This shouldn't apply to all rooms, but having rooms that allow a player to hide off the beaten path gives Impostors places to catch a breath and plan their next move and can give Innocents somewhere to hide from Impostors after their tasks are completed.
- The spawn should be as close to map center as possible.
 - Innocents should not have a disadvantage completing their tasks because meetings prevent them from getting far enough to complete them.
- Avoid angled walls that are not exactly 45 degrees.
 - Players using a keyboard can only move in eight directions, in 45-degree increments. Walls that are not aligned with that become troublesome to navigate for those players.

Level Setup

When a new level is created, it is best to follow the following steps to ensure it is consistent with previous levels and works as expected.

- New Levels should be created in the Impostors/Maps Folder.
- Copy/past the World folder from Supermall (or other previous level) into the new level

- This will bring in post-process effects, blocking volumes, base lighting, and other features required for level function and consistency.
- Open the `EncapsulationLevel` and add the new level as a sublevel of `EncapsulationLevel`.
 - If this is skipped, the level will not appear as a selectable level in the menu and it won't load from the Lobby.
- The level should now be playable through the Lobby in-game like others.
- After construction, Ensure the level is surrounded by meshes that have the `GhostContainer` custom collision profile.
 - Without this, ghosts will be able to move infinitely far off the level.
- Room bounds will have to be adjusted individually, per room, per level. This is particularly relevant if the new levels have a new theme with new rooms.
 - Drop in a `RoomBound_BP`, scale it to the room. It is acceptable to have multiple volumes to follow the shape of the room.
 - Open the `DT_LevelTools_RoomData` datatable and add an entry.
 - Fill in the room name and color (color is an editor-only cosmetic so designers can better see these volumes while editing them.)
 - For some reason, changing room color will affect all rooms on the level until the level is reloaded, this is a small visual issue and is expected.
 - In the Details panel for the `RoomBound_BP`, Enter row name for the newly added room in the 'Row Name' section.
 - Room bounds are used to display the location of minigames in the top left as well as indicate to the player which room they are currently in at the bottom of the screen.
 - Room bounds can overlap and intersect without issue.

Level Decisions

The smaller-scale choices for the level creation covering concepts like merging meshes, modular construction, tools for optimization, prop placement, and wall height. Ideally, this section explains why certain things were done with the hope to eliminate the trial and error process from future levels by different people.

Wall Height

Currently, walls are 350 Units tall. This should probably not be changed for future levels without very specific intentions. This wall height was determined in tandem with the camera angle through a trial and error process.

- Wall height determines the depth of the blocking volume needed.
 - Taller walls mean more space allotted for blocking volumes.
 - The taller the walls, the more vertical the level will feel.
 - Most players mention a distaste for the y-axis length of Supermall.
- Shorter walls feel out-of-scale with the player model.
 - Even at 300 Units, the walls feel too small to properly contain players.
 - Shorter walls also limit environment details, as most details are on or along walls.

Prop Placement

Props not only make the level look more alive, but should be used to control player movement. This can be done by placing them along player paths between rooms as well as towards tasks. Placing larger props in the center of paths makes for more interesting movement. Below, I've illustrated some prop placements; the blue line indicates 'boring' movement, red illustrates movement due to prop placement, black illustrates the start and end locations.





Note: the goal isn't to impede the player but to make travel more interesting. This change in direction also give players opportunities to kill, dodge killers, and gather information on player movement.

- Larger rooms should have more path-blocking props.
- Particularly large rooms should make use of taller props that can be used to cast shadows and limit player information to balance the large spaces.

Optimizations

Impostors aims to gather a large audience from more casual gamers as well as NFT collectors and those who do not have dedicated gaming machines. For this reason, Impostors has no room for lack of optimization. Presented are some steps to maximize level performance. Optimizations for art are on the [World Art Asset Creation and Implementation](#) page.

- Camera location is a double-edged sword - the camera is far from the world, so we can get away with less detail but since it is so far, more objects are rendered on screen at a time.
 - A Culling Distance Volume was created for Supermall featuring aggressive numbers, culling things out the instant they are off-screen. This volume can be copied to the new level and adjusted to fit.
- Since [walls are created modularly](#), we cannot rely on backface culling for faces that are hidden by the camera angle. Walls should be created with this in mind.
 - Personally, I had 3 different materials on each wall (top, front, back), and in locations the wall could be seen, I applied an unlit black to that surface since unlit is the cheapest shader model in Unreal.



- Avoid dynamic lights at all costs
 - The only light that should be dynamic is the primary Directional Light which is used to light minigames (which appear dynamically) and to give the players moving shadows.

- Skylights must be dynamic to function properly, that is an engine-side requirement.

Modular Construction

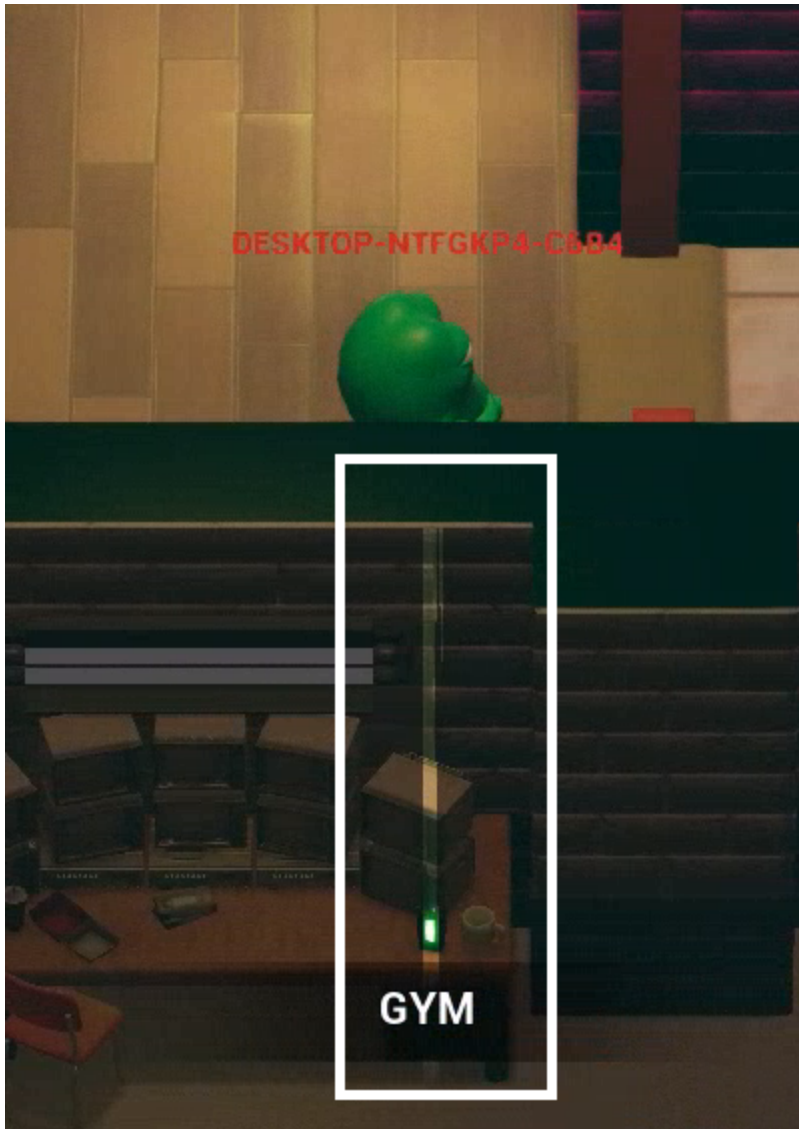
The levels of *Impostors* are built using modular architectural kit construction. This allows for rapid iteration between playtests without leaving the engine to use additional software or creating art issues to solve.

- If a player states a room should be moved a few meters in one direction or another, modular pieces allow this to be done rapidly.
- Modular construction also allows for more level consistency, since everything is built from the same kit.
- Modular construction reduces draw calls, since, for each piece, the engine only needs to call it into memory once - each unique mesh needs its own draw call, increasing frame time.
- Modular construction plays nicely with a grid.
- A level editor is planned for the future, allowing players to create their own levels in-game.
 - With modular assets, the player can be handed a functional set of architectural pieces that are already used in the game.
 - Players will expect to be able to build levels with similar complexity to developer levels, using the same kit as them guarantees this expectation is met.

Merging Meshes

Merging architecture meshes is as important as art meshes for the level. Merged meshes solve several issues in-game and further reduce draw calls, improving optimization.

- Only merge level architecture once playtest feedback comes in almost exclusively positive to avoid redoing work.
- Since floor textures tile based on world position, it is a good idea to merge them to reduce the dozens of floor meshes into a single mesh.
- It is critically important to merge horizontal walls.
 - Once merged, open the assets and create new, solid collision.
 - If this is not done, expect to see the following issue in levels which is caused by the shadow system picking up the separate collision boxes of the modular assets.



Feedback

While developing the level, plenty of feedback or level concepts have been mentioned from developers and playtests. Noteworthy, non-level-specific feedback is listed here so further levels can be aware of potential impediments or shortcomings and avoid them, or lean into this feedback to create greater level variety.

- Skeld, the most popular Among Us level, has the players all follow a single flow loop, with offshoots for rooms. Some players believe this is a superior way to construct a level.
- Since we are using a 3D engine, we should take advantage of that with verticality.
 - Having areas that are above (not on top of) others which allows players to drop into other rooms but not return can have an interesting impact on player movement.
- For Supermall, a 5m base was used for scale. It may be better to use a 3m base, with hallways width being 6m and rooms being at least 12m.
 - This would allow for more easily achievable room scale variety and would result in less 'janky' use of modular assets.
- Some players express a desire to move more diagonally through levels.
 - This comes from wanting to more quickly get from one location to the other, more freedom of movement. That said, designers should be careful of this; the more free the movement, the less interesting the gameplay, and the more advantage Innocents have over Impostors.
 - Diagonal movement also doesn't feel good, usually, do to the viewing angle of the camera.
- Players prefer moving horizontally on the screen.
 - This may have to do with screen aspect ratios normally displaying images wider than they are tall.
- Avoid large spaces that are outside the playable space within the level.
 - This would include particularly thick walls, mostly. The client sees these as a visual eyesore.
 - Supermall used to be on the 'second floor' of the mall, with gaps in the hall to see the floor below, these gaps in the playspace felt like a chore to navigate.

Levels

Summary

This page provides documentation on *Impostors* levels. Here, you can see sketches, final designs, possible future designs, and common feedback pertaining to levels. The goal of the page is to build level design context so future designers understand level shortcomings and past decisions relating to level design.

- [Summary](#)
- [Macro Concepts](#)
- [Supermall \(10-15 Players\)](#)
 - [Sketches](#)
 - [Current \(Final\) Design](#)
 - [Layout](#)
 - [Design](#)
 - [Philosophy](#)
 - [Feedback](#)
 - [Various Other Designs](#)
- [Minimall \(5-10 Players\)](#)
 - [Sketches](#)
 - [Current \(Final\) Design](#)
 - [Layout](#)
 - [Design](#)
 - [Philosophy](#)
 - [Feedback](#)
- [Superstadium \(10-15 Players\)](#)
 - [Sketches](#)
 - [Current \(Final\) Design](#)
 - [Layout](#)
 - [Design](#)
 - [Philosophy](#)
 - [Feedback](#)
- [Future Level Concepts](#)

Macro Concepts

A list of things to keep in mind for any level design in *Impostors*.

- Maps made to fit 10-15 players, 'Super' maps, play best at 70m tall and 90m wide.
 - This shouldn't be strictly enforced, only suggested. Player should know that 4 players in a Super map won't be balanced but if they enjoy that gameplay, we should allow them to do so.
- Maps made to fit 5-10 players, 'Mini' maps, play best at 45m tall and 60m wide.
- Maps should have less rooms than the maximum number of expected players.
- Maps should be wider than they are tall or at least square.
 - This was discovered after the creation of Supermall.
- Diagonal walls add character but make movement challenging or frustrating.
 - Diagonal walls should be used in places where they can be aesthetic but don't have a major impact on player movement
 - A diagonally hallway won't work well but a diagonal corner in a room will.

Supermall (10-15 Players)

Supermall was the first level built for *Impostors* and took the brunt of the trial and error process. Because of this, it has been reworked numerous times to arrive at the current design. It is also far from ideal, even through iterations, we liked the layout too much to make huge changes later on, and we ended with a map that breaks a lot of the conventions that should be in place for future levels.

Sketches

When sketches first began for Supermall we didn't know how large the blocking volumes were (or that we should plan for them) we didn't know the camera angle, the max number of player intended, how many minigames there would be, etc.

For this reason, Supermall saw an endless number of small tweaks and iterations that could not have been foreseen, you will notice this between the *Final Sketch* image and the current Supermall Layout.

Note: dozens of sketches were created for this level, presenting all of them seemed impractical.



Pros

- Central, diagonal hallway

Cons

- Too Complex in southeast
- Central area gives too much information



Pros

- Clean, simple layout
- Figure 8 flow
- Back hallways can be interesting

Cons

- Probably too open/basic



Pros

- Central hallway

Cons

- Needs more vent groups
- Start area is too southern

Current (Final) Design

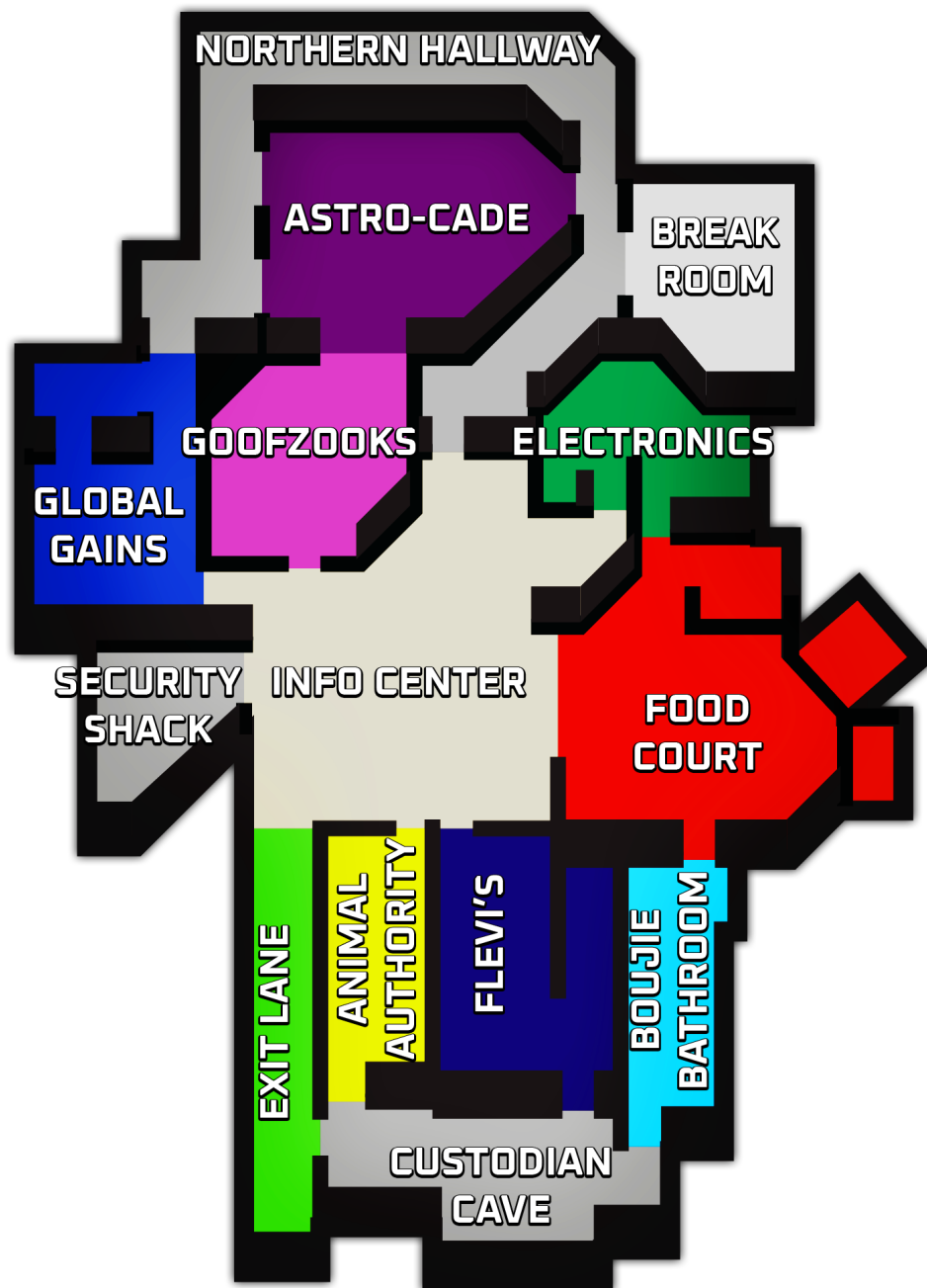
This is the latest version of Supermall, modified to player feedback and designer observations.

Layout



Changes

- Too many to count, really.
- Removed all internal dead space
- Removed many doors and hallways
- Simplified room shapes
- Accounted for blocking volumes
- Added walls with windows



Design Philosophy

This section covers some of the core concepts and goals for the design of Supermall.

- Figure 8 level flow.
 - Supermall has two primary movement loops: the south loop and the north loop.
 - These two loops overlap in Info Center and extend to the north and south-most points of the map.
- Room variety
 - Every space should feel fundamentally different than the others.
 - Implemented through scale, door placement, wall shape.
 - This is all considered before art further differentiates the spaces.
- Originally, most rooms had a main entrance and a less-used side entrance, this was mostly cut from the level but you can see this influence with Northern Hallway and Custodian Cave.

Feedback

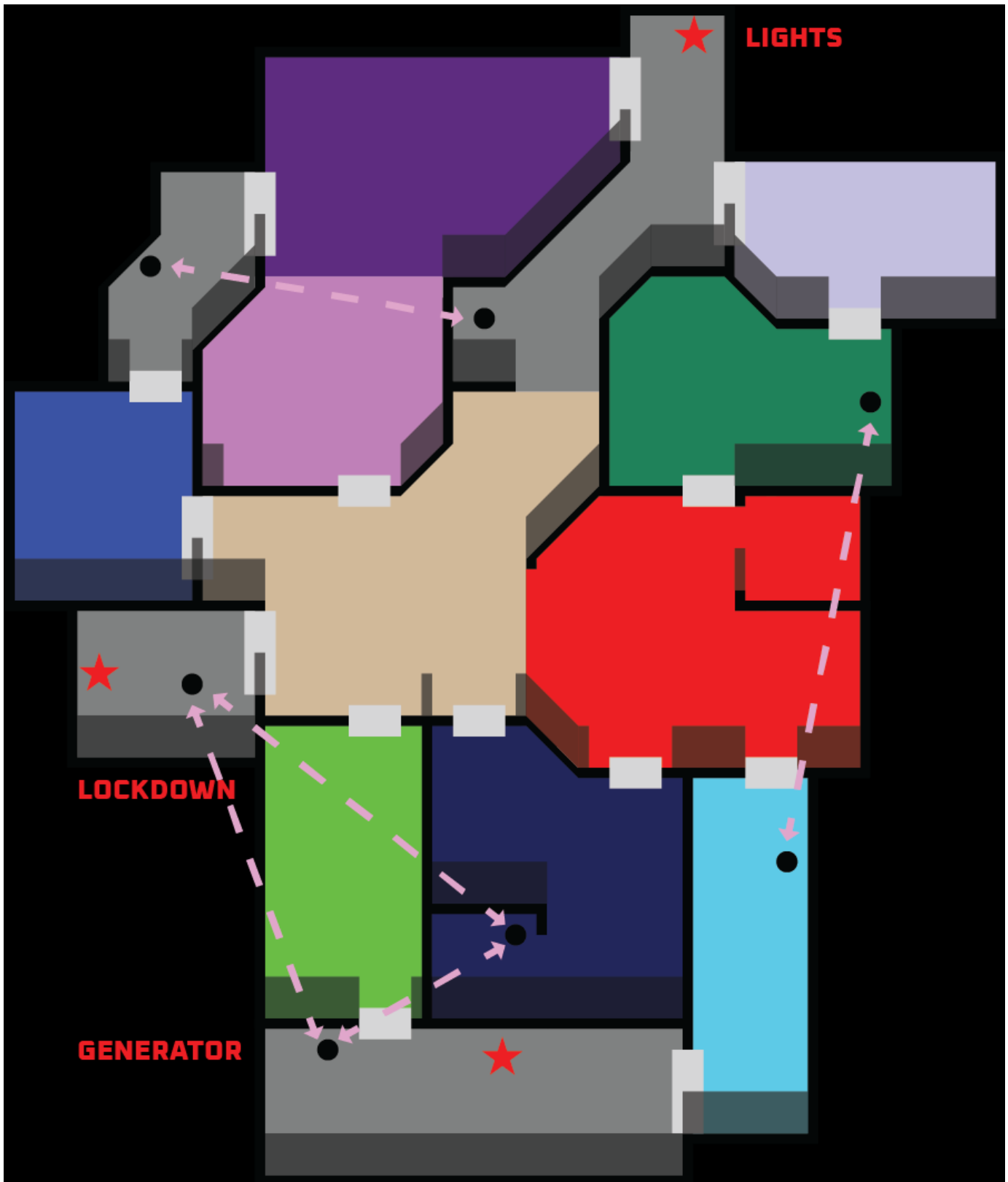
This section contains common feedback on Supermall from playtesting. It is intended to make designers aware of potential feedback, level shortcomings, or particular features that players really liked or disliked.

Note: This feedback only pertains to the current iteration of the level.

- Some players found the level too large
 - A minority wanted to see an even larger map, though.
- Most players disliked that the level was taller than it was wide.
- Players feel Security Shack is very dangerous - too dangerous to justify entering just to fix the Electrical Sabotage.
- Without a task in it, being seen in the locker room of Global Gains results in being seen as immediately sus, since there is only an Impostor hole there.
 - Same for the back room of Flevi's

Various Other Designs

The scale of Supermall often came into question. Supermall underwent numerous downscales between the Final Sketch version and the one currently seen in game. That said, further reductions were discussed, and that lead to V2, V3, and V4 versions of the level, presented below.



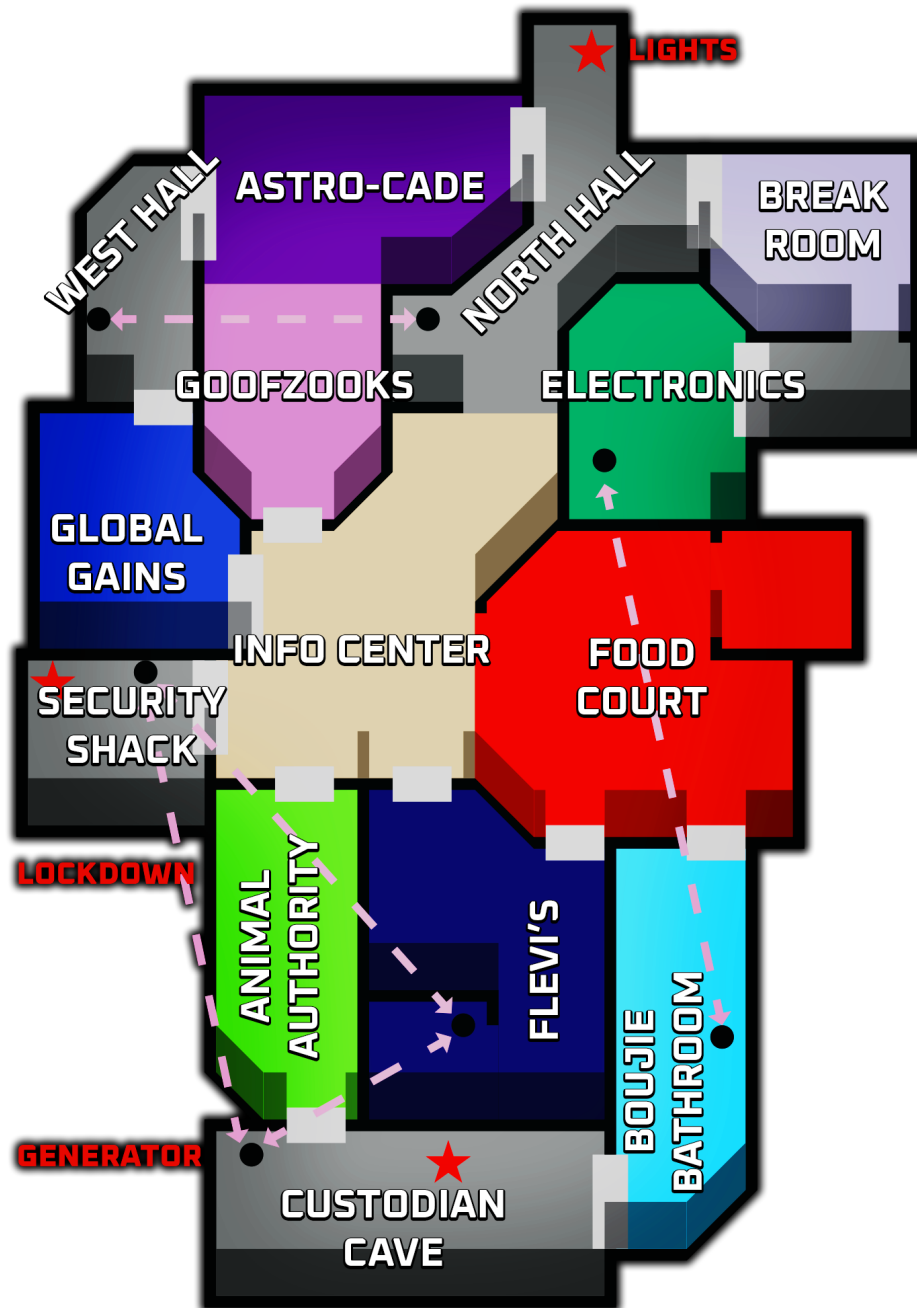
Supermall V2 was created to cut more space from the original level, making it about 25% smaller.

Changes Players Liked

- Electrical (Lights) in the north instead of Security.
- A door between Electronics and Break Room.
- No hallway above Astrocade.
- Many players felt the reduction of scale was a big positive.

Changes Players didn't like

- No door in the south end of Flevi's
 - This was common but I think it was because players were used to having a door there in the original level.
 - This made Flevi's a more balanced location for Impostors.
 - No more Exit Lane.
 - West Hall diagonal didn't feel good.
 - Security Shack felt pointless
-



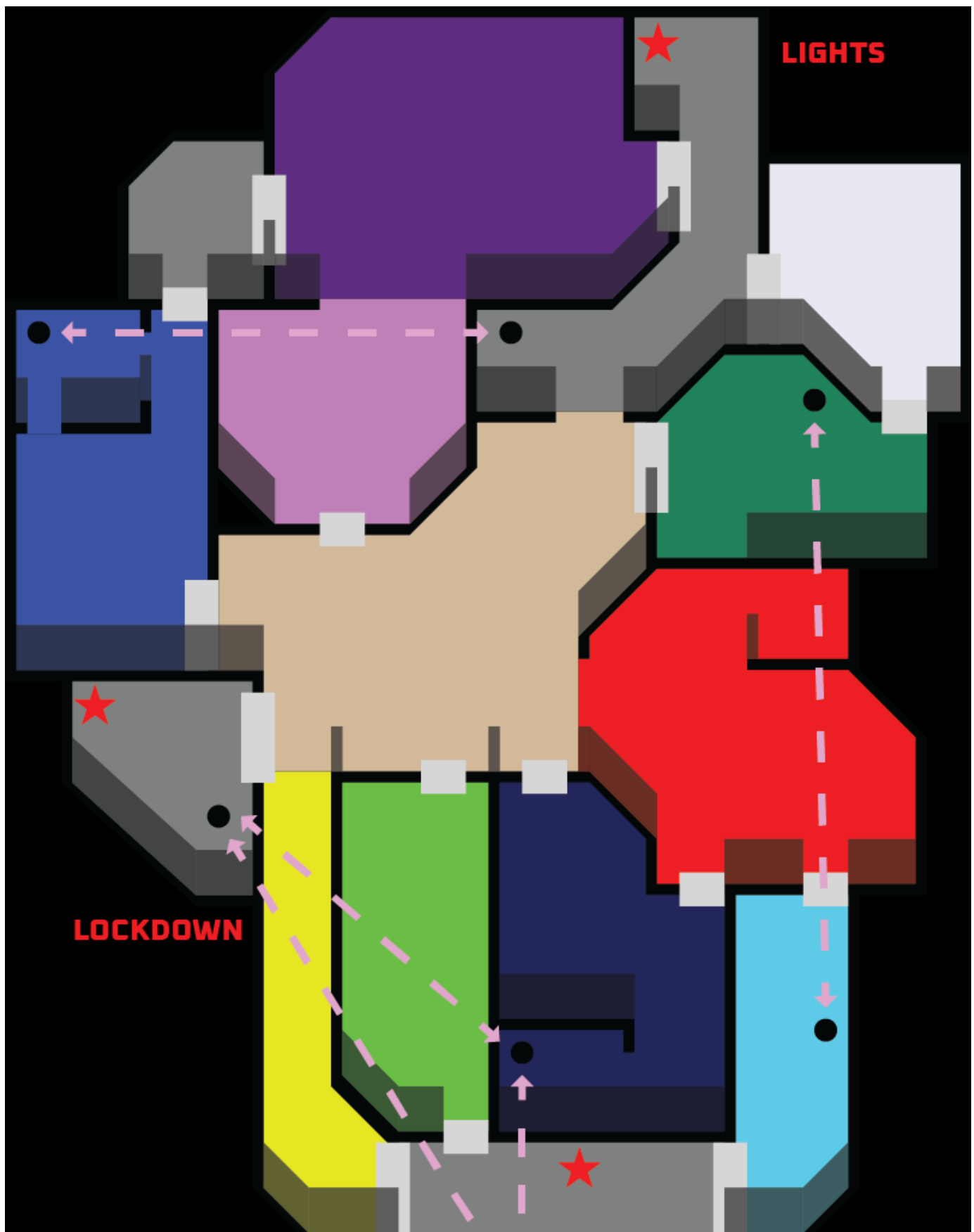
Supermall V3 was created to cut more space from V2, making it an additional 10% smaller about 60% the size of the original Supermall. Very little time was spent playing Supermall V3 before it was cut and the project went back to using the original Supermall as the primary level.

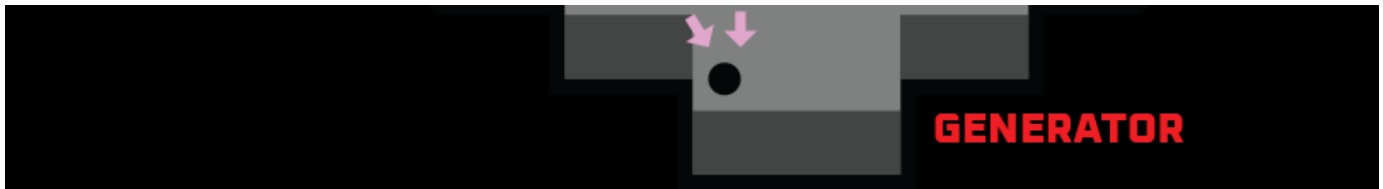
Changes Players Liked

- Info Center's scale felt better.
- Food Court felt less massive.

Changes Players didn't like

- The scale felt claustrophobic
- Electronics felt isolated (Originally it didn't have a door on the left, only the south)
- Players were disappointed to see that there wasn't enough space for many art assets which made the level feel uninteresting.





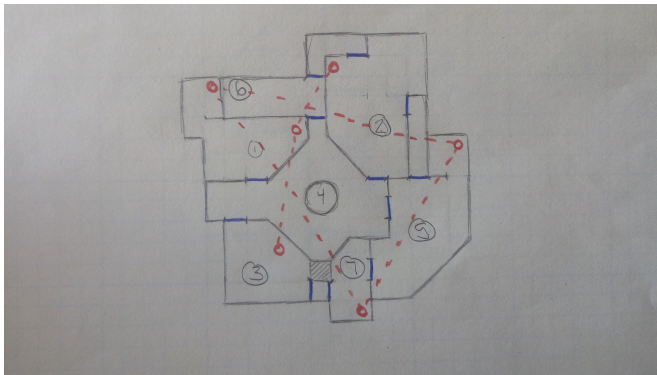
Supermall V4 was never made or tested. This is a concept sketch created with feedback in mind from V2 and V3. It is mostly a blend of the OG Supermall and Supermall V2, bringing many of the aspects people liked about V2 to a level more faithful to the original. Personally, I think this level has a lot of potential to be better than the original.

Minimall (5-10 Players)

Minimall can be seen as a variant of Supermall intended for people with a smaller group to play with. It contains many of the same features like windowed walls, stores, shelves, etc. but with less rooms and an overall smaller scale.

Note: More than a dozen sketches were created for this level, presenting all of them seemed impractical.

Sketches

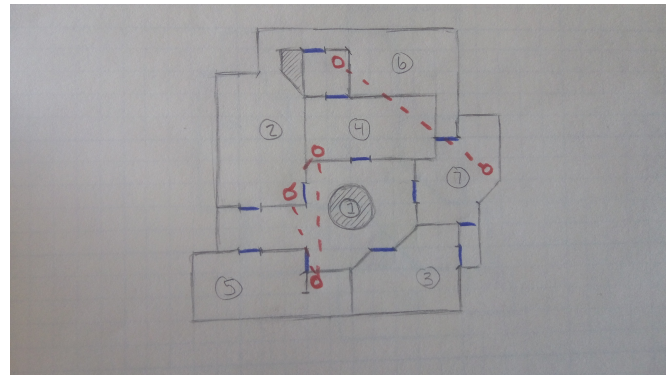


Pros

- Very striking layout

Cons

- Too vertical

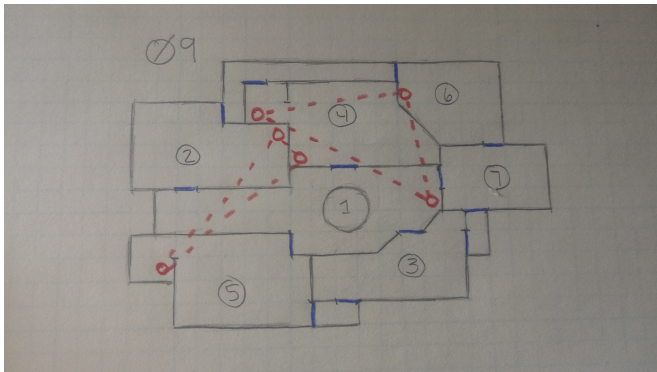


Pros

- Everything is close to the center

Cons

- Too vertical

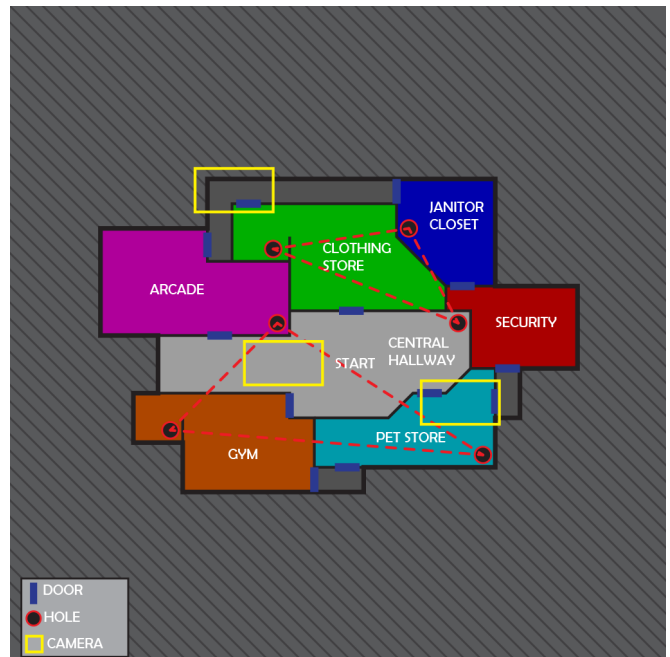


Pros

- Hole locations weren't the best

Cons

- Going through the dressing room to get to the top hall was awkward.
- A bit boring
- Security Shack felt pointless (7)



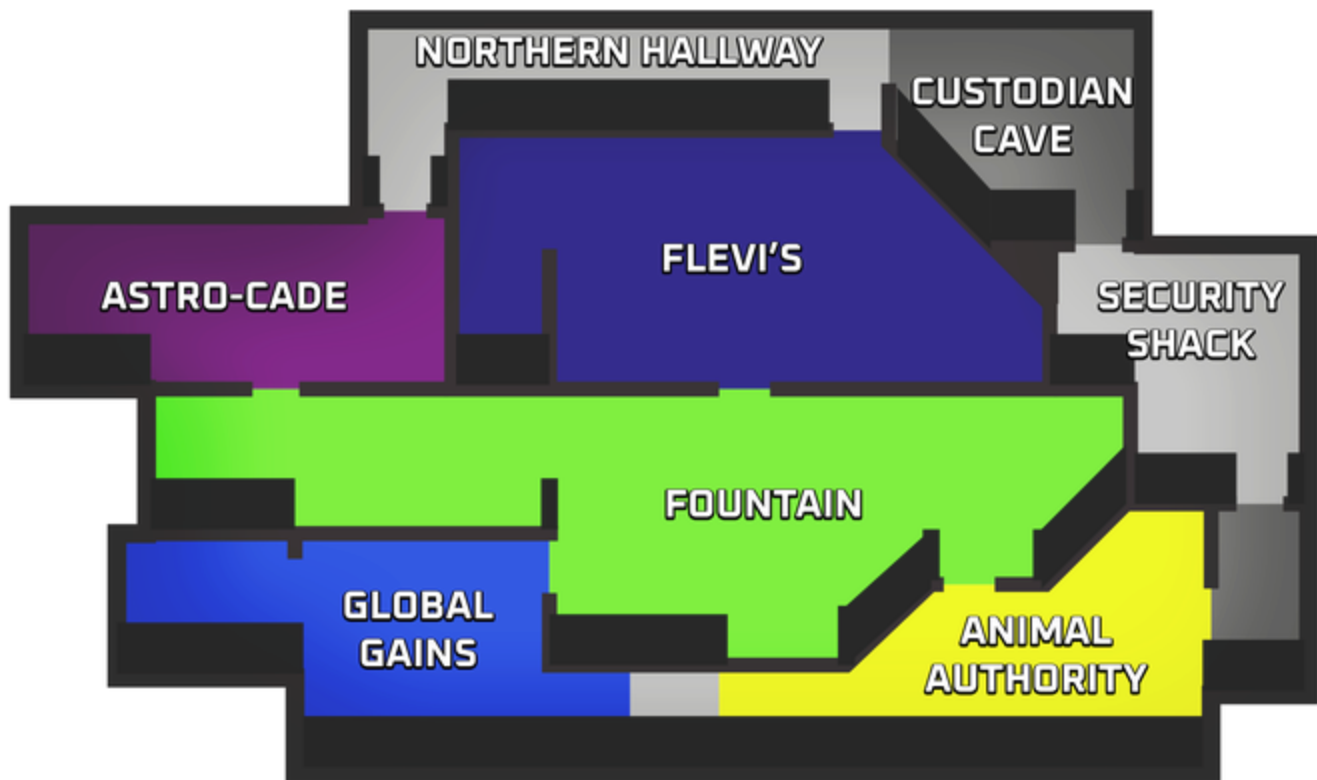
Changes

- Moved the dressing room south
- Places Southern Hall up between Global Gains and Animal Authority
- Made the level shorter overall
- Simplified room shapes

Current (Final) Design

This is the latest version of Minimall, modified to player feedback and designer observations.

Layout



Design Philosophy

This section covers some of the core concepts and goals for the design of Minimall.

- One large, exterior loop.

- This loop kept players on the outer edges of the level to move from place to place.
 - This allowed space between players in a small level by encouraging players be on the outer edge
- Player cut inward to get to tasks.
- This was intended to make it feel very different from Supermall.
- Larger rooms but with more sightline blockers within them.

Feedback

This section contains common feedback on Minimall from playtesting. It is intended to make designers aware of potential feedback, level shortcomings, or particular features that players really liked or disliked.

Note: This feedback only pertains to the current iteration of the level.

- This level is much harder for Impostors.
- Players want to more easily get to Security Shack
 - This would balance the level further in favor of the Innocents and Security Shack acts like a safe kill zone due to its isolation.
- The Southern hall feels too narrow.
- Flevi's feels too spacious.
- Doors don't feel useful
 - Likely because there are less doors as a result of less rooms.

Superstadium (10-15 Players)

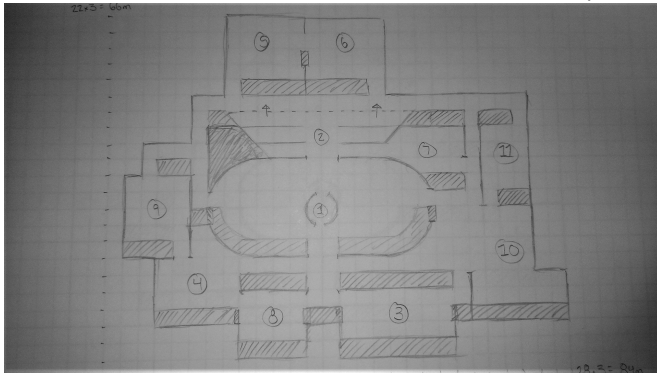
Superstadium was proposed as the next big level to be added to Impostors. It went through a design and greybox phase but eventually stopped production. Note: the files have been removed from the project in an attempt to keep the project clean but the source files are all still present.

Superstadium introduced new features such a bleachers that allowed players to access locations above other players from drop kills or sneaky escapes. The main feature of Superstadium was a central field area that the rest of the level encapsulated.

Sketches

A collection of the most liked initial sketches of the level. For this level, I had an understanding of the importance of the blocking volumes, and took them into account. I also changed the scale I was using, here, each square is 3m instead of 5. Since blocking volumes on walls are 300u wide, this allowed by sketch to be mathematically perfect. Later, this really helped level construction.

Note: More than a dozen sketches were created for this level, presenting all of them seemed impractical.

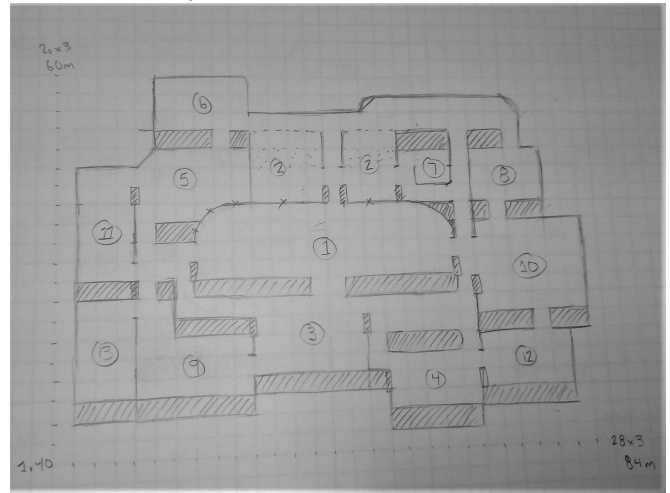


Pros

- Big central stadium area
- Easy to move from one side to the other

Cons

- Might be too vertical
- Halls above 8&3 are too narrow

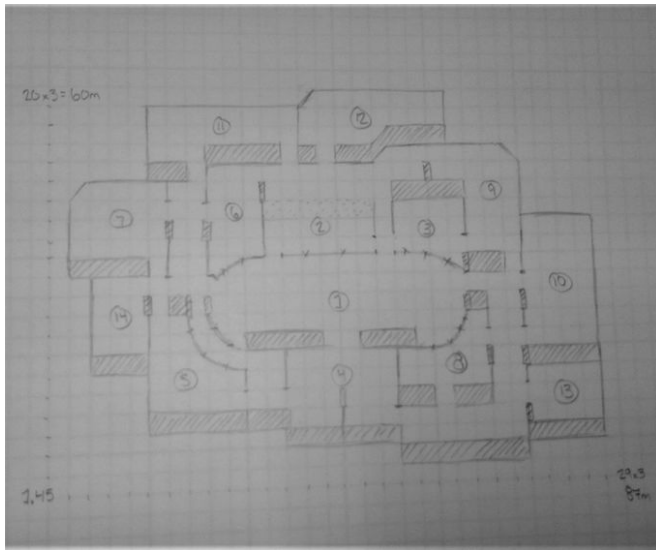


Pros

- Path through bleachers

Cons

- Odd stadium shape



Pros

- Most unique layout

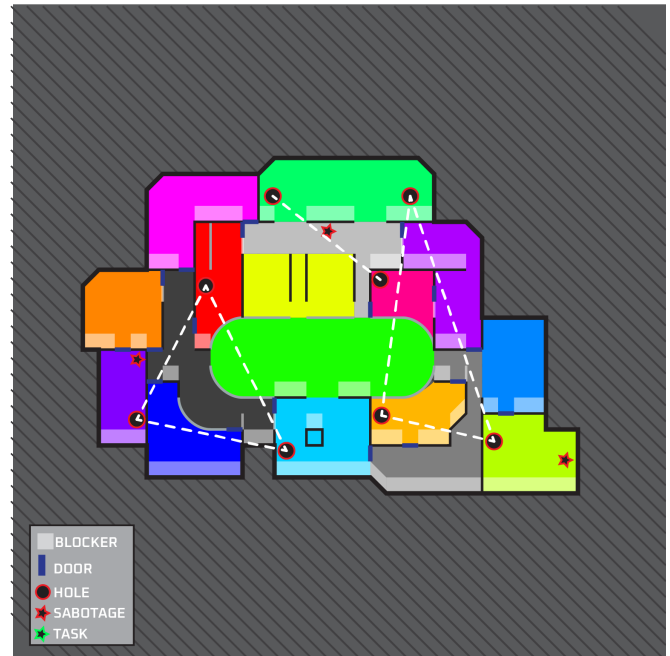
Cons

- Curved hallways may be difficult to navigate

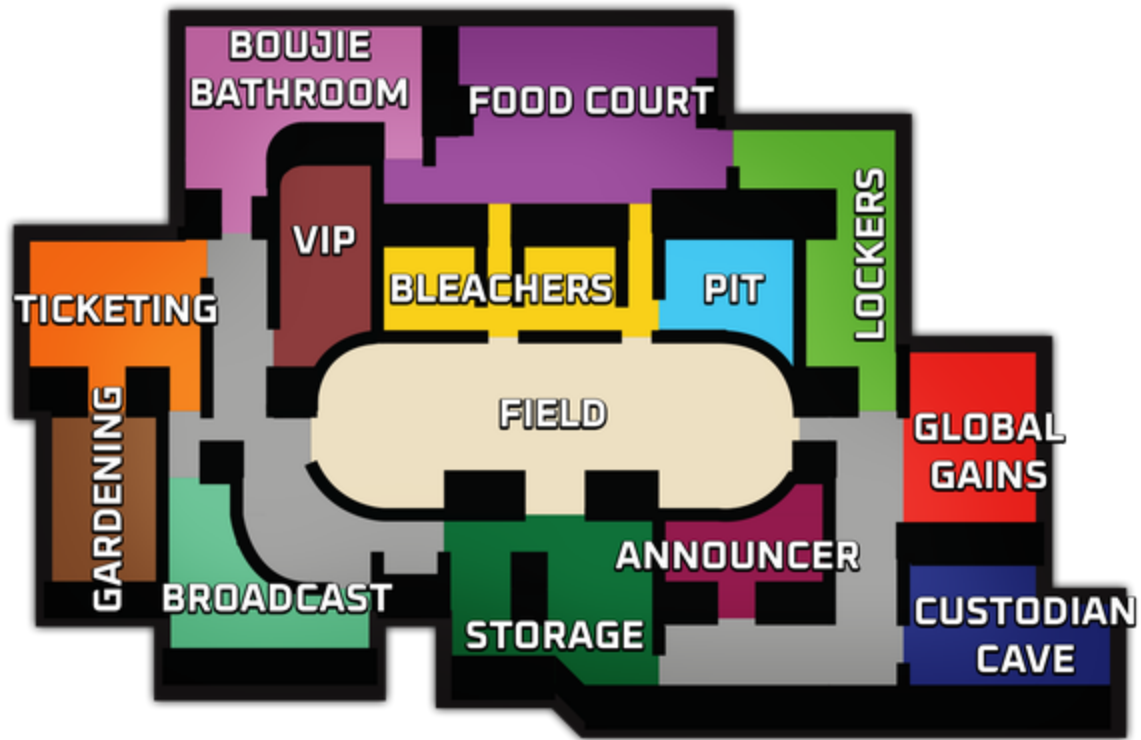
Current (Final) Design

This is the latest version of Superstadium, modified to player feedback and designer observations.

Layout



- Southeast hall was too narrow
- Northern area was too complex
- VIP, Announcer, and Pit were too busy
- Custodian Cave was hard to get to



Design Philosophy

This section covers some of the core concepts and goals for the design of Superstadium.

- Bleachers
 - Bleachers were an experiment in [verticality](#).
 - There wasn't anything under the bleachers but the bleachers allowed players on top to see into the Food Court while players in the Food Court were unable to see player on the Bleachers.
 - Players loved the way this changed the level's meta for information gathering.
 - Being on the bleachers would allow an innocent to witness a kill in Food Court but was also a convenient place for an Impostor to hide a body.
- Flow Loop
 - Players showed a preference for levels that had one distinct loop and the rooms extended from that.
 - Hallways all the way around lead to lots of dead space, so some hallways became rooms like Lockers or Boujie Bathroom.
- Stadium seating

- While on the field, a ring of stadium seats would appear over the other rooms to give players the sense they are in the center of a large stadium and the the space they are playing in is the backrooms often seen underneath the crowd in real stadiums intended as private space for the players and maintenance crew.
- Room scale
 - The overall level scale needed to be smaller than Supermall but support just as many players.
 - The answer for this was significantly more rooms, but the rooms were much smaller.
 - This divided the space, keeping players separate even if there were 15 in the game.
 - Smaller rooms meant players had more information about their immediate space.
 - This room scale better met the desires of the client, who wanted to be able to see the entirety of the rooms he was in.

Feedback

This section contains common feedback on Superstadium from playtesting. It is intended to make designers aware of potential feedback, level shortcomings, or particular features that players really liked or disliked.

Note: This feedback only pertains to the current iteration of the level.

- Overall, reception of this level was very positive, some players considered it their favorite map and were distressed when it stopped being worked on.
- Despite being smaller than Supermall, players felt as though this map were particularly large.
 - This is likely due to the increased number of rooms.
 - This may not be a bad thing, the smaller rooms were seen positively.
- Players like the more clear loop when compared to Supermall.
 - The ring of rooms that interconnect around the exterior of the field area is favorable.
- Players would love to see the arena area randomly change for different sports.
 - Sports suggested: Football, American Football, Hokey, and Tennis

Future Level Concepts

This section covers a few macro concepts for level layouts in the future.

- Verticality
 - It would be nice to take advantage of out 3D Environments
 - This would require different floors on top of each other, but could also contain levels that are split, with sections of the map being raised over others.
 - 'Gates' could be made with ledges that players can only cross in one direction, by drooping down, but are unable to go back the other direction. This would add a lot to player flow and has interesting implications for social deduction.
 - Having areas that are raised enough that players above and look down but players below cannot see up would make for interesting information gathering in social deduction gameplay.
 - Verticality was experimented with using the bleachers in Superstadium
 - Portals
 - This would allow innocents to cover large distances instantly, similar to an Impostor hole.
 - Open Layout
 - This concept plays with the idea that a bulk of the level is exterior and that rooms act as small, self-contained spaces within that open space.
 - This would make the level feel much different than Supermall, where movement was room to room.
 - This would feel very different from Superstadium where most movement was room to constricted hallway to room.
-

Materials

Summary

- [Summary](#)
- [Master Materials](#)
 - [Material Functions](#)

A simple list of Master Materials and Material Functions. This covers what each of these were created for.

Master Materials

Here is a brief explanation of each master material in the engine and what they are intended for. These materials are located in the `Impostors/Materials/MasterMaterials` folder for props and the `Impostors/Architecture/Materials/MasterMaterials` for architecture.

- All Master Materials have a Color Vector Parameter.
- All Master Material have Texture Parameters for Albedo, AORM(RMA), and Normal maps.
- `M_BaseDecalAlpha`: A material for creating decals
 - The Albedo map **must** have an alpha channel for this material to function properly.
 - Scalar parameters include: [MF_Adjuster_01](#) features and opacity adjustment.
 - Grout Mask uses [MF_TilingMask_01](#) features
 - Variety Mask allows the artist to change the transparency of the material so the same texture maps can be used several times with different masks to provide visual variety.
 - Example: All grunge decals use the same dirt material but the Variety mask changes the decal to look like streaks, splotches, or pools.
- `M_BaseEmissive`: A master for creating emissive materials that require texture definition.
 - Scalar parameters include: [MF_Adjuster_01](#) features and brightness adjustment.
 - Emissive Map allows artist to mask which areas of a material receive emissive and to what extent.
- `M_BaseGlass`: A master for creating glass and other transparent materials.
 - Base glass does not use texture maps.
 - Scalar Parameters include adjustments for Fresnel attributes and reflectivity.
- `M_BaseHeight`: A master made for the [Blind Pack minigame](#).
 - This material should not be used outside of the Blind Pack minigame.
 - Scalar parameters include: [MF_Adjuster_01](#) features and height/bump adjustment.
- `M_Baselight`: A master for creating emissive materials that don't require texture definition.
 - Scalar parameters include: [MF_Adjuster_01](#) features and brightness adjustment.
 - This master does not have control for AORM or normal maps
 - This master has a mask to control which parts of the light receive emissive.
- `M_BaseMaterial`: A master for creating any simple, opaque material
 - This is by far the most common Master Material to use.
 - Scalar parameters include: [MF_Adjuster_01](#) features.
- `M_BlinkingLight`: A master for creating emissive materials that require inconsistent brightness.
 - Scalar Parameters include: Brightness, flicker time controls, and flicker clamp controls.
 - This master has a mask to control which parts of the light receive emissive.
- `M_ClearPlastic`: A master for creating clear and other transparent materials.
 - Similar to [M_BaseGlass](#).
 - Does not have reflectivity controls.
 - Adds albedo texture parameter.
- `M_MaskedMaterial`: A master for creating materials that require an opacity mask that aren't decals.
 - Scalar parameters include: [MF_Adjuster_01](#) features.
 - This master has a mask to control which parts of the texture are transparent.
- `M_Mirror`: A master for highly reflective surfaces.
- `M_NGMI`: A master exclusively created to bring shifting colors to the NGMI machine emissive.
- `M_Plants`: A master for creating plants.
 - This master has a mask to control which parts of the texture are transparent.
 - This material adds a subsurface color and secondary color to better mimic plant appearance.
- `M_Unlit_Base`: A master for creating materials using the `unlit` shader model.
 - Only offers color control.
- `M_Water`: A master for creating very simple water.
 - Uses two opposing normal maps to create the appearance of waves.
 - Scalar parameters include: Opacity and wave speed adjustments.
- `M_BaseWall_01`: Simple wall material with basic shader properties.
 - You can also apply a color mask if you want the adjusted color vector parameter to only impact part of the material.
 - This material allows you to control the rotation, scale, and translation of the texture maps to ensure they line up nicely with the environment.
- `M_TilingFloor_01`: A floor that tiles according to world position.
 - This material ensures that floor materials always tile no matter the UVs or orientation of modular floor assets.
 - You can also apply a color mask if you want the adjusted color vector parameter to only impact part of the material.
 - This material allows you to control the rotation, scale, and translation of the texture maps to ensure they line up nicely with the environment.

Material Functions

Here is a brief explanation of each material function in the engine and what they are intended for. These materials are located in the `Impostors/Materials/MasterMaterials` folder for props and the `Impostors/Architecture/Materials/MasterMaterials` for architecture.

- `MF_Adjuster_01`: A function that gives control of texture map transforms to the material.
 - This mask scales, rotates, and transforms any texture map.
 - This function is used for almost every Master Material in the game to grant real-time control of material appearance.
 - `MF_TilingMask_01`: A function that adds a texture to tiling materials.
 - A variation of `MF_Adjuster_01` that applies the transforms to a texture parameter before output.
 - This mask scales, rotates, and transforms a texture map.
 - It is currently used in-game for decals to change their shape with a transparency mask.
 - `MF_WorldTiling_01`: a function that tiles a material based on world position.
 - This ensures seamless material tiling that works independent of UVs.
 - This only works for X and Y axis - faces along the z-axis like walls will appear stretched.
 - This works best for tiling floor textures.
-

Minigame Art Asset Creation and Implementation

Summary

Minigame assets are unique in that they are presented to the player close to the camera. This presents a few challenges but mostly allows artists to create detailed artwork in a controlled environment. This page covers how art has been created for the *Impostors* minigame environments.

- [Summary](#)
- [Constraints](#)
 - [Mesh Constraints](#)
 - [Material/Texture Constraints](#)
- [Mesh Setup](#)
 - [Importing](#)
- [Additional Notes](#)

Note: Minigame assets are usually high-fidelity representations of in-world assets. This was done to ensure the visual detail of the levels are consistent and optimized. Level props are much lower in detail and don't work well in minigames, thus a separate assets is created for minigames.

Constraints

Most artwork for the *Impostors* levels was created with rigorous constraints to ensure they are performant. Covering these constraints to begin with help explain many of the different art decisions made so it makes sense to discuss them first and foremost.

Constraints are less rigorous for minigame assets. Since the player is closer to these assets, visual fidelity takes priority.

Mesh Constraints

- Poly count should not exceed 5,000 before engine triangulation.
- Be sure to remove backside faces - the player will never see behind a mesh.
- UV Texel density is roughly 256px/m.
- Bevels should contain two to three loops.
 - At this distance, these loops are needed to create nice, round bevels.
 - Round bevel increase the 'fun' of the meshes.

Material/Texture Constraints

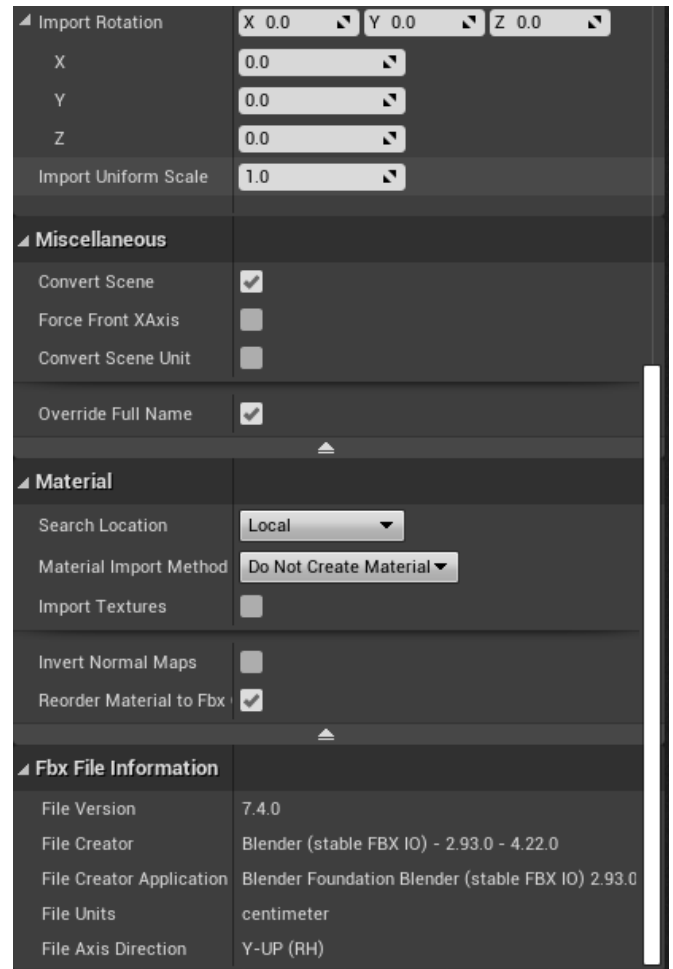
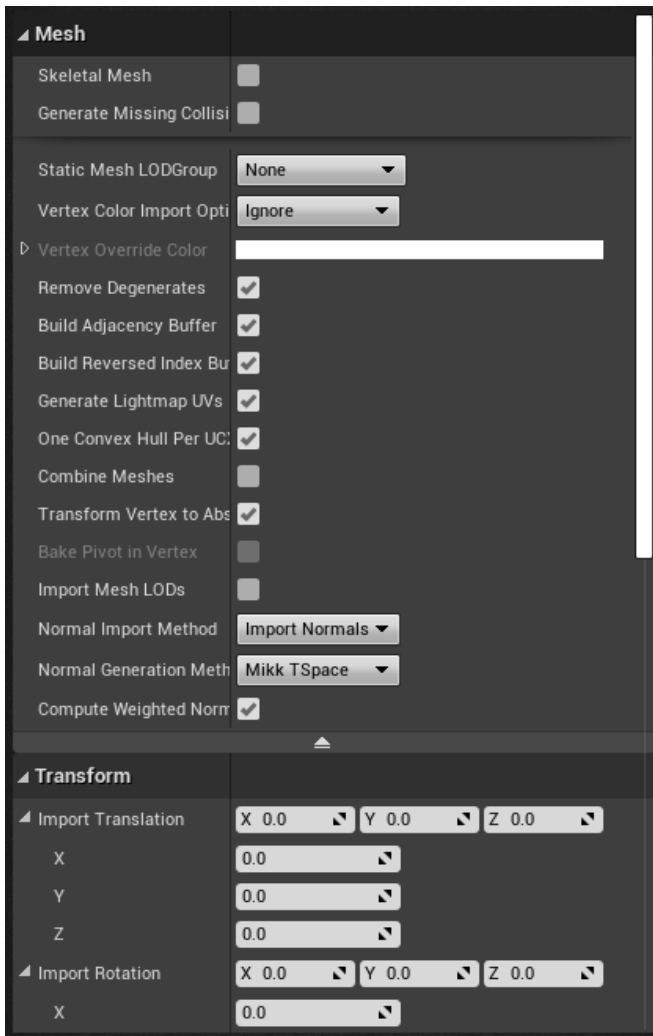
- Albedo/Defuse textures that feature iconography or script should be 2048x2048 to ensure legibility.
- Albedo/Defuse textures that are simpler (plastic, metal, etc.) should never exceed 512x512.
- Pack Metallic, AO, and Roughness maps.
 - This isn't only suggested but required if you want PBR maps to work correctly with existing master materials.
 - Red = Metallic
 - Green = Roughness
 - Blue = Ambient Occlusion
- Most materials should be simple, tiling textures.
 - Texture variation across objects should be achieved through additional materials slots on meshes or by applying decals to meshes.
- Do not make individual maps for each and every object.
 - Baking texture maps to individual objects is not consistent with the current setup or art style.
 - Creating a trim sheet that all meshes reference on a per-minigame basis seems to work best.

Mesh Setup

Once a mesh is created and ready to import, there are a few steps to perform in-engine to ensure the best appearing and most performant mesh possible.

Importing

- Minigame meshes should be imported to the `Impostors/Objects/Tasks/[RelevantMinigame]` folder.
- Refer to the images below for settings when importing meshes.



- Do not import meshes with collision.
 - Few meshes will actually require collision
 - Removing collision is tedious
 - Extraneous collision is a heavy weight on performance.
- Minigame collision often works best with invisible planes placed in the Blueprint viewport.
 - Buttons or other objects the player interacts with will need collision
 - Collision shouldn't be more complex than a simple collision hull created in-engine.
 - Collisions should be oversized to ensure easy intractability.

Additional Notes

Need more information? If you need to know more about materials, you can find that information [here](#). If you need more information on art style or material setup, you can find that information [here](#).

World Art Asset Creation and Implementation

Summary

World assets are unique in that they are presented to the player at a large distance from the camera. This presents a few challenges but mostly allows artists to create highly optimized artwork with little loss of visual fidelity. This page covers how art has been created for the *Impostors* level environments.

- [Summary](#)
- [Constraints](#)
 - [Mesh Constraints](#)
 - [Material/Texture Constraints](#)
- [Art Style](#)
- [Mesh Setup](#)
 - [Importing](#)
 - [Collision](#)
 - [LoS Shadow Setup](#)
- [Material Setup](#)
 - [Importing and Setup](#)
- [World Art Asset Decisions](#)
- [Additional Notes](#)

Constraints

Most artwork for the *Impostors* levels was created with rigorous constraints to ensure they are performant. Covering these constraints to begin with help explain many of the different art decisions made so it makes sense to discuss them first and foremost.

Mesh Constraints

- Poly count should not exceed 3,000 before engine triangulation.
 - Meshes smaller than the player shouldn't exceed 500.
 - Meshes up to twice the player size shouldn't exceed 1000.
 - Only particularly large meshes (Like NGMI machines) should exceed 2000.
- Be sure to remove underside faces - the player will never see under a mesh.
- UV Texel density is roughly 128px/m.
- Bevels shouldn't exceed single loops.
 - Adding more than one loop to a bevel isn't noticeable in-game but adds significant polygon count.

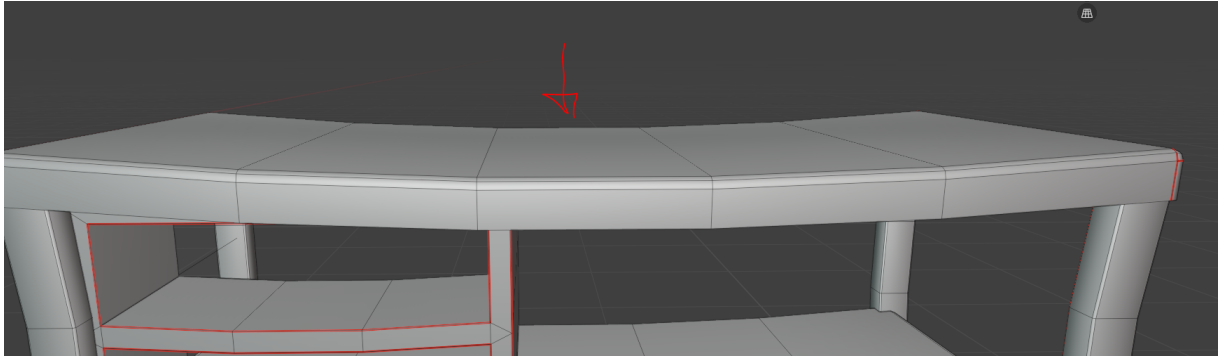
Material/Texture Constraints

- Albedo/Defuse textures that feature iconography or script should be 1024x1024 to ensure legibility.
- Albedo/Defuse textures on architecture should be 1024x1024 to ensure no fuzziness at scale.
- Albedo/Defuse textures that are simpler (plastic, metal, etc.) should never exceed 512x512 and, in most cases, can be much smaller.
- Other texture maps, like Roughness, Metallic, Ambient Occlusion, Masks, Emissive, etc. can usually be only 256x256 without noticeable impact on visuals.
- Pack Metallic, AO, and Roughness maps.
 - This isn't only suggested but required if you want PBR maps to work correctly with existing master materials.
 - Red = Metallic
 - Green = Roughness
 - Blue = Ambient Occlusion
- Most materials should be simple, tiling textures.
 - Texture variation across objects should be achieved through additional materials slots on meshes or by applying decals to meshes.
- Do not make individual maps for each and every object.
 - Baking texture maps to individual objects is not consistent with the current setup or art style.

Art Style

A few tips and tricks to help create artwork that is consistent with the current artwork.

- Avoid straight lines.
 - Most world meshes in *Impostors* have curvature to them, making them concave from their bounding box.
 - This increases the 'fun' of the world meshes.
 - This can be done by adding a central edge loop to a surface, moving it inward, then beveling that edgeloop outward to create nice curvature.



- Add 'slotchy' material variation
 - This helps create subtle visual complexity in the world, making it feel more lived in and interesting to look at.
 - For each material, use Photoshop's **Render Clouds** feature on a new layer.
 - **Posterize** those clouds to roughly 4 levels so it appears like this:



- Use blend modes to make this appear subtly on the Albedo/Defuse map of the textures before exporting the texture into Unreal.
- Minimize use of normal maps.
 - The client does not like the appearance of normal maps.
- Never make textures fully rough.
 - The client wants a Fortnite-like plastic look to the in-game materials - he wants to see some kind of highlight on all objects.
- Round corners when it makes sense to do so.

Mesh Setup

Once a mesh is created and ready to import, there are a few steps to perform in-engine to ensure the best appearing and most performant mesh possible.

Importing

- Architecture meshes should be imported to the `Impostors/Architecture/` folder.
 - For walls use `Impostors/Architecture/Walls/[MAPNAME]`
 - For floors use `Impostors/Architecture/Floors/[MAPNAME]`
 - Note: merged walls and floors for those levels are also stored here.
- Prop meshes should be imported to the `Impostors/Props/[MAPNAME]`
 - The `Impostors/Props/Generic` folder is intended for props that would work in any level theme.
- Refer to the images below for settings when importing meshes.

Mesh	
Skeletal Mesh	<input type="checkbox"/>
Generate Missing Collisi	<input type="checkbox"/>
Static Mesh LODGroup	None
Vertex Color Import Opti	Ignore
Vertex Override Color	
Remove Degenerates	<input checked="" type="checkbox"/>
Build Adjacency Buffer	<input checked="" type="checkbox"/>
Build Reversed Index Bu	<input checked="" type="checkbox"/>
Generate Lightmap UVs	<input checked="" type="checkbox"/>
One Convex Hull Per UC	<input checked="" type="checkbox"/>
Combine Meshes	<input type="checkbox"/>
Transform Vertex to Abs	<input checked="" type="checkbox"/>
Bake Pivot in Vertex	<input type="checkbox"/>
Import Mesh LODs	<input type="checkbox"/>
Normal Import Method	Import Normals
Normal Generation Meth	Mikk TSpace
Compute Weighted Norm	<input checked="" type="checkbox"/>
Transform	
Import Translation	X 0.0 Y 0.0 Z 0.0
X	0.0
Y	0.0
Z	0.0
Import Rotation	X 0.0 Y 0.0 Z 0.0
X	0.0

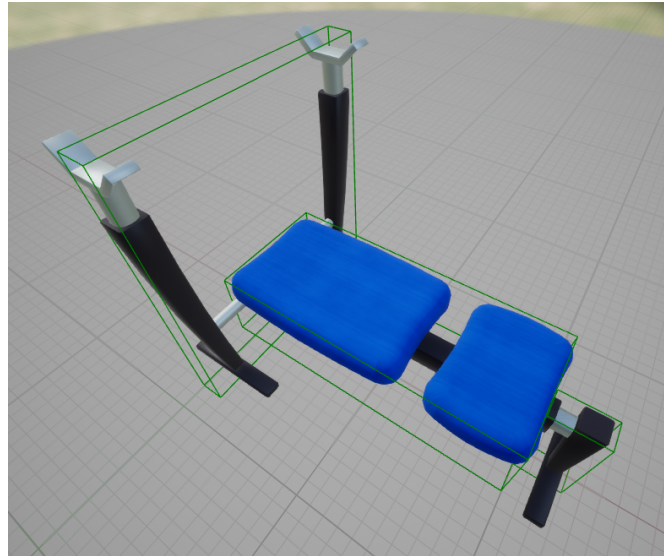
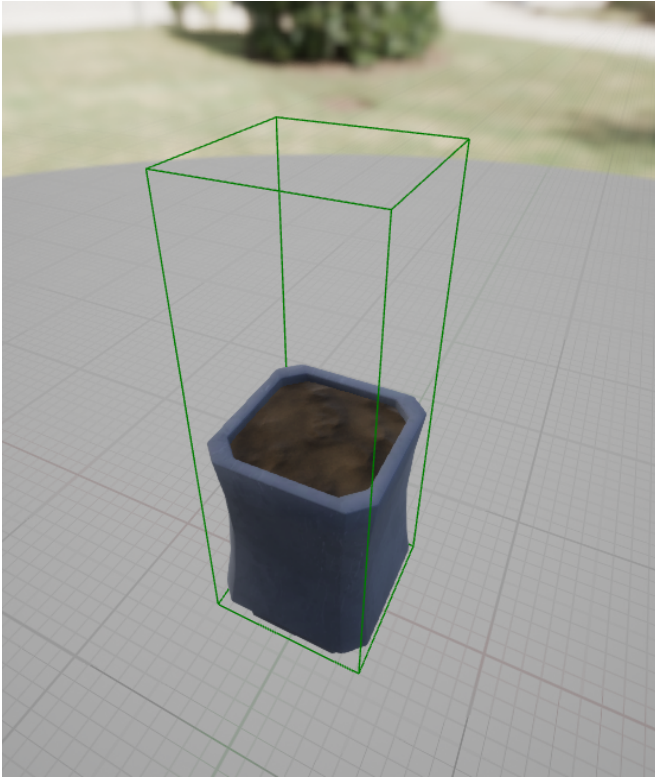
Import Rotation	X 0.0 Y 0.0 Z 0.0
X	0.0
Y	0.0
Z	0.0
Import Uniform Scale	1.0
Miscellaneous	
Convert Scene	<input checked="" type="checkbox"/>
Force Front XAxis	<input type="checkbox"/>
Convert Scene Unit	<input type="checkbox"/>
Override Full Name	<input checked="" type="checkbox"/>
Material	
Search Location	Local
Material Import Method	Do Not Create Material
Import Textures	<input type="checkbox"/>
Invert Normal Maps	<input type="checkbox"/>
Reorder Material to Fbx	<input checked="" type="checkbox"/>
Fbx File Information	
File Version	7.4.0
File Creator	Blender (stable FBX IO) - 2.93.0 - 4.22.0
File Creator Application	Blender Foundation Blender (stable FBX IO) 2.93.0
File Units	centimeter
File Axis Direction	Y-UP (RH)

- Do not import meshes with collision.
 - Few meshes will actually require collision
 - Removing collision is tedious
 - Extraneous collision is a heavy weight on performance.

Collision

After the mesh is imported, if necessary, add collision. This should only be done for architecture, large meshes, or meshes that you want to block the player otherwise (like trash cans or stanchions.)

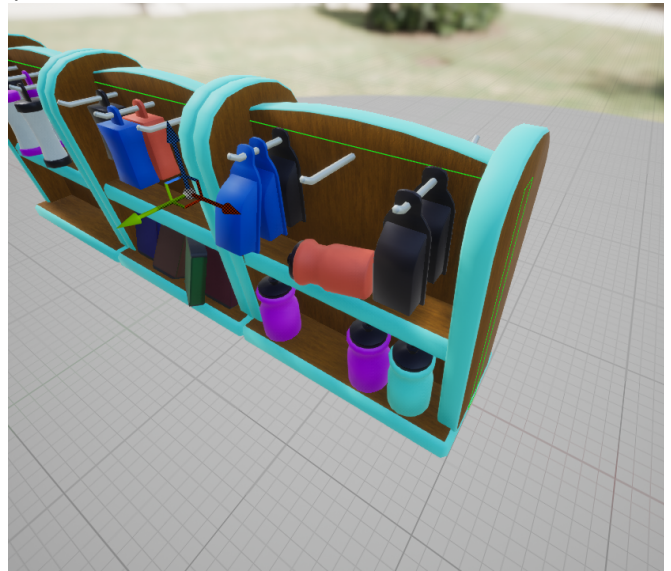
- Collisions impact [LoS Shadows](#), keep this in mind.
 - This isn't by default, unless you want shadows to cast from the object, don't worry about this.
- Collisions should be derived from simple box collisions and be perfectly perpendicular to the ground.
 - Players have a steep climb height, anything that isn't vertical could potentially allow the player to climb on top of it.
 - Complex collision isn't noticeable in-game.
 - Having one box collision hull is the most performant collision.
 - Using two or three is acceptable for more complex objects.



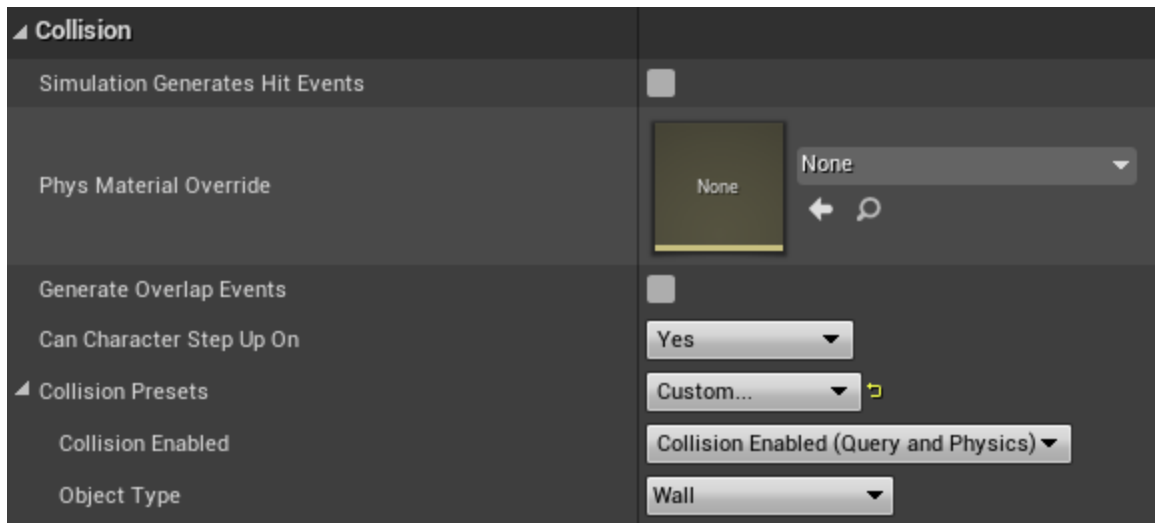
LoS Shadow Setup

LoS shadows are an important mechanic in a social deduction game. They break up spaces, allow players to hide, and ensure players cannot see other players through walls.

- LoS shadows derive their projection from object collision.
 - The shadows will grab the edges of the collision and project them. Note comparison below.
 - This will take a lot of iteration to get looking correct.
 - The collision must be a simple shape for this to work properly.



- Most meshes should not cast shadows in this manner.
 - Reserve shadows for objects the player could reasonably expect to block their vision, usually taller objects.
- Meshes will not generate a LoS shadow by default.
 - To get shadows, look in the meshes `Details` panel, under `Collision`.
 - Set the `Collision Preset` to `Custom`.
 - Set the `Object Type` to `Wall`. This enables shadows on the mesh.



Material Setup

Once a set of textures are created and ready to import, there are a few steps to perform in-engine to ensure the best appearing and most performant material possible.

Importing and Setup

- Import textures into the `Impostors/Materials/[MAPNAME]/Textures` for props or `Impostors/Architecture/Textures/[MAPNAME]` for architecture
 - Since Supermall and Minimall use the same assets, there is only one folder for both.
 - the `Impostors/Materials/GeneralEnvironment/Textures` folder contains textures that can be used regardless of level theme such as fabrics, plastics, and metals.
- After import, double-check all texture settings for each texture in their *Details* panel.
 - When set correctly, these option improve appearance and performance.

Once textures are imported, you are ready to set up the materials. *Impostors* uses almost exclusively Material Instances to texture the game environments which makes for a unique setup.

- Choose the relevant [Master Material](#), right-click, and choose to `Create Material Instance`.
- Move the Material instance to the `Impostors/Materials/[MAPNAME]` folder for props or `Impostors/Architecture/Materials/[MAPNAME]` for architecture.
- Setup the Material Instance through the parameters
 - Plug in relevant texture maps
 - Adjust the color
 - Note there are numerous scalar parameters.
 - The names are self explanatory
 - These parameters allow you to rotate, scale, and translate the texture maps in most cases
 - Depending on the [Master Material](#), these can also adjust metallic, roughness, opacity, etc.
 - Most materials won't need to use these.
- Rename the material instance according to naming conventions.

World Art Asset Decisions

This section explains various choices made during the art creation process for *Impostors* so later artists can avoid the trial and error process seen early on while creating art for *Impostors* levels.

- Using tiling textures and/or trim sheets.
 - Since the camera is so far from objects, creating highly-detailed textures only adds visual noise.
 - The client found these more realistic/detailed textures unpleasant.
 - Since there was a single artist on the team for environment artwork, it was inefficient to create unique materials for every object.
 - Without art direction, artists had to rely on an iterative process to achieve good-looking artwork. Recreating/repainting /retexturing objects with each iteration was inefficient.
- Material Instances
 - Material Instances are cheaper to run than full materials.
 - Material instances allow for quick and easy iteration.
 - It is faster to swap a Albedo map parameter in an instance than hook up a new one.
 - Seeing changes in run-time allowed for easy and accurate color adjustments without constant compiles and saves.
 - Material Instances allow for changes in run-time.