Tycoon Template v1.5

by Odium Studios 2025

Beginner Warning: This template is designed for individuals with a moderate to strong grasp of Unreal Engine Blueprints. However, this document will try to cover all the aspects required for a beginner to learn from, and potentially grasp the design. Though the document is long, it should cover the fundamentals of using this template to design your own theme park/zoo game.

Table Of Contents

Introduction:

Welcome to the Tycoon Template. This template is a large template designed to help with productions for games in which the player designs Zoos, Theme Parks, Circuses, or anything in which visitors will come and pay money. It can be modified to achieve other types of games as well, with minor variations. It is designed currently to be flexible, but slightly more aimed towards a zoo type atmosphere than a theme park with some of the currently available structures and "attractions". With modifications to the "attraction" type classes you can easily achieve a theme park outcome.

Along with the below features, this Template also contains 160+ additional UI buttons, icons, and widget boxes

Features:

While sizable and a good starting point for these types of games, Tycoon Template does not cover every single possibility for a Park Management game at this point. In future updates, and with community requests, it is hoped to upgrade this to a more fully-inclusive asset. As such, this template does feature a fairly large list of functions, features and capabilities worthy of getting your own game up and running in a short time. Most of the options here are based within Data Tables, and Enumerators, allowing for you to quickly add new options without changing any code. There are a few however which will require a little bit of code, which will be explained.

Current Features:

Visitor Features:

- Visitor Spawning and Despawning
- Visitor Arrival times and Intervals
- Functional AI system for visitors to fulfill their needs.
- Spouse and children spawning
- AI system for Spouse and Children to follow the initiating visitor to attractions/shops/etc
- Visitor payments account for a spouse and number of children in family
- Visitors purchase set available products and pay attraction entrance fees
- Visitor park opinions change depending on needs and available park structures and distance to them
- Visitor Name and Stat "information tag" accessible to find out their opinions/needs at that time
- **UPDATE V1.1**: Visitors now rest at benches and have a new stat for energy (not shown in the visitor name tag but can be added easily)

- **UPDATE V1.1**: Visitors now disappear when entering buildings for purchasing products to prevent pile-ups in front of buildings
- **UPDATE V1.1**: Visitors prefer to follow paths instead of walking on platforms, but will walk on platform areas if no path exists or they require to cross it to get to a location.
- **UPDATE V1.2**: Visitors have a markup threshold, which prevents them from purchasing overpriced products based on the value that is paid to purchase the product for sale.
- **UPDATE V1.2**: Visitors are now dirty. They will throw garbage on the ground if garbage cans are too filled, or if no garbage can is near.
- **UPDATE V1.2**: Staff can be hired, fired, given raises, and extra training.
- **UPDATE V1.2**: Animations updated to use ABPs instead
- **UPDATE V1.2**: Visitors can carry objects around the park with them.
- **UPDATE V1.3**: Staff statistics are tracked now.
- **UPDATE V1.4**: Visitors now follow queues for outdoor food kiosks (Canteens). Can be made to have any building use a queue if wanted
- **UPDATE V1.5**: Attractions can now have queues. Visitors stay closer to buildings and paths. Buildings can be renamed and customized. Visitors can be banned. Buildings have warning displays when the building is closed. Vendors can be assigned to buildings, reducing visitor shop times and increasing satisfaction. Entertainers added, increasing visitor happiness in a radius near them.

Building Features:

- Shop types and items sold list, all customizable by the player in runtime
- Item sale prices and choice to stock or not stock products for sale
- Building placement on “platform” surfaces, OR with slight modification (1 minute time to change) described in documentation, can be placed on terrain, with a slope detection
- Overlap prevention for specific objects, OR allowed to have “special” objects which may be placed within, overlapping, or crossing through others such as fences.
- Z offset and rotation abilities for structure placements
- Building and structure target locations for visitor locations built in
- Attraction pricing customizable and duration the visitor will stay at that location
- Building can be toggled with power or not
- New buildings and objects are easily added to the correct category for the build list
- **UPDATE V1.1**: Buildings and other objects can now be placed using the “remove snap” option by holding Ctl to allow for fine placements.
- **UPDATE V1.1**: Paths are now incorporated into the system. They operate in a grid format like the rest of the objects, though curved paths are added if you want.

- **UPDATE V1.1**: Buildings can now be quickly changed to use a different trace method for building/object placement. This makes it quick to find the right method for each situation by just changing an enum in the data table, removing the need for a number of tags.
- **UPDATE V1.1**: No need for custom (larger) temp meshes anymore for small objects
- **UPDATE V1.2**: Custom Building Data Table for use for adding all the buildings you want so updates will not overwrite your existing files OR to use as special Data Files in game.
- **UPDATE V1.3**: Statistics system set up for most common attributes tracked.
- **UPDATE V1.3**: Quick find employees now available.
- **UPDATE V1.3**: Visitors purchases are now tracked and displayed for each building.
- **UPDATE V1.4**: Buildings can now be set up to use a "queue system"
- **UPDATE V1.5**: Buildings can be renamed, can be assigned vendors, and have indicators if closed.

Overall Park Features:

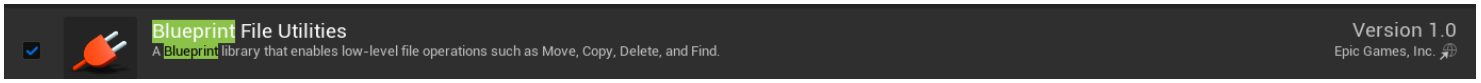
- A fully functional and expandable "Pedia" for offering further information on species/rides, etc to players
- Customizable construction buttons which open up custom build lists for each option
- Time of Day, Month, and Year (functional with third party skyspheres)
- Time Dilation (Single Speed, 2x speed, 5x speed and pause options)
- Income and Expense System
- Automatic updating of the navmesh when building/object placed
- Monthly upkeeps for powered buildings and other structures requiring maintenance
- Tip Bar for player notifications with custom messages triggered when desired
- Open/Close Park to prevent customers from showing while performing maintenance
- Easily expandable options for buildings/objects
- Star Rating updates based on visitor experiences
- **UPDATE V1.1**: Full Multi-Save/Load/Delete system functional. No limit on number of saves, and added "date/time" of save (can be changed however you like)
- **UPDATE V1.1**: Now utilizing the UE5 enhanced input mapping for all functions, making quicker switches
- **UPDATE V1.1**: Now controllable 3rd Person Character with all the build and operation functions available to the standard controller. This option triggers when the park entrance building is placed down.
- **UPDATE V1.1**: Code updates for efficiency, better organization and cleanliness.
- **UPDATE V1.3**: Edge Scrolling added
- **UPDATE V1.3**: Efficiency updates
- **UPDATE V1.3**: Statistics Display and UI updates.
- **UPDATE V1.5**: Vendors are now added and can be assigned to work locations, given raises which increase productivity, happiness, etc.

With all that mentioned, let's get started. At any point you can skip to a section you are wanting more information on. Keep in mind that some sections may refer to other sections for further information.

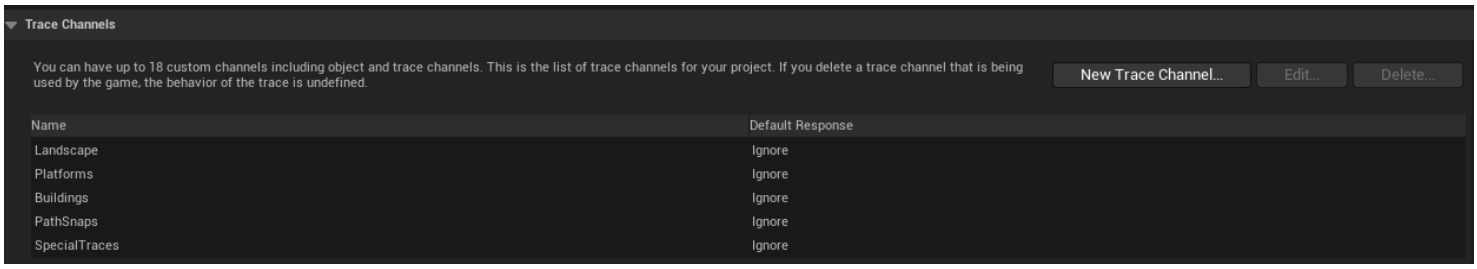
Setup and Merging into Another Project

(This is ONLY needed if you are not using as a complete project)

Plugins and Saving: If the project is merged or added to an existing project, OR if the save system isn't working as expected, likely you have a required plugin supplied in unreal engines' builtin plugins. Make sure that the plugin called "Blueprint File Utilities" is activated.



If you are merging into an existing project, you will need to make sure that you set up the trace channels as in the original project. These can be found in Project Settings. Add the following with the default response to ignore. They include Landscape, Platforms, Buildings, PathSnaps, and SpecialTraces.

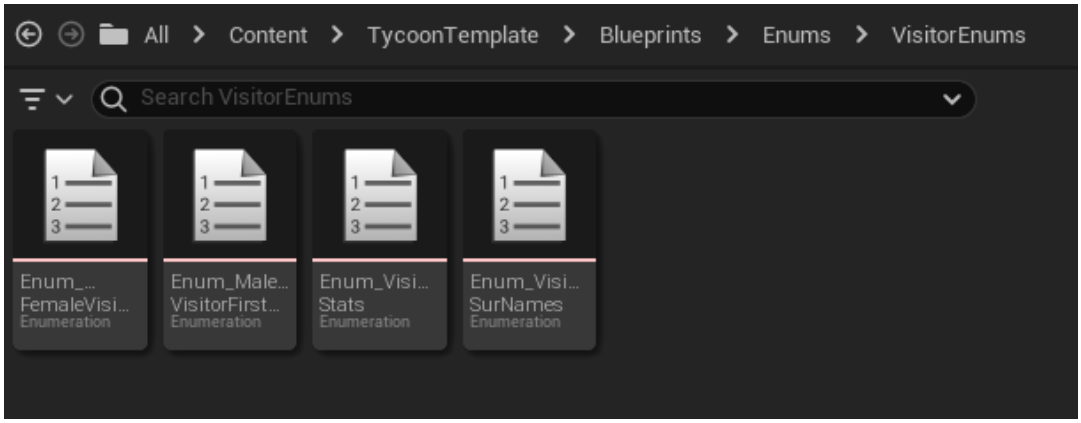


Additionally, enter the Project Settings and under Navigation Enforcing, make sure the "Generate Navigation Only Around Navigation Invokers" is checked.

Once the migration is completed, check the data tables to make sure that the trace channels for them are set correctly. There is a chance if you do not have the same order of trace channels above that the tables may use the wrong ones. For instance, if the projectile trace channel is before the landscape, it will change the data tables to have projectiles inside.

Adding A New Visitor Name

Adding a new visitor name is one of the most simplistic things to do. Within the project folder, the enums has a subfolder for Visitor Enumerators. This contains the First Name (Male/Female), Last Names and a Stats Enums.

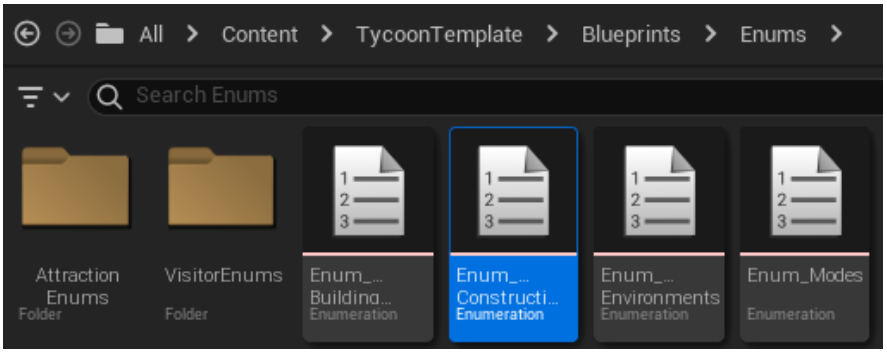


Inside of each of these has a premade list of names I added. By clicking the Add Enumerator, you can create a new entry for a new visitor name. Keep in mind the first entry must be left None, Unknown or blank for the First Name enums. This is so if a child has only a single parent it does not show a name. If you wish for it to show a name always, you can just fill this in with whatever name you wish.

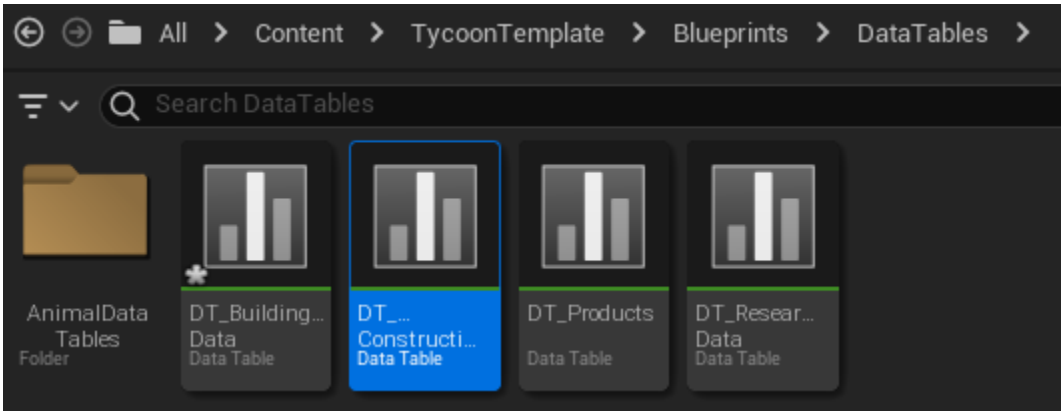
Next time you run the game, your names will be added and randomly pulled from the pool.

Changing the Construction Menu Categories

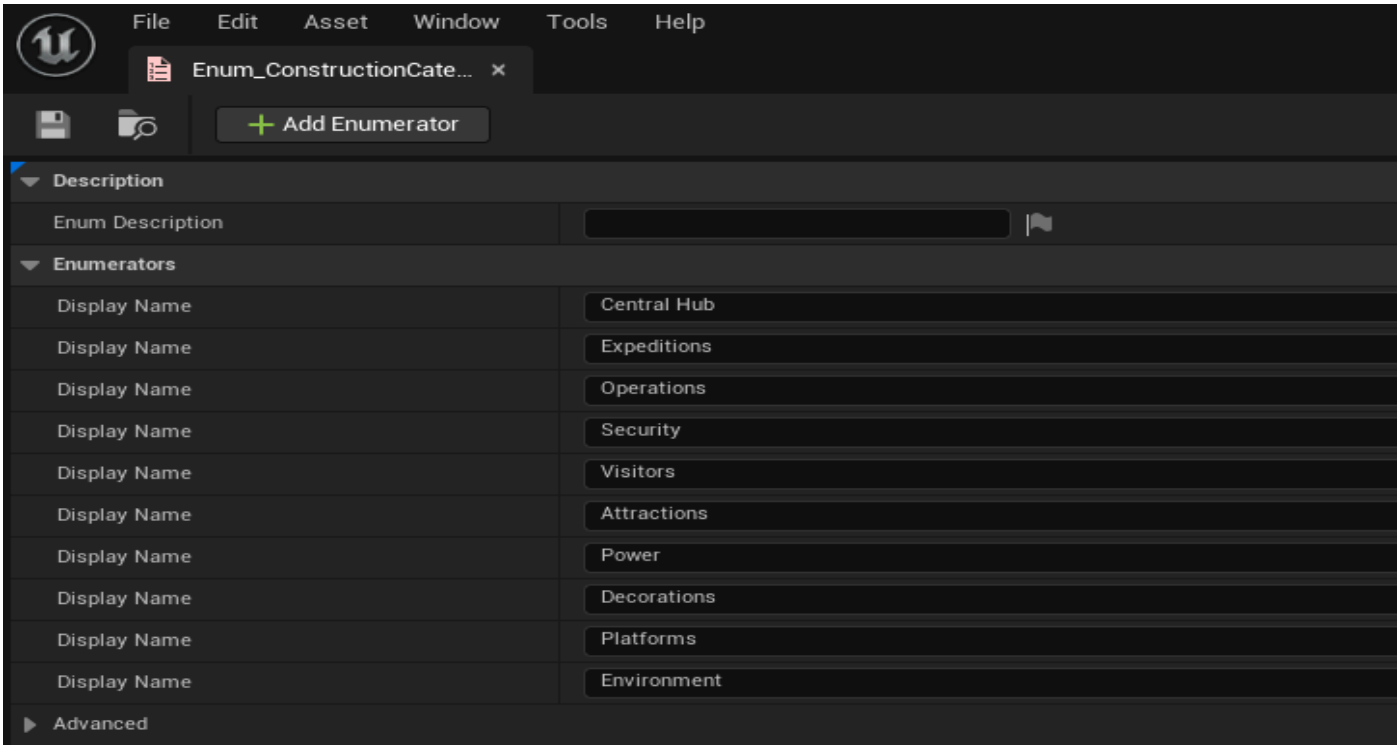
In game, the Construction Menu is the bar on the far left side of the screen. Clicking on any one of these options will pull up a selection of all available objects within that category. Changing the construction menu category names is simple. Again, within the enums, you will find the Construction category enum.

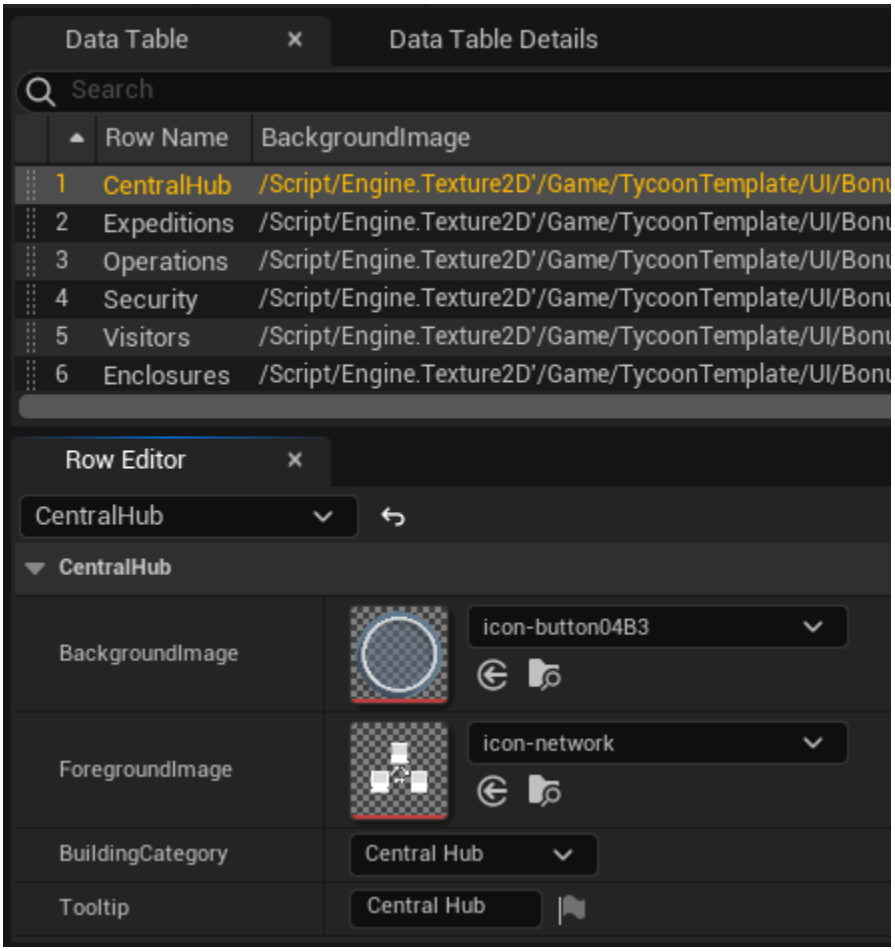


You will also need to open the data table labeled DT_ConstructionButtons



This time, instead of adding a new one, you can just overwrite one of the current names in the list.



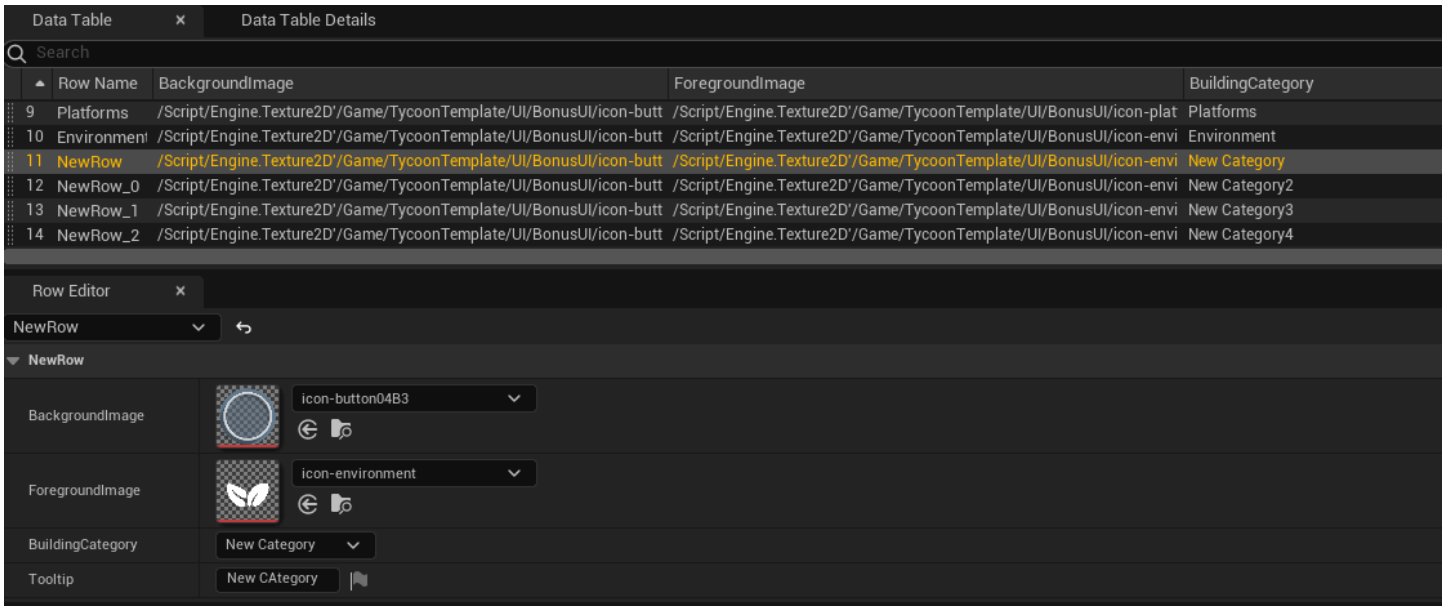


Inside the Datatable, you will see a list of the current categories. If you replaced one, you may also want to change the icons, the building category should automatically update itself, if not, select the new one from the list and update the tooltip to whatever you want it to say. This will then overwrite one of the buttons, and in game, you will now see your new name and images for this item.

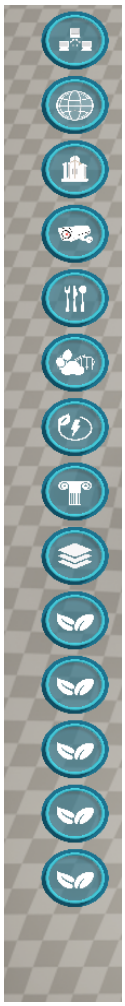
Additionally, so long as you have a building of this category to fill in the display, when you click on it, it will update the title of the build list.

But, what if you wanted to add a new category, not just replace one? What if you

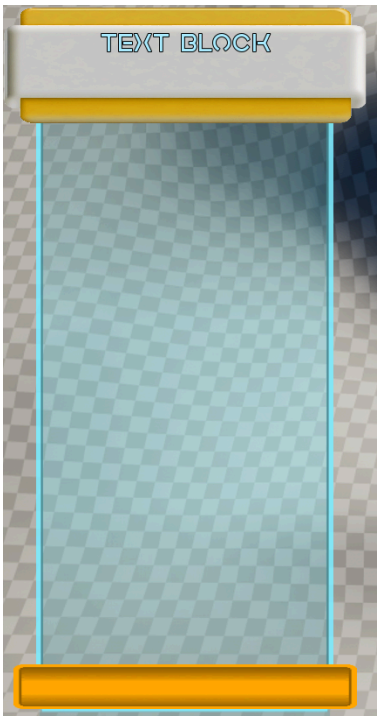
wanted 2 more buttons on the side panel? This isn't much harder to do either, and is as simple as adding a new enumerator in the category list, followed by a new entry in the data table. Ensure to assign the proper category name to it, and you can add as many new categories as you wish.



And the result:



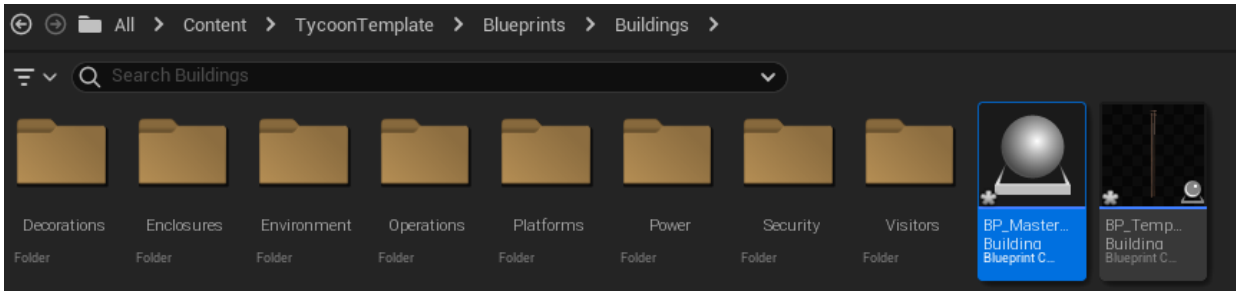
Quite a sizable list on the side with four new buttons added to the bottom. In this case, I just used the same icon as the environment one above it for the new four. There is one last thing. As you may notice when you click on one of these new icons, it brings up the build panel, however, it doesn't say it's from the right category, it will say text block, or whatever the previous option was you had.



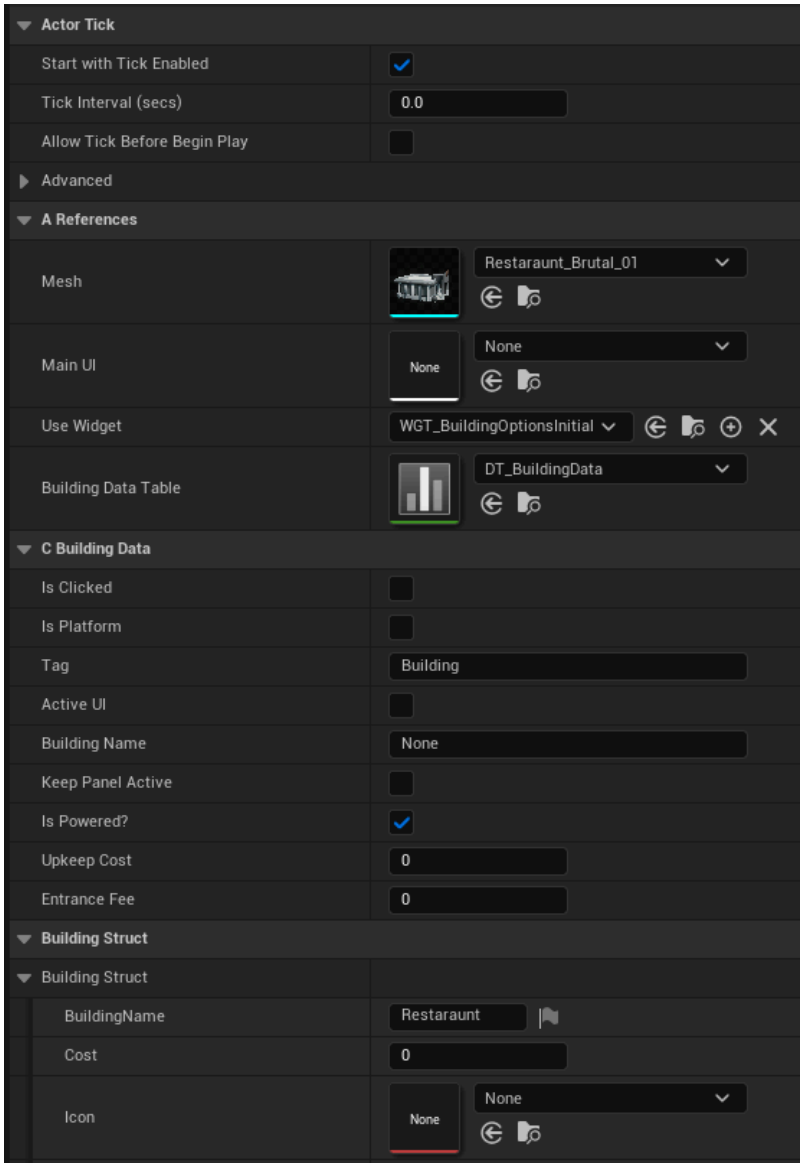
Don't worry, this is normal. The issue lies with the fact that no buildings or structures that this new category relies on have been created yet. Once a new building is made or current one assigned to the new category, the build options and text at the top will change to reflect the new category name.

Adding a New Building

New buildings are simple to create, but can be complex depending on what function you wish it to serve. All buildings come from one master blueprint called: **BP_MasterBuilding** found in this location.



To create a new building, simply right click on this and create a child of this blueprint. To keep things organized, place it into the folder of the category it falls under. If it's a new category, create a new folder for it. When you open up this blueprint it should be or likely will be in data only format. This is ok for most building types, unless you have a specific usage or function in mind for it.

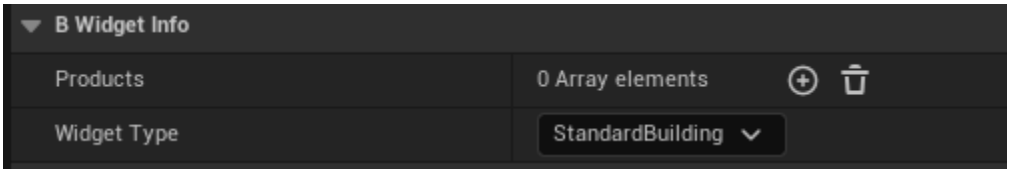


There are a few areas of any new building which must be filled in to function. Firstly, the mesh is the final display mesh of the building. The widget to use is the main widget which will open when you click on the building. In this case, the **WGT_BuildingOptionsInitial** is the most versatile and likely the only one required as it is quite expandable for any situation.

Next the building data table contains almost all of the important information needed to have the building function. Most of the rest you can ignore at this point. However, there are two VERY important pieces of information on this which must be filled in still.

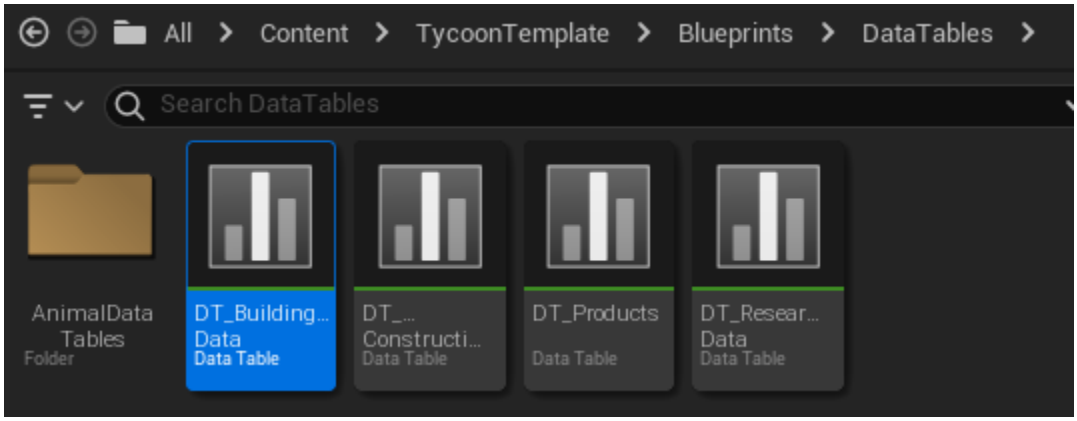
Under the building struct, you will see the Building Name. This field must be filled in precisely with the name you will be calling the building. This field determines the types of

products which it can sell. Secondly, under widget Info, there is an enum called Widget Type. This refers to what widget selector to use in the actual **WGT_BuildingOptionsInitial** Widget. This list can be expandable as far as you want, which will keep all widgets for new building types consistent and functional. Currently, there are 6 different widget types available which includes **Standard**, **Attraction**, **Entrance, Facilities, Staff House**, and **General**. Additional types can be added as needed.

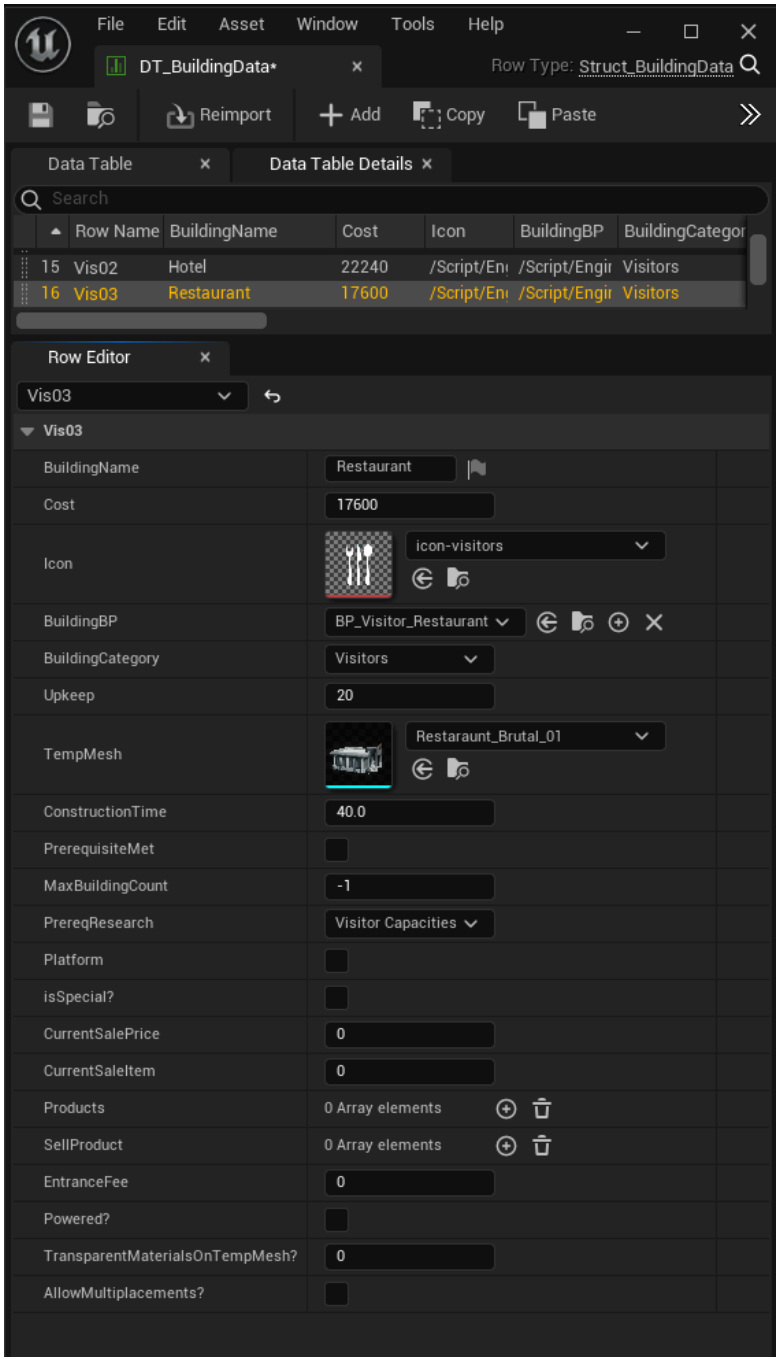


Standard Building type is used for most building types, such as restaurants, kiosks, and any building which sells products.
Attraction widget is designed specifically for buildings/objects which visitors will go to and spend money on fun. These usually have ride fees, or entrance fees associated with them.
Entrance is only used for the main park entrance. This is where a player will open/close the park, and set a fee for how much each visitor will have to pay. Finally, there is;
General widget. This is the most barebones widget, which is used on buildings which have no function except scenery.
Staff House is a very specific widget which is used for hiring and firing staff members.
Facilities is another simplified version of the entrance widget, which provides some of the same options.

If the building you created is going to serve a specific function, you will have to open it up in full blueprint mode to make the necessary functions you want. After this has been done, you will need to find the data table folder within the blueprints folder, and open the **DT_BuildingData** table.



This table contains all the most vital information about the building you will be making and will ultimately let it appear in your construction menu.



In **DT_BuildingData**, you will find a list of all kinds of buildings from every category. This is your master list for every single buildable object in the game. A one stop location. You will notice there are a lot of fields for every building, but don't worry, it's not as scary as it looks.

Firstly, for your new building, you will need to Add a new entry. You can call the row name whatever you want, this is just to keep it organized. The building name here must match exactly to the name you put in your blueprint above in the blueprint struct. This is the reference which the data will pull from for any products sold so make sure it matches exactly.

Next, you will want to set the cost of the building. This is how much the player must have to put it down, and how much will be deducted from their current currency.

The icon can be anything you like, it will be the displayed image of the building in the build menu.

For the **building BP,** find your building you just created and add it here.

After that the upkeep is how much the player will be forced to pay every month to keep the structure operational. Consider this to be employee wages, maintenance, power, etc. If the building is not powered, this fee will be waived to the player.

BuildingName: This variable is used for the display of the building name in the world, which will add an additional number to the end to keep each structure with its own unique name. This name needs to match the name of the name variable inside the building BP stated above.

Cost: This determines the initial cost to the player for placing this building down.

Icon: This icon displays in the build options for each building. It can be replaced with actual images of the structure or anything you want.

BuildingBP: This field is important, as it directs the template to know what the building blueprint will be used for this item.

BuildingCategory: As important as the BuildingBP, this field determines under which tab the building will be placed for in-game selection.

Upkeep: The upkeep is how much each day/month, this building will cost the player to keep it active.

TempMesh: The TempMesh is a static mesh physical representation of the building that will be placed down when selected from the build menu. This helps the player see what it will look like, how big it is, and change the rotation of it.

Construction time is the length of time it will take for the building to be fully completed, in seconds. PrereqResearch and PrereqMet are with regards to if the research needed for this building has been met yet or not. Currently, the Research section of this asset has not been finished, but is in the works to be completed.

MaxBuildingCount is the number of buildings of that type allowed. To have infinite, set the value to -1. Anything else will check if the building count is reached or not.

TempMesh is what shows up when you are attempting to place a building down, without spawning the actual actor, this acts as a visual representation of the object, nothing more.

The **Platform** boolean is used for when using the platform build option. By flagging this, it will mean that this object is a platform that buildings can then be placed down on top of. If it is left unchecked, this will mean that it is a building and can be placed on the platform.

isSpecial, is an important flag. This flag means that the building you created can be placed down and ignore the collision offsetting code. It will also allow the player to lower the building through platforms. This flag is great for things like ramps, fences and anything else which may have odd shapes that collide easily with others interfering with the placement.

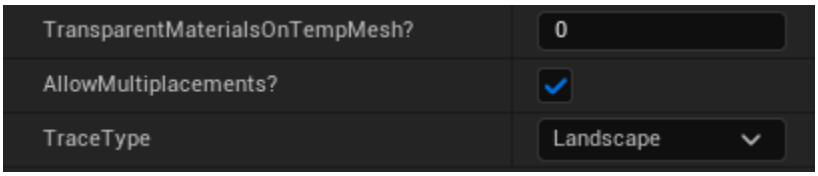
CurrentSalePrice, CurrentSaleItem, Products, and SellProduct: These values are not to be changed, as they are used internally for calculations.

EntranceFee: This value determines how much of an entrance fee is associated with the building as a base. This value will also require that either the WGT_EntranceBuilding, WGT_Facilities or WGT_Attraction be used to display correctly (set inside the building actor)

Lastly there are the **TransparentMaterialsOnTempMesh?** And **AllowMultiPlacements?** The TransparentMaterial integer relates to any material group on the mesh which should be invisible on the temp mesh. Any material number you put in there greater than 0 will result in that material number becoming see-through for the mesh. This is useful if you want to have certain parts of the mesh not seen in the placement mode.

The **AllowMultiPlacements?** Flag means that if you select this, when you place an object down, it will not keep the building active so the player can quickly place a second or third one down. If this is not selected, once the building is placed it will cancel placing buildings down requiring the player to select the building again. This is useful for buildings such as power stations, or the park entrance. **One last thing**. The order in which you fill this data table is the order in which they will appear in the build menu.

NEW: Additionally, there is now an enum called Trace Type. This new enum will be the determining factor on how the building will be placed down. This has every trace type in the engine, including the ones created for this project. If you wish to create a new trace for specific types of objects, this will determine how it will be placed down.



VisitorsOutside?: This boolean is used to determine if the visitors remain outside during the interaction with the structure or if they will go inside, which they will disappear temporarily until they finish their interaction. This is important for Queues. If using a Queue on the building, this must be checked or they will disappear instead and bypass the queue.

GarbageFillVolume: This integer is used for garbage containers, such as cans, and determines how much garbage it can hold before it begins to overflow and produce garbage on the ground.

Has Queue: This boolean flag in the building struct can be used to tell the system if a building requires a queue to be made for it to allow customers in, or not. If this option is selected, then the building MUST also make sure that it has the widget changed to have the queue button in it as well. Currently, the WGT_BuildingOptionsStandard widget

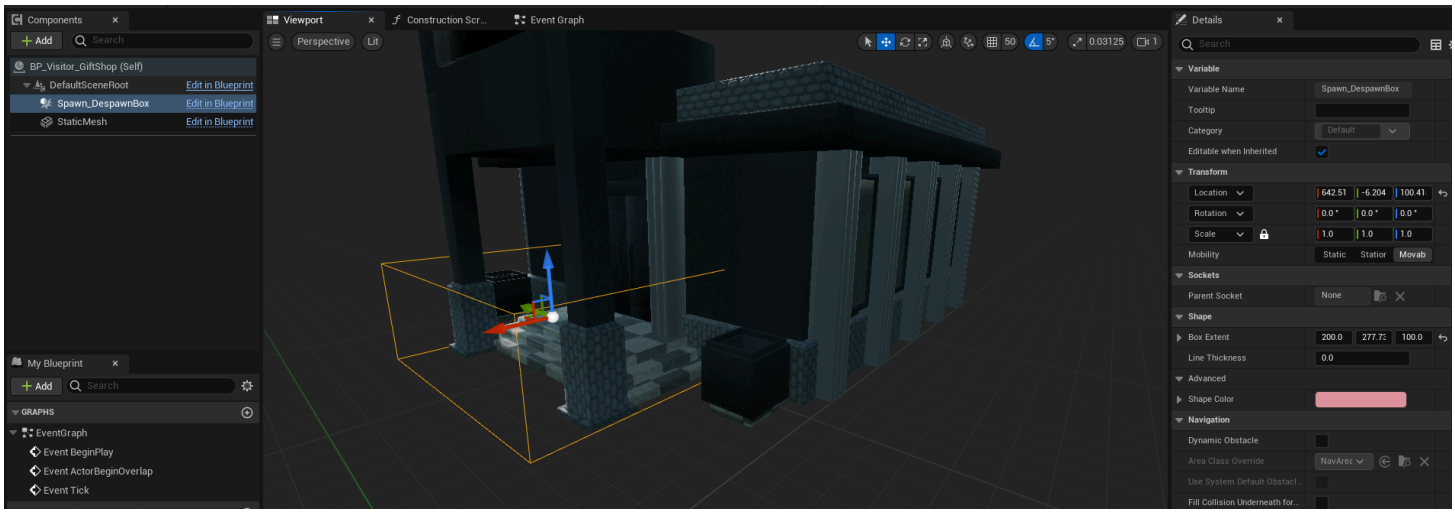
contains the queue option. At this time only one building has a demo for a queue, which is the Canteen/Kiosk. This setup can be used for any building however, as long as the above was followed.

Needs Power: This boolean will be used to determine if a "no power/closed" widget should appear over top of the building. If this is set, then when the building is closed or doesn't have power, it will display an icon above it.

Max Vendor Staff: This variable determines how many vendors this type of building can allow.

Buildings with Visitor or Vendor Interactions

For buildings with visitor interactions such as gift shops, restaurants, kiosks, or even attractions, you will need to open up the blueprint in full mode to complete the rest of it. Inside the blueprint you will notice two collision boxes called "**Spawn_DespawnBox**" and **"EmployeeDoor"**. The Spawn_DespawnBox is used for the AI to determine where to go for interaction with the building, while the "EmployeeDoor" is used strictly for Vendors to enter and leave the building. You can change the scale/placement of these boxes. Make sure that these are accessible to AI when deciding on its location.

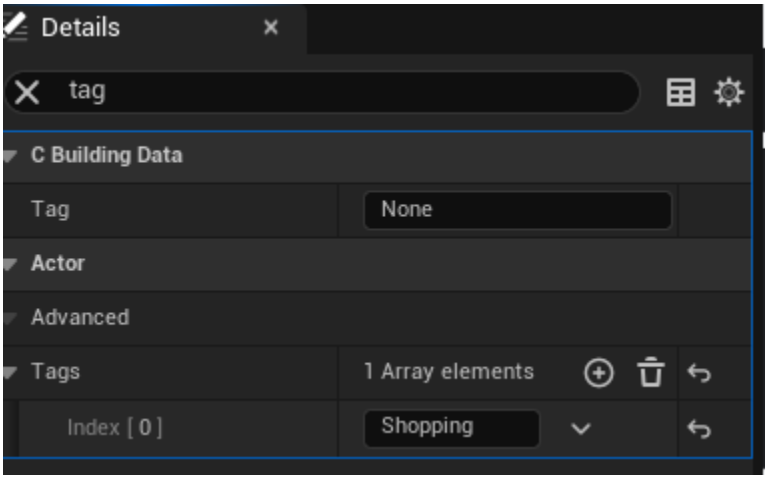


By changing the box extents found on the right panel, and its position, you will effectively change the location which the visitors will travel to to interact with this building, and that is it for changing the interaction location and size. By changing the size you also change the allowable interaction range, which now accepts more individuals, but may also look odd if it's too big. From this, you can add any extra code you want to allow the visitor to disappear during the interaction, or perform an animation.

Each building also now contains the NavigationInvoker and VisitorQueue components. These are used to provide some area around the building for navigation to properly allow the AI to detect the buildings presence and allow movement towards.

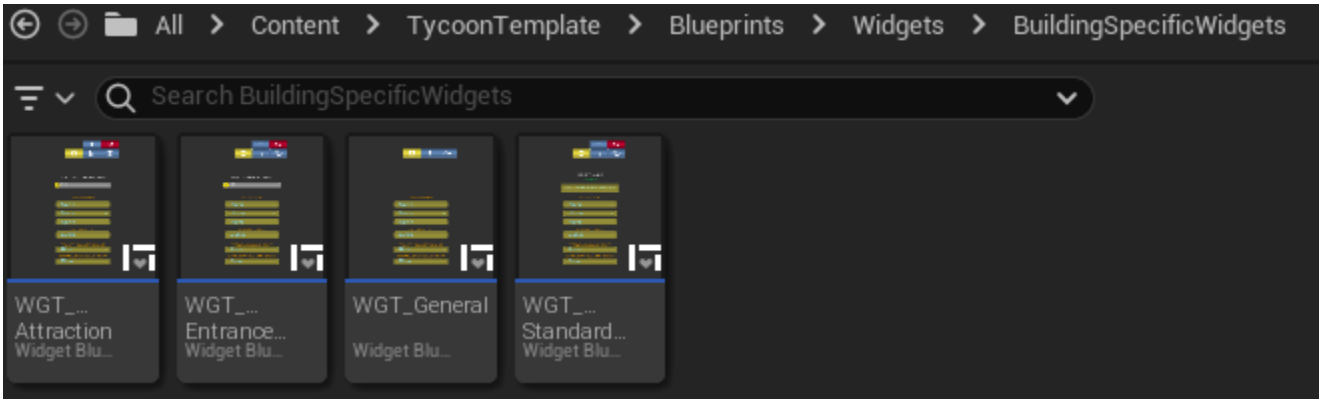
Important

The Final and one of the most important parts of creating a new building for visitor interactions is the **Tag**. The tag lets the AI know what function the building serves. In the example below, this tag is used for any Gift Shop or place individuals can go to fulfill the desire for shopping or physical goods.



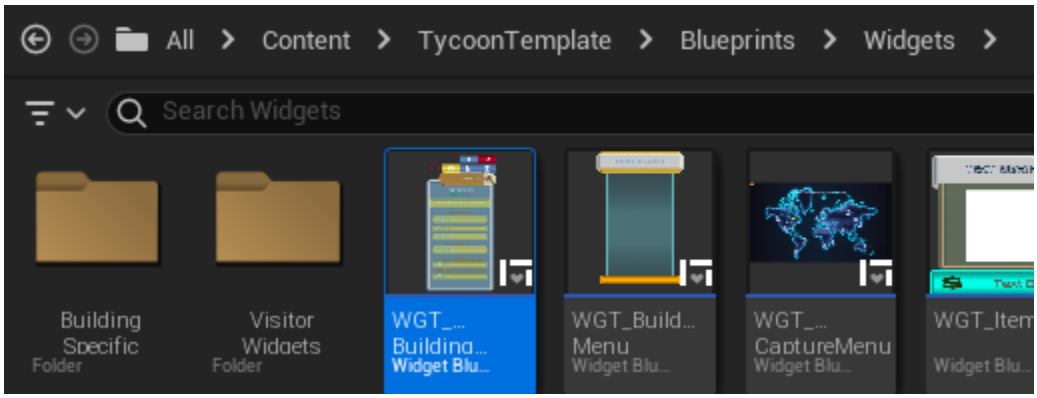
The current tag types are:
Food: Used for fulfilling the hunger of visitors
Shopping: Used for gift shops and other good stores
Attraction: Used for objects that visitors go to for rides or viewing animals.
Facilities: Used for finding a washroom or such.
Additional tags can be used/made as you see fit. If you want to have multiple entrances, you may need to add a flag for "entrance" and change the behavior tree to search for this, like it does for the other tags.

Adding a New Widget Type



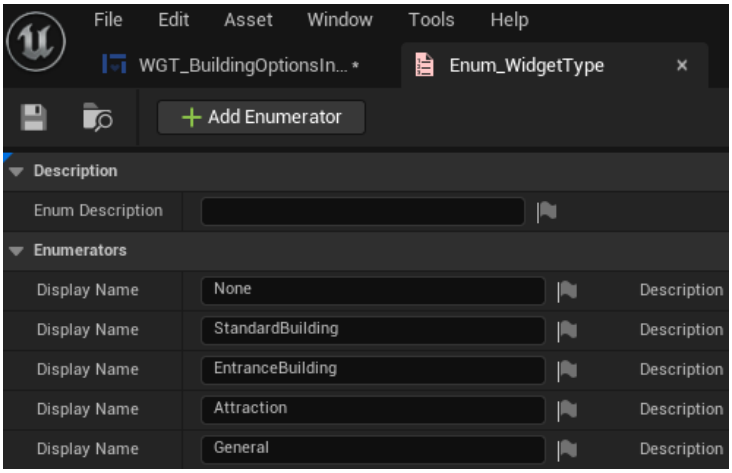
New widget types are useful if you have a building type you want to have specific functionality. Inside the **BuildingSpecificWidgets** folder there are four different widgets which relate to the building's widget you assigned in the building BP earlier. If you are wanting to create a new unique widget to appear when you click on a specific type of building. Duplicate one of the existing widgets which closest resembles the type you would like for the widget. Rename it, and save.

Next you will need to open the **WGT_BuildingOptionsInitial**



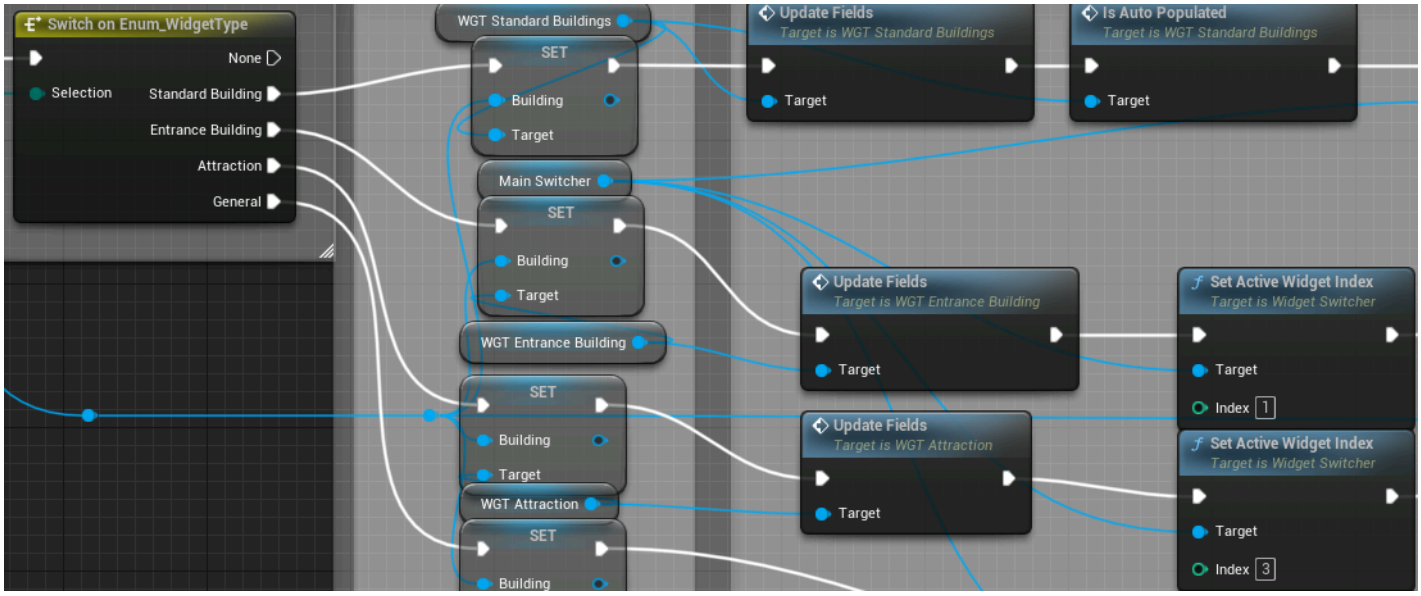
The hierarchy panel contains a section **MainSwitcher**. To add your new widget first create a new Overlay, and then search for your new widget inside of the palette and add it attached to the new overlay, like the others.

Next, open up the **Enum_WidgetType** file and add the new enumerator to the list of widgets.



This enum is required to allow the switcher to function correctly.

Back inside the **WGT_BuildingOptionsInitial** a few additional changes will be required. You will firstly notice a new pin on the **Switch on Enum_WidgetType** with your name you just created in the **Enum_WidgetType**. This is your new widget.



You will next want to follow the same design as currently established by getting a reference to your new widget like the others and then setting the building building variable to the current building. After that you will have to run the update fields function if you have any information such as power status, entrance fees, or any further data that needs updating.

The next step is to set the **Active Widget Index**. You may be wondering what number to use for this. If you are, and are unfamiliar with the widget switchers, the index will be the order you put the overlay folders in **WGT_BuildingOptionsInitial**.

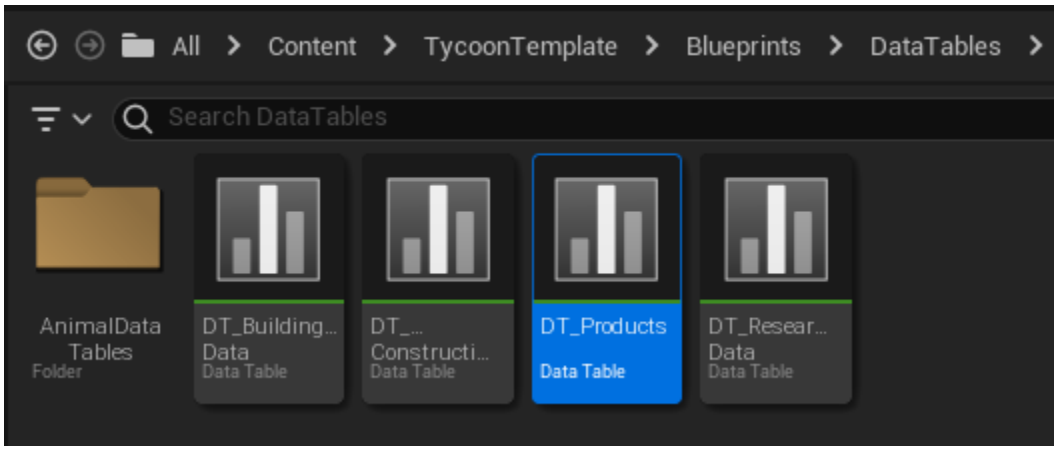
Notice on The Building Widgets

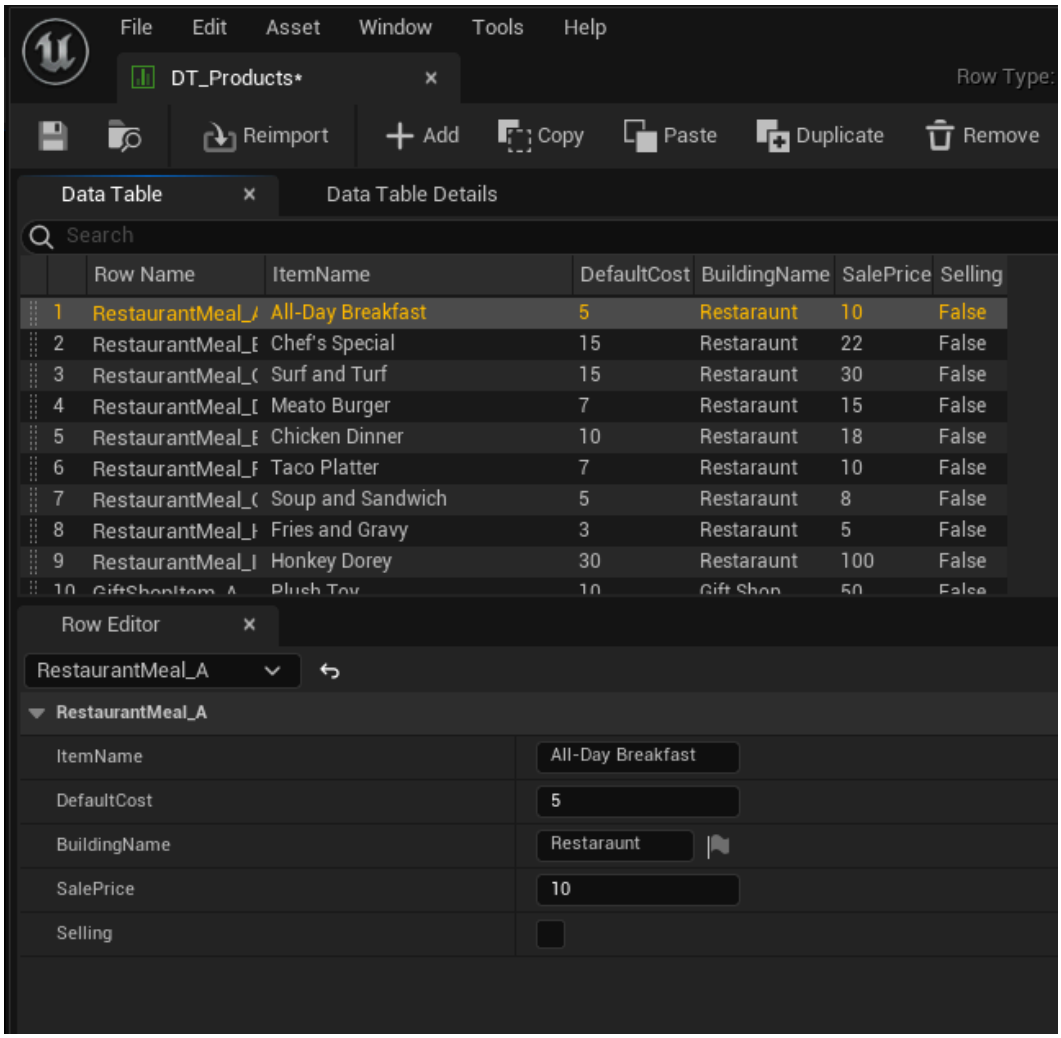
Though there are a number of drop down selections visible on the settings panel, these currently do not produce any sort of function, but are there to show how you can have various options in that area. At a later date, these functions may be added.

Reason for not adding functionality for this? The main reason for not adding any functionality to these is because it would require specific buildings, specifically designed materials to allow for these kinds of changes for each building.

Adding New Products For Visitors to Buy

To add new products to the list for customers to purchase, first open up the data table called **DT_Products**.



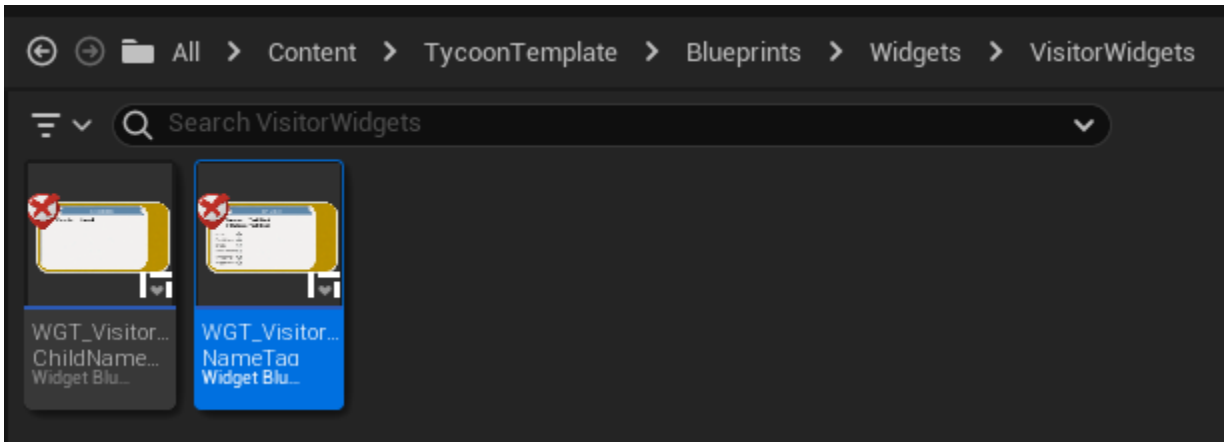


This data table contains all the current information needed to start selling various products. Let's take a look at what is in here.

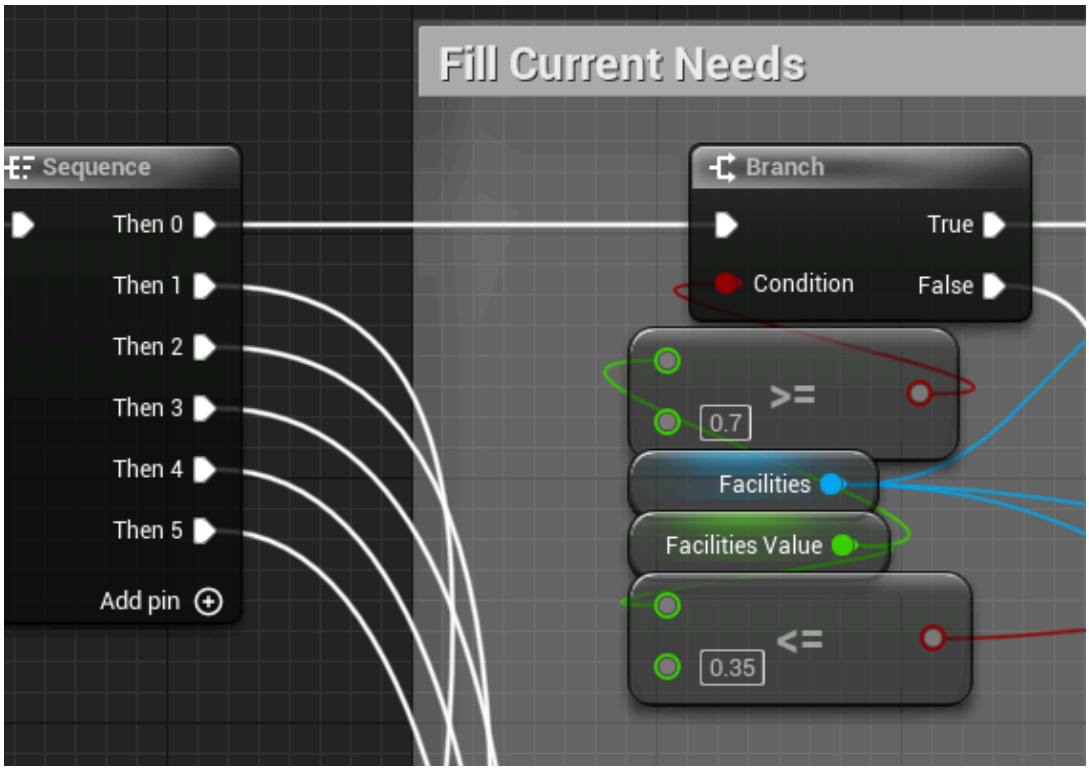
Row name is simply for your own benefit to keep things neat and organized.
Item Name is the display name the player will see in the building sell list.
Default Cost is what the price of the product is to the player.
Building Name specifically names the building that carries these products. (If you ran through the process of making your own building so far, this will sound familiar. It must match the building name you made to have it function correctly.)
Sale Price is how much the product will cost the visitor to purchase. The income the player receives is the sale price - default cost. Finally;
Selling Is a flag which determines the initial state of the product in your product list. If this is set to true it will be the default product that is sold. You may have as many as you want set as selling.
Like the construction list, the product list will also populate in order that they are put into this list. However, ONLY the products associated with the kind of building you specify will show up to the player when selecting that building. This helps keep products in an easy to find space and quickly expandable.

Changing Visitor's Tag Display Rating Thresholds

What this means is that each visitor if clicked on brings up a statistic/demographic of the visitor. This includes displaying their current thoughts and ratings on the park. Though it is a percentage of 100, it is currently simplified to a thumbs up down or sideways. This makes it a bit simpler for the players, but can be changed easily enough.



Inside the widgets category, there are currently two visitor widget types. One is for adults, while the other is for children. Let's open the adult or default widget. The event construct section of the widget fills in the required information on the visitor selected. Within this, you will see a list of variables being filled.



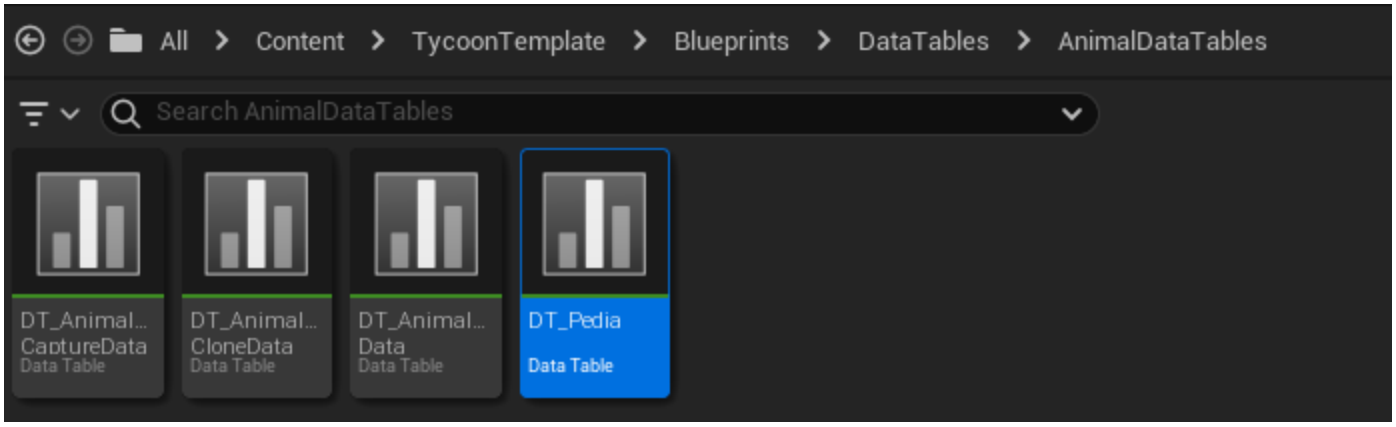
Here you will see the facilities value which has a threshold of 0.35 or 35%. This means that if the visitor's opinion of your facilities (washrooms and such) is equal to or less than this value, you get a thumbs down. If the value is above 0.7 (70%) you get a thumbs up, the rest is neutral. Currently the reason for this set up is because each statistic can have a different threshold, so some can be very tight with 60% being good, and 50% being bad.

Changing What Visitor's Default Stats Are

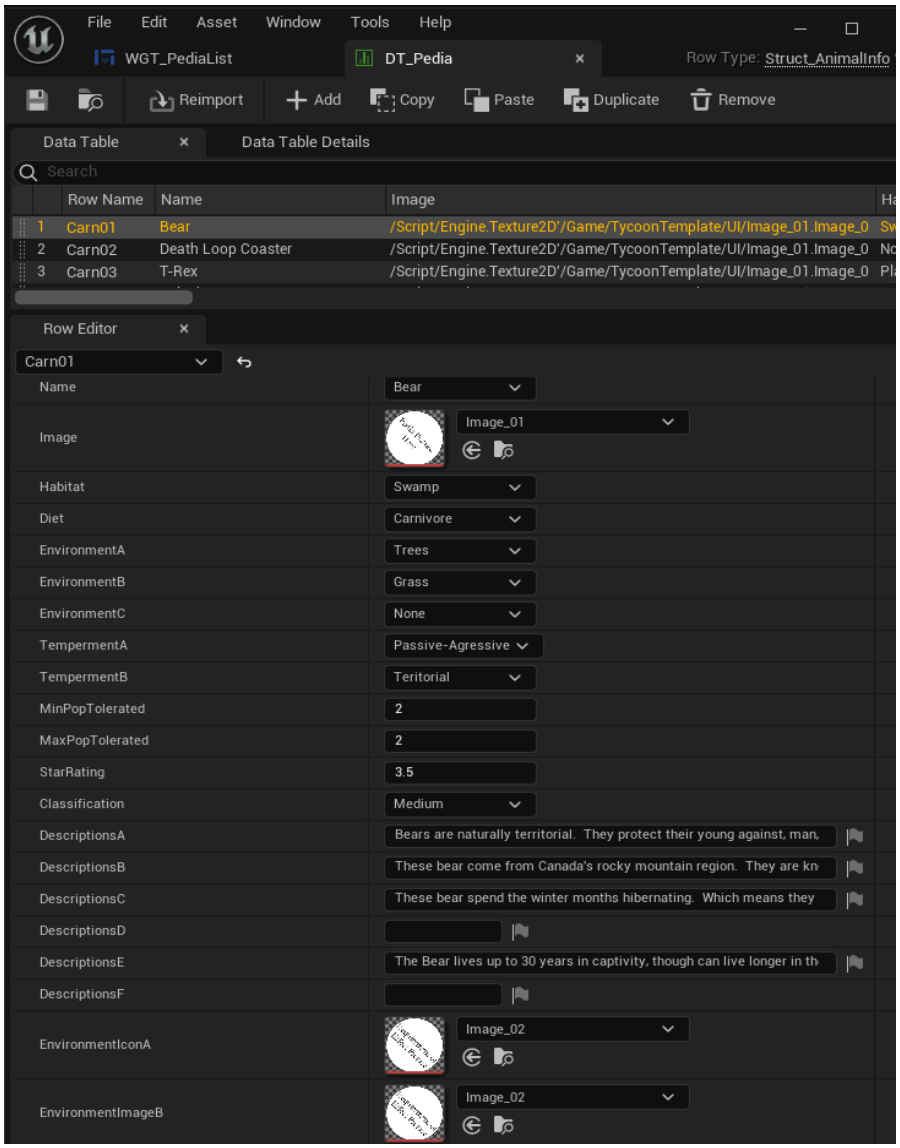
Visitor default stats are created on visitor spawn. This means that it's random, and unique to each visitor to show up. There are controls however to set the defaults of how they feel and what kinds of thresholds they have. Inside the **BP_Visitor** blueprint, there are a number of variables that can be changed. One important variable to keep in mind is the **NeedDecayRate** float. This is the rate at which some of the stats for the visitor decays. There are a few which do not decay by this means, such as happiness or excitement.

New Pedia Entries

Pedia entries are fun and nice for filler in a game of this nature. Sometimes having a bit of story behind the animal, statistics for it, or background information/ratings on rides may be just what you want. This section helps address this.



The pedia is currently designed mainly for animal entries, however, it can be easily customized for your own needs. Open up the data table named **DT_Pedia**.



It will open up a fairly large data table and may seem intimidating at first. However, it is fairly simple once you look at it.

In the list you will see various creatures, rides, and things. These are all examples that can be added in for the pedia.

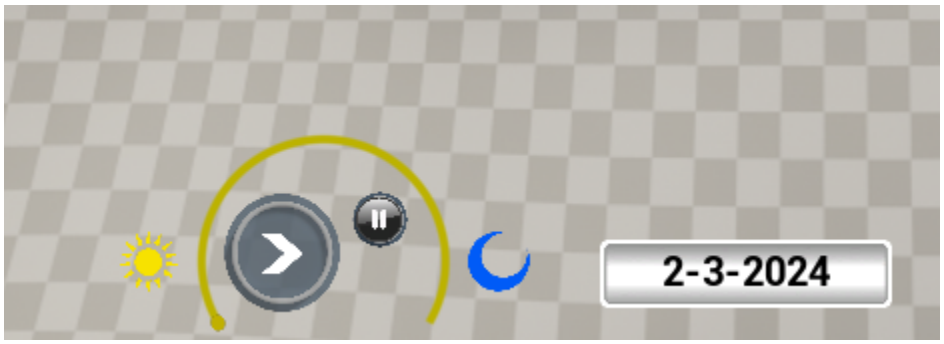
Once you look through the entries you will see how it shows within the pedia in game. By just clicking Add, you will create a new pedia entry which will automatically appear in the list (again in order of first to last).

Current Tycoon Controls

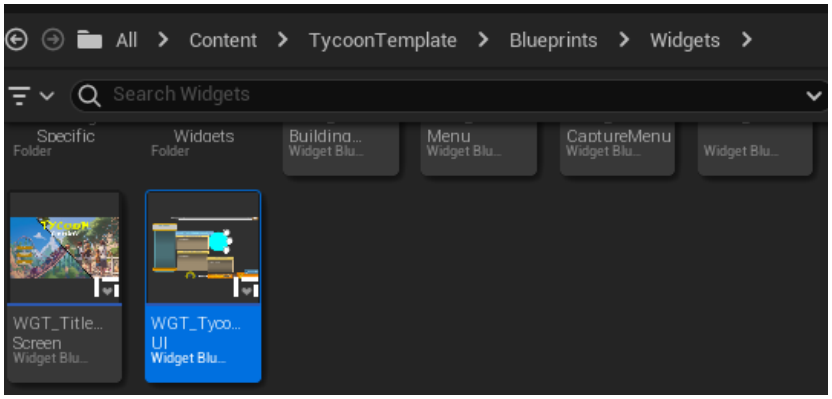
The system currently uses the UE5 Enhanced Inputs. These have been used for both the main build mode and the new third person mode. The controls established are;

WASD for camera movement (Third Person Movement)
LMB to select buildings, items, etc
MMB Scroll to zoom in and out
MMB down and mouse move left/right to rotate
RMB to cancel (Hold in third person mode to rotate)
Q, E to rotate the building being placed
Z, X to raise/lower the building being placed
These controls can be changed however you would like, within the **IMC Asset**.

Changing Dilation and Time Options

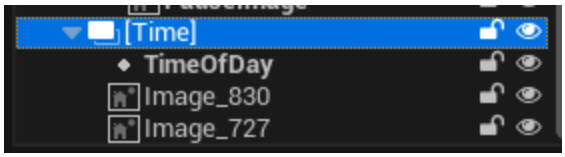


During gameplay, you will notice this odd looking part of the main hud. While it may not look the best (sorry about that), it functions as expected for these types of games. Essentially, when the player clicks on the arrow, the time will speed up, clicking a second time will speed it up more, and a third will bring it to normal speed. The pause button will stop all actions and resume them, while still leaving the player with control. The ring around it represents the time of day, and to the right is the date.



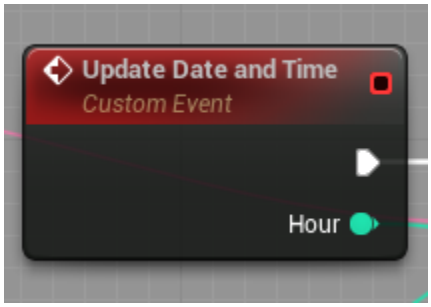
Any of these can be changed quickly, or visually altered to your likings.
Within the widgets folder you will find **WGT_TycoonUI**. This is your go to for a number of changes.

Open up the UI and go to the designer section. Inside the designer, you will find an overlay section for these features called **Time**.



These relate to the arch which tracks the day/night cycle, as well as the sun and moon. The images can be changed easily, as can the design of the day/night arch.

Within the graph section, it is possible to change the dilation speeds as well. Finding the On Clicked for the dilation button you can follow the line along to the three time dilations set to 1, 2 and 5. These are a multiplication factor of normal time. You can change them to whatever you feel to a maximum of 20. Anything above that will not take effect. After that the next nodes will set the images associated with that time.

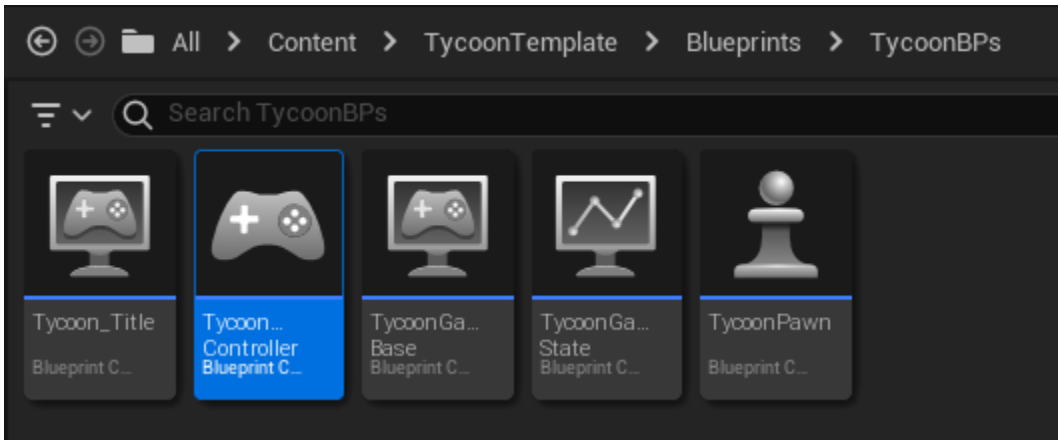


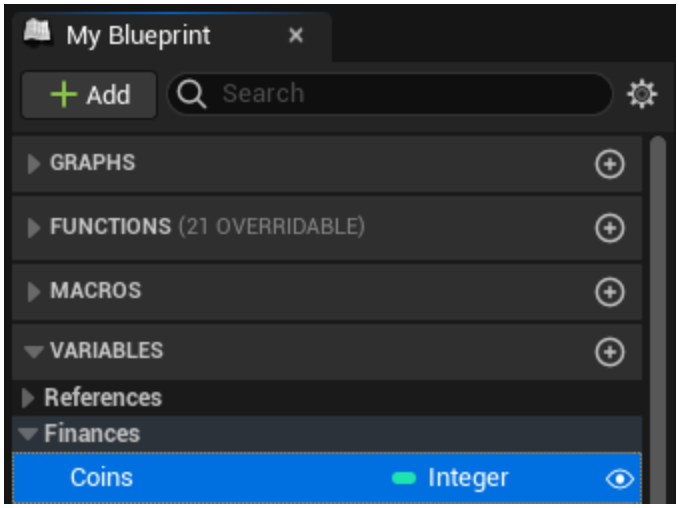
Using Ultra Dynamic Sky For Time

If using something like Ultra Dynamic Sky, you can also plug in the time of day variable into this to read it and track your days for you. By locating the update Date and Time function, you can change the variable that is pulled from to the Hour/Date stored within the Ultra Dynamic Sky, or another Skysphere asset.

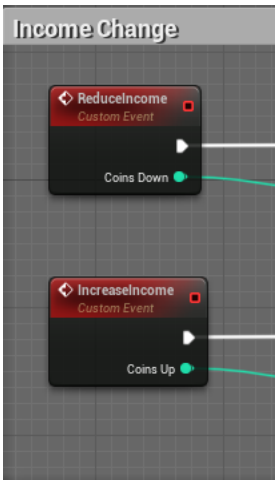
Changing the Starting Finances

The starting finances can be found within the **TycoonController.**



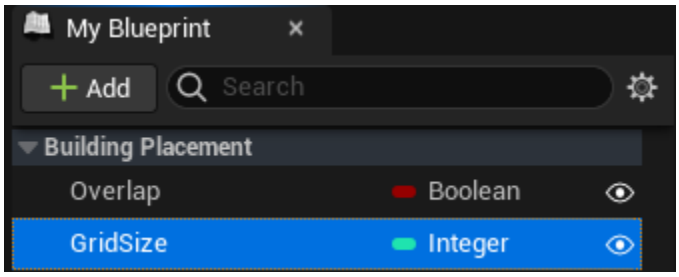


By accessing this blueprint, you can find the integer called **Coins**. This variable's default value is what the player will begin with.



You can increase and decrease this variable with the use of the **ReduceIncome** and **IncreaseIncome** Events within the **TycoonController.**

Grid Sizes (Snapping)

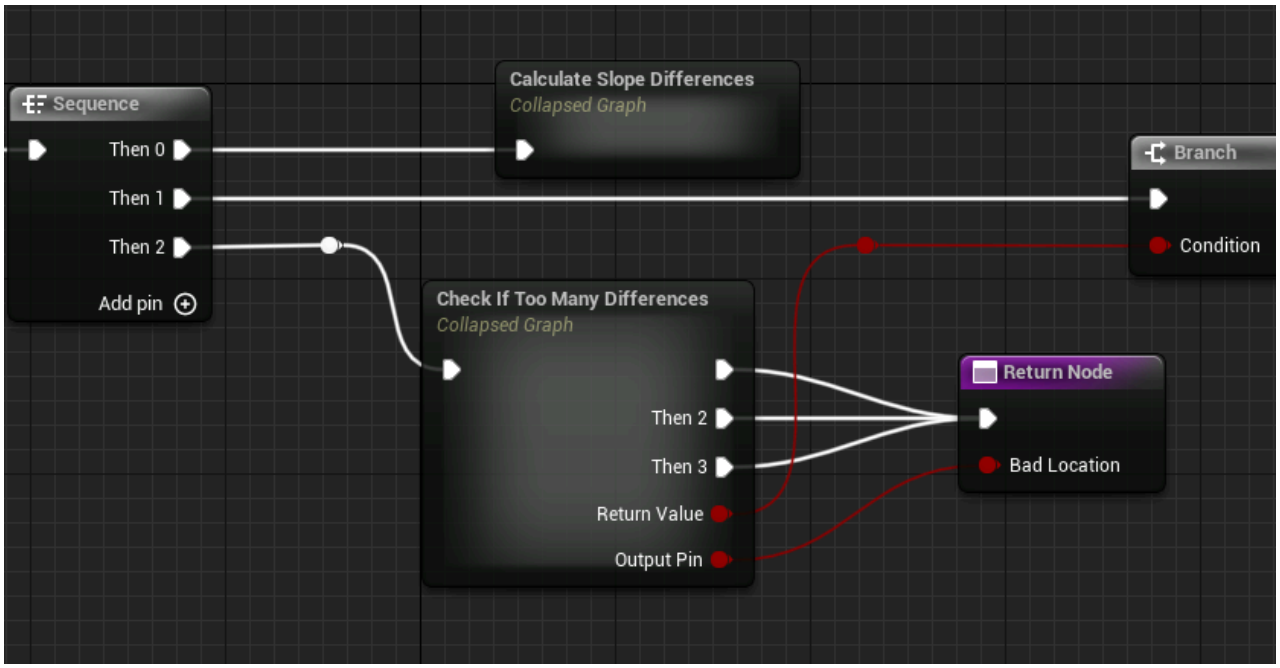


The grid size can be changed quickly by opening the **Tycoon Pawn** and finding the variable called **GridSize**. It is currently set to 125 units or 1.25 meters. You can change this to any interval you wish, just keep in mind your models may need to follow this sizing as well if you want objects to fit correctly. Vertical snapping is a little different. Currently, the process uses half the height of the grid size for vertical snap placement. This can be changed as well. To change the Grid Size for X/Y axis, open the **Tycoon Pawn** and change the value of **GridSize**. Additionally, if you want to change the height, change the value of **SnapHeight**.

Building on Any Surface

With the V1.1 update, the detection system has been greatly simplified. The previous method requiring going into the code to tweak it has now been boiled down to a simple enum in the **DT_BuildingData**. This can be seen in the Adding a New Building section.

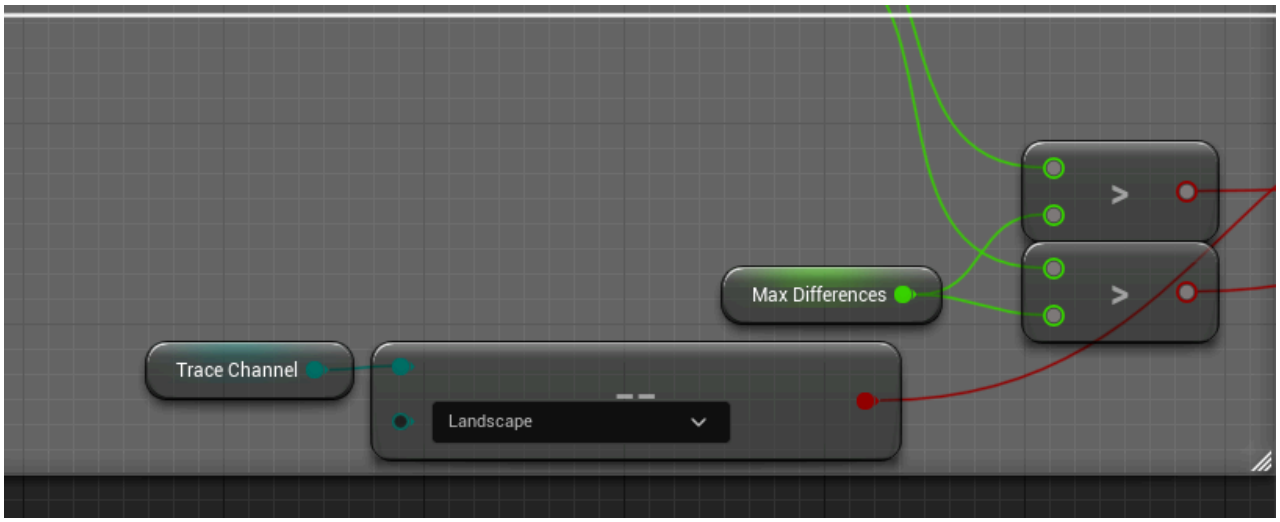
Slope Detection Allowance



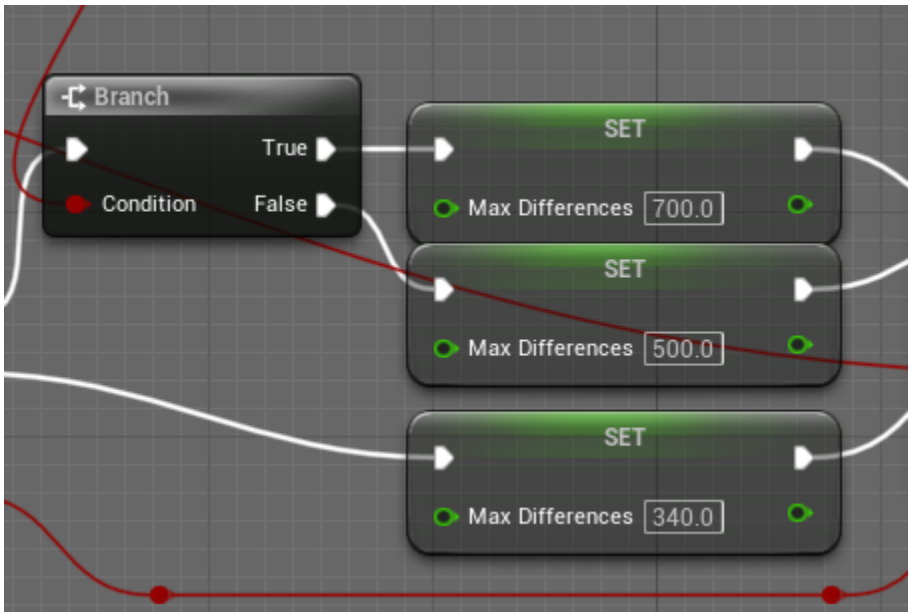
Within **BP_TycoonPawn**, there is a function called **Calculate Slope**. This contains a lot of the fine details on how the project does its tracing and slope detection. In this case, when dealing with landscape and the allowable slope, it can get a bit trickier, and will require a bit of playing around to determine what is an acceptable amount for your project.

Inside this function, you'll see a collapsed graph called **Check If Too Many Differences**. This graph contains the logic for the second part of the slope detection. The first part is contained in **Calculate Slope Differences**.

Inside the Check If Too Many Differences graph, you will find two areas which can be modified to meet your needs.



The second area you will need to chance is beside it and looks like this:



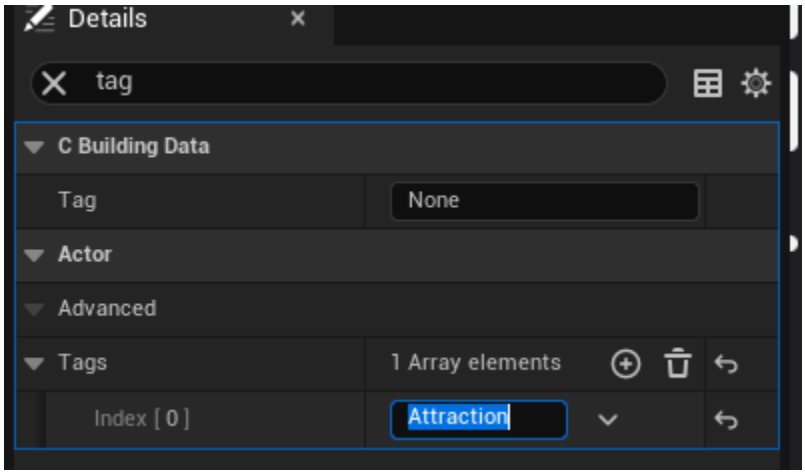
The reason for three max differences settings is because of the different kinds of tracing and the size of the building. The bigger the building, the more variance it likely will require. In this case, the entrance building has a tolerance of 700, while everything else on landscape has a tolerance of 500, and on platforms, it's 340.Additional tolerances can be made for specific objects if you find they do not react the way you wish.

MaxDifferences is essentially the value which determines how many points can be above the threshold before it is considered a bad location and thus not allow placement of the structure. By increasing this number, you effectively increase the allowable slope. Reducing this number reduces the slope. Currently it is set to 380.

The value set in **MaxDifferences** does not relate to the angle itself. More so it relates to the distance from each corner of the structure and the middle of the structure compared to the height before it hits the landscape. This can be

skewed if your landscape is higher than 0 in initial placement. In this case, my test map is set to 3000 units above the ground.

Creating A New Attraction



Attractions are much along the same lines as creating a new building. So, the first step is to make a child of the **BP_MasterBuilding**. After that the first thing you will want to change is adding a **Tag** to the top level of the blueprint actor you made. You may recall this from the 'making a new building' section. This MUST be labeled as **Attraction**, unless you have altered the naming of them, in which case you will have to change the behavior tree of the visitors as well.

After the tag has been added, You will want to adjust the **Spawn_DespawnBox** position and extents similar to creating a new building for visitor interactions. This tells the visitors where they should go to interact with the object. For the remainder of the creation process, refer to the **Adding A New Building** section.

Park Star Rating

The park star rating is calculated from the visitors' experience. As they leave the park they give your park a rating based on 6 different factors/needs they have. These are **Excitement**, **Shopping**, **Hunger**, **Facilities**, **Happiness**, and **Thirst**. These can be changed to whatever you would like. The calculation involved takes into account all visitors ratings prior to and gives a rolling total as a scale of 5 stars. This does decrease if they leave the park unhappy. Which also affects the number of individuals who will enter your park. The amount that each of these stats is located in the **behavior tree** of the visitor type, with the associated task.

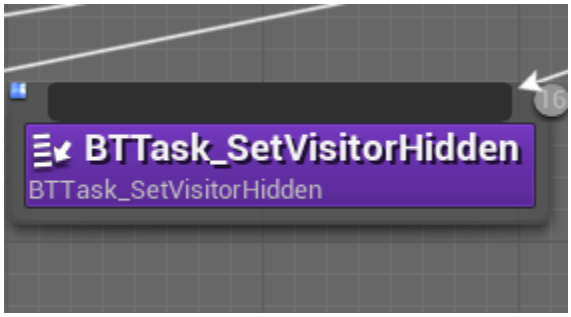
Changing Visitor Thresholds

Every visitor when created has randomly set up thresholds. What are thresholds? A threshold in this case is the limit to which a visitor can take before they get too upset with the park and decide to leave. Within **BP_Visitor**, and in the construction script, there are a series of set functions relating to each of the thresholds. By setting the min/max values it will randomize what that visitor's personal threshold is. In that same construction script, there are a series of set

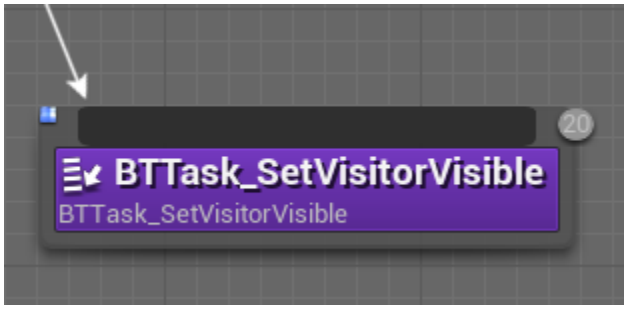
nodes which set the visitors beginning feelings/needs. If these are set to be lower than the threshold, the visitor will wander in, and turn around and leave.

Getting Visitors to Disappear (aka enter a building)

Visitors can either stay outside the building to interact with it, or “enter the building” by disappearing. To make visitors disappear instead of interacting outside of a building, you will see an example with the facilities within the behavior tree.



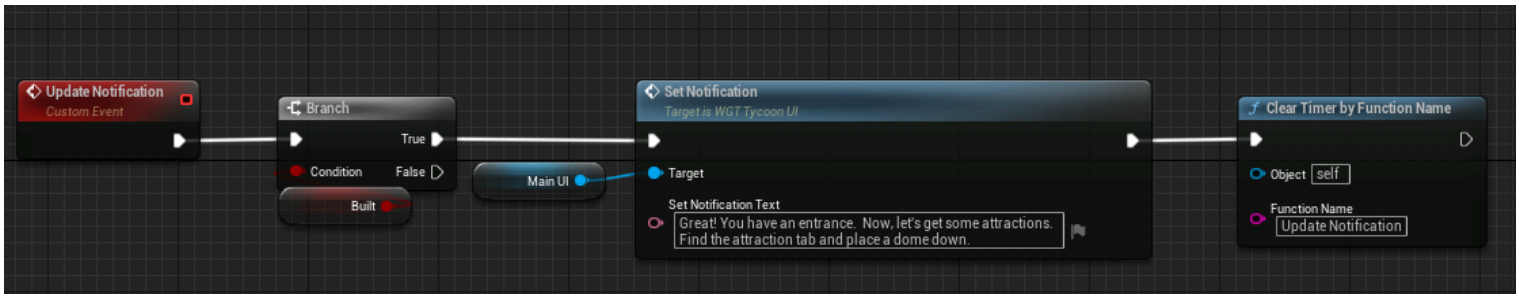
Two nodes are needed to be added within the behavior tree on the interaction. First is the set to invisible. This node not only will set the visitor to hidden, but will also remove collision from it, allowing for other visitors to step into its place as well.



The second node is the set visible. This will undo the hidden and no collision on the visitor. These should be placed on opposite sides of a wait node to allow them time within the structure.

Setting Notifications in Tip Bar

Notifications or tips, and even tutorials can be added to the tip bar at the top, or into any other kind of container you wish, such as a log. In this case, they are set up currently to function with building placements, or when specific functions are triggered inside the child BP actor.



In this case, when the building is completed, it triggers the notification to show the next part of the tutorial (seen in the demo). It is very important to clear the timer once this trigger is done, or to add a do once at the beginning if it is intended to occur only the first time. This will reduce the amount of calls, and increase performance.

Calculating Visitors

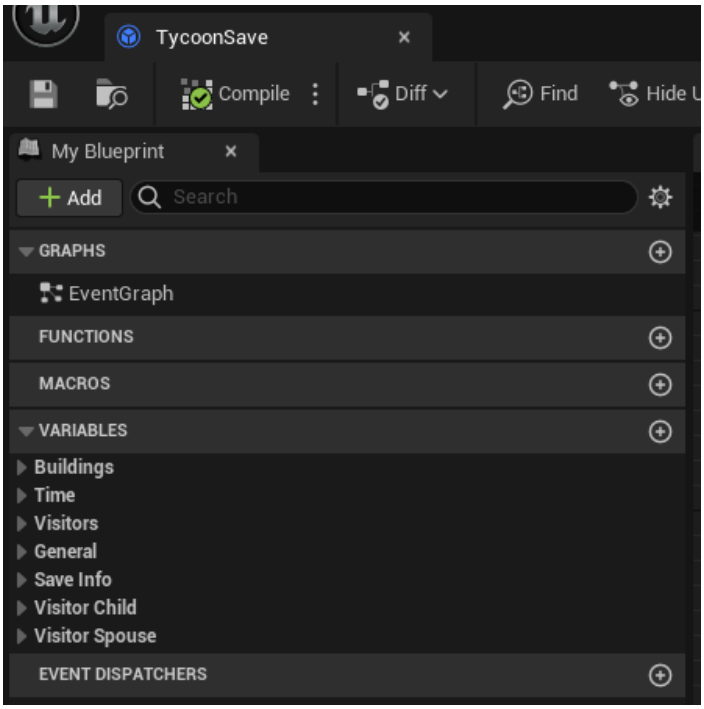
BP_Operations_Entrance is the hub of spawning visitors in your park. You can handle them showing up however you wish, however, the core function has been established. Inside the BP for this building, there are a number of functions. The main one of interest is the **Spawn Visitors**. This area contains a function called **Visitor Calculation**, which is used to calculate the number of visitors which will spawn. This can be handled in any way you see fit. Currently, it operates using the parks rating and a **MaxSpawnVisitorsAtOnce** and **MinimumVisitors** variable, which determines the thresholds of visitors to allow in at a time and the least that will show at minimum.

Taking Control of Your Own Character

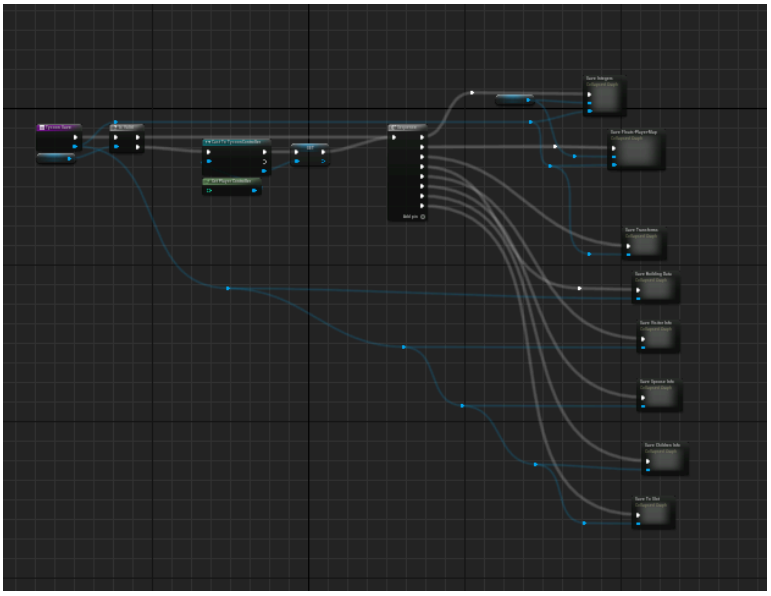
New in V1.1 is the ability to control your own third person character (or pawn of your own creation) in conjunction with the current controls to make your own park, or take a fun stroll, who knows maybe enjoy some animals or roller coasters.

With this new update, there is a symbol in the bottom right corner of a person with a + sign. This will allow you to take control of your own third person character. Currently, it is designed so that you **ONLY** get the character once **BP_Operations_Entrance** has been built. This character also offers all the functions of building your park in third person (or first person if you change the camera positioning). Inside the **BP_Operations_Entrance** actor, there is a World Player variable, which is related to the player that will be spawned when the building is made to allow the player to control.

Save/Load and Delete



This is another new to Tycoon Template feature. This full featured save/load and delete system allows you to save as many files as you want. It currently allows all features, and assets to be saved and loaded, including the products being sold in stores and the availability/price.



The save file, known as TycoonSave, contains the following variables for saving. If there are new features that need to be tracked, the adjustments can be easily made in the save and load functions.

The save function is located inside **WGT_SaveSlot**. This widget controls the data which is being saved to file. Any additional information wanting to be added, such as player name, or custom character info can be added here by following the same design I have used for reference.

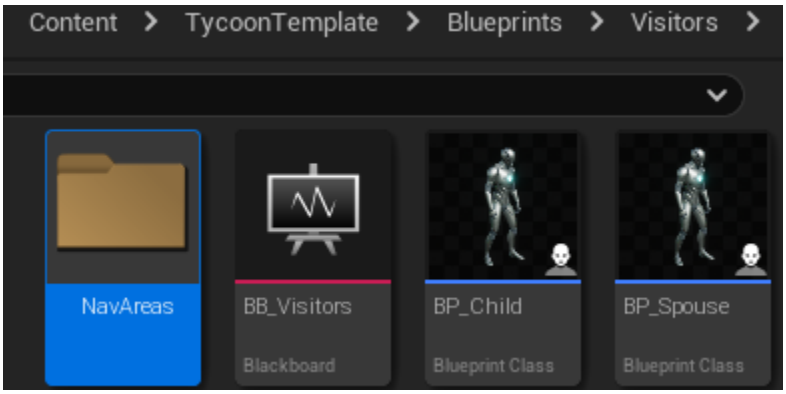
On the "load" side of things, it is a two part piece. The first takes part in **WGT_SaveLoad** which is designed to populate a container with the current save/load files to choose from or a new one to create. This will then pick the correct level from the data in the save file to load, and load up the level.

Because of the nature of how this is done, the load of the data can ONLY occur after the new map has been opened. This means that, once the map is opened, the game instance communicates with the Player Controller to determine what should be loaded. Part two is found in **BP_TycoonController**, under the function called **LoadTycoonData**.

Following the same order of saving, the loading system quickly populates the world with the actors, and repositions the Visitors and repopulates their stats. When your game begins to get more detailed meshes, and information in this save file, you may wish to add a loading screen to mask this while it does so. Currently, it occurs instantaneously.

Paths

In V1.1, paths have been added. This feature functions as a "preferred" route rather than a restricted to a spline. The paths are made to be versatile, allowing for straight, curved, or rounded ends. The path system can be tweaked quickly and easily depending on your needs. Inside the Visitor folder, there is a folder named NavAreas. This folder contains two types of areas currently in use.



The **NavPath** and **NavPlatform** are two kinds of navigation modifiers added to specific meshes/buildings. They determine a cost for traversing certain terrain. In this case, as the names suggest, the **NavPath** is assigned to each of the path blueprints, while the platform is assigned to each of the platform blueprints. Higher numbers are more difficult/less appealing to the visitors to walk, while lower numbers are easier. So, in this case, the value makes the pathways more appealing than the platform to walk, but they won't go out of the way if it's quicker to keep going forward.

The path system does have another feature, currently in development, which can be seen within the path blueprints. This is experimental, and not required for the path system to work. You can create your own paths from the standard **BP_MasterBuilding** as children to create whatever paths you wish.

With this, you may now add these two Nav's into any modifier you want to cover specific objects/areas by following the same design as the current paths structure.

To Snap or Not To Snap

The snap method which has been established in this template, and earlier mentioned now has its nemesis. The Anti-snap! The function works simply by holding Left Ctrl down, while placing a building to remove the snap to grid, which will allow you to place objects such as lamp posts, buildings, benches, etc at any position you want. When Ctrl is released, the snap will return back to its normal state.

Staff

NOTE: Security is not fully functional at this time.

Staff are hired on to perform specific tasks. They take breaks, they can quit, they also are like visitors in their happiness levels. Though the happiness factors of employees are much more simplistic in nature, they are also more complex than visitors. Staff can become unhappy from simply being overworked and not being paid enough to compensate for this. They can also become annoyed if there is too much of one task to do in their area.

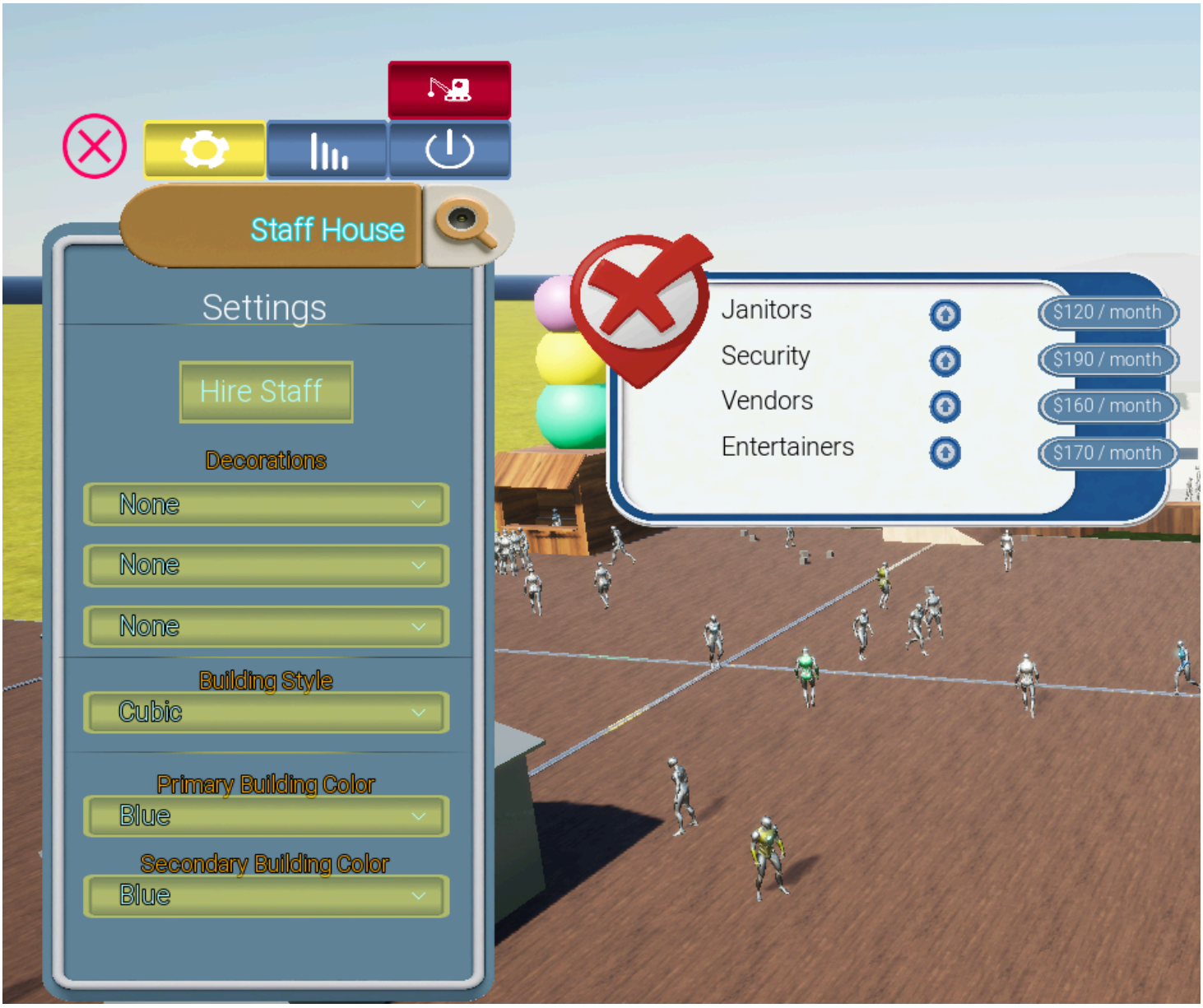
Like visitors, staff are given their own names, and randomized stat thresholds. They also have an unique name tag with them giving access to the new staff features. To access the staff, the player must build a staff house first to be able to hire them.

Staff can be provided extra training, which can offer some great benefits. Staff that get extra training work harder, longer and do not get as easily frustrated with the amount of work to perform. Higher levels of training also increases the amount of stressors they can manage. However, trained staff do cost more.

Staff can be provided raises. These raises are similar to the training, however it does not give the employee the benefits that extra training provides. Staff given raises will bring all their stats back to 100%. It is a good preventative maintenance method to help keep them around. Giving a raise also increases the number of stressors that the employee can handle.

Breaks are also another good method to increase staff happiness. They automatically take breaks based on how many tasks they perform. This can be increased or decreased in the behavior tree.

Currently (as of V1.5), Vendors, Janitors and Entertainers are all functioning as expected. When hired, they will begin to work right away, finding the next available building or task to perform. Vendors can be assigned to buildings as a hired employee at that location, or they will automatically look for the next available location to work at. Entertainers will wander around the park, increasing visitors' happiness and excitement levels. The Janitors will look for garbage to pick up or garbage cans to empty. They can only carry so much at a time, and therefore require extra training to carry more and handle more before requiring a break.



Stressors

So, what's this "stressor" I mentioned? A stressor is any task that the employee sees within its sight range which is greater than 1. This means that if an employee sees 2 piles of garbage in its view or more it will increase that staff members stress. Once the staff reaches the stressor threshold, it begins to reduce the staff member's job happiness. Raises increase the stressor limit, which helps negate this from being an issue and taking breaks reduces the stressors back to 0. Each Staff type has a different stressor threshold. For instance security can only take 5 stresses before being overwhelmed, while a janitor has a much higher stress limit of 50. In the current state, garbage is represented by small cubes. These can be replaced with whatever static mesh you wish. The mesh contains physics and therefore can be kicked about by guests.

Training

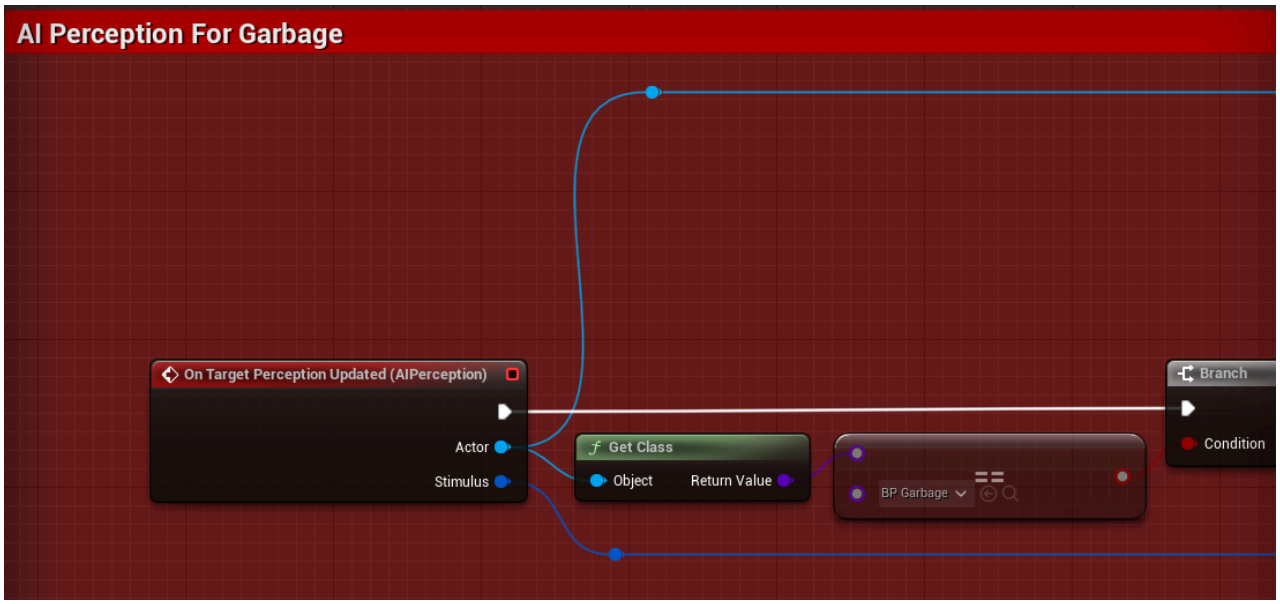
Training is accessed once an employee is hired. When the player clicks on the staff member, a name tag will pop up like it would with a visitor. In this case however, there is a button which will increase the training level up to 5 stars, maximum. Once training is provided, you cannot reduce it back down. The same is true with raises, seen here with the + icon button.

Breaks

Everyone loves breaks. But do you as a business owner? It's annoying but true. Everyone gets breaks, and are paid for it. So, this is no different. However, in this case the breaks are based solely on the number of tasks performed. If the employee performs no tasks, they take no breaks, which also means they are available to clean up any issue as soon as it happens. The stats for this is found inside the **BP_Staff** Statistics section.

Visitors

Visitor Perception



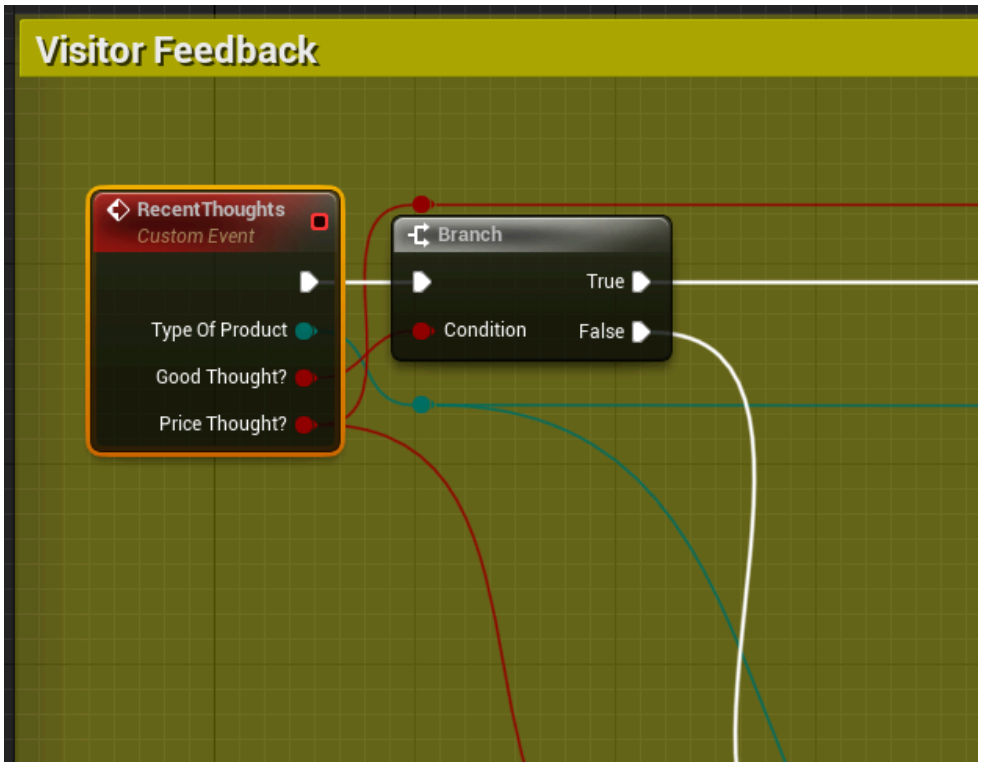
Visitor Perception is a new addition as of V1.2 which now adds a factor which will trigger a response by the visitors. In this case, garbage found on the ground will cause a negative response by visitors if they pass by it. The perception can be expanded upon for many different factors, such as crimes being witnessed, park business, etc. Inside the **BP_Visitor** the section named AI Perception for Garbage contains the class of object to be used for the garbage response. This can be changed to whichever type of class or object you are using for your garbage in game.

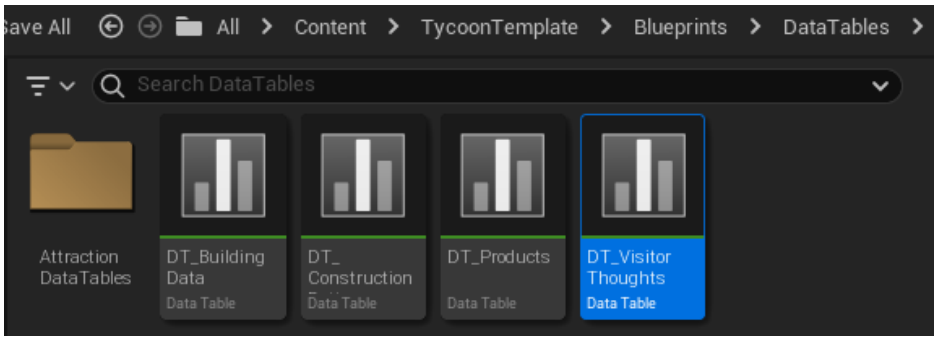
Banning Visitors

This is a small feature that will tie in together with the addition of functional Security. At this time, you can manually ban visitors from the park by clicking the shield icon on the visitor card. This will force them to go to the exit and leave.

Visitor Feedback

Visitor Feedback is another new feature as of V1.2. This feature gives feedback on the visitors' experience in the park, and will give you an idea on how to improve the park's ratings and situations. Inside the BP_Visitor section you'll find a Visitor Feedback area. This pulls information from a data table to determine the wording that will be displayed on the visitors name tag.





The DT_VisitorThoughts data table contains the reaction wording for each situation. In general there are four situations in which a visitor will react to situations. These include Good or Bad State Thoughts, and Good or Bad Price Thoughts.

Data Table × | Data Table Details

Search

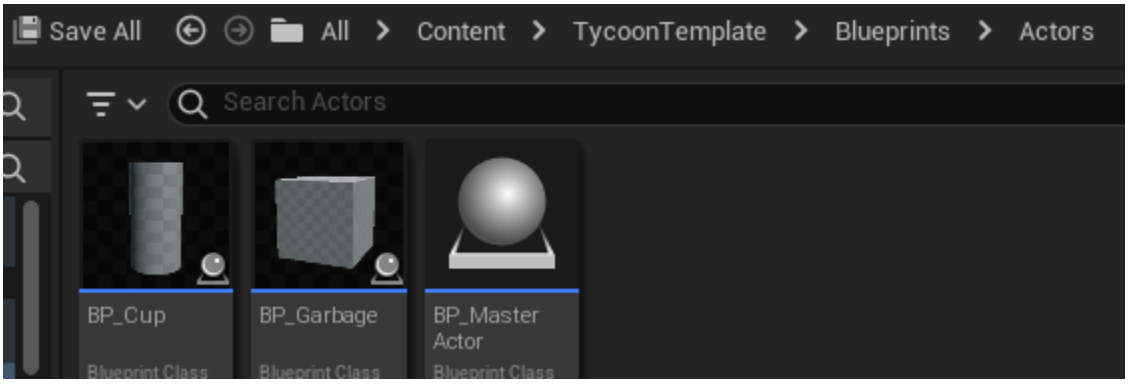
	Row Name	Building BP	GoodStateThought	BadStateThought	GoodPriceThought	BadPriceThought
1	Empty	None	None	None	None	None
2	Entrance	/Script/Engine.B	Nice staff at the park entrance	The staff were rude at the park entrance.	Cheap park pricing!	A bit overpriced for entry and then still charge for rides.
3	Toilet	/Script/Engine.B	The washrooms are nice and clean	Filthy, absolutely filthy bathrooms!	It's reasonable pricing at the washrooms	How dare they ask for money to use the washroom!
4	KioskFood	/Script/Engine.B	Great Canteen food, and quickly	Lineup is crazy for a Canteen	Great pricing at the Canteen	It's outrageous pricing for kiosk food!
5	KioskDrink	/Script/Engine.B	Great Canteen food, and quickly	Lineup is crazy for a Canteen	Great pricing at the Canteen	It's outrageous pricing for kiosk food!
6	GiftShop	/Script/Engine.B	Wasn't too crowded at the Gift Shop	Way too overcrowded at the Gift Shop	Reasonable pricing at the Gift Shop	It's very pricey at the Gift Shop!
7	Restaurant	/Script/Engine.B	The restaurant has the best food ever!	I think the food at Roadkill Joes was better	The price at the restaurant was very cheap	Wow! I don't think I'll be buying a meal at this restaurant
8	Cleanliness	/Script/Engine.B	The park is always kept clean!	What a filthy place, I will not be coming back!		

These values can be changed to whatever wording you wish. Currently, the state of buildings such as kiosk and gift shops do not function, only the pricing does. The state will be updated if a degradation system is in place for the buildings.

Filthy Visitors

Inside **BP_Visitor** there is a "Food Carry Time" variable. This determines how long a visitor will carry food bought from the Kiosk before they need to toss it out. Once this time reaches 0 they will look for a garbage can within a specified radius (Specified in the behavior tree). If there is no can found within this radius, the visitor will throw the food on the ground, creating a garbage actor, and therefore work for the Janitor to handle. If there is a garbage can nearby, they will throw it into the garbage can instead.

Obviously, like in real life, garbage cans can only fit so much trash before they overflow. Well, that is exactly what happens here. Each garbage can has its limit on how much garbage it can handle. Currently, this is set to 4. After that, the visitor will try to put it into the can, but it'll fall out creating garbage on the ground. So, where do you find this garbage actor it creates?



You will now find a new folder called "Actors" These are designed to be anything that is not an interactable building or object type class. In other words, you cannot click on

them and remove them manually, build them, etc. They are spawned automatically, and are used by the system to create Stimuli or interaction zones for the Staff/Visitors to interact with.

In this image, you can see that the visitor is now carrying a cup actor. This is spawned when purchasing a drink from the kiosk. They will carry it around until they are done with it and then toss it out.

The calculation for the cleanliness gauge on visitors can be altered, just like every other one inside the **BP_Visitor.** Note that like before, only visitors have these features. Children and spouses do not have these functions fully working at the moment.

Wealthy and Average Visitors

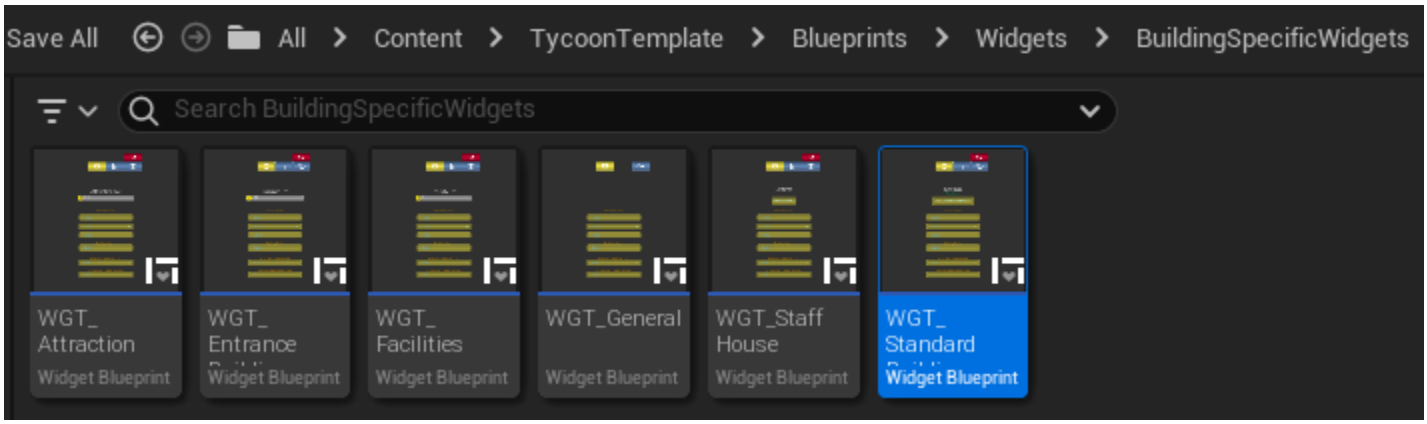
Along with the normal visitors there are wealthy visitors who will show up. They have a higher threshold for price markups. In the normal visitor price range an acceptance of 200% to 300% price increase is allowed (aka double the base cost to 3x the base cost). However, there are some people who don't care as much about the pricing and there "rich" people have a higher acceptance of 300% to 500% price markup. This will bring in higher sales, but less often. These ranges can be changed and more added in the **BP_Visitor** construction script.

Custom Building Data Table

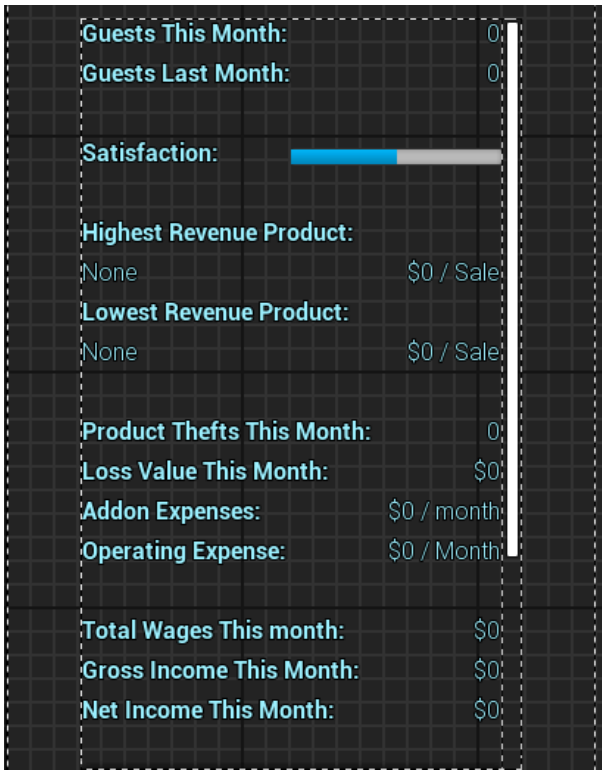
So, why have this? This is a better way to be able to have your own custom buildings added and not have them affected by any future updates that occur. The custom data table is found in the same directory as the main building data table. If all the data is put into this table for your custom buildings, there will be no issue replacing the table with content you previously created, saving you the effort of having to re-enter data.

Statistics

As of V1.3, the statistics tab offered for buildings such as the entrance, kiosk, restaurant, facilities, and staff houses, now have a variety of statistics which will display the current sales, employee wages, expenses, and other information that can be useful in managing the park. These updates can all be found in the building specific widgets.



Below is an example of one of the statistics offered in this set. The variables and stats you wish to display can be changed to anything you want. You will just need to change the variables which it pulls from inside these widgets.



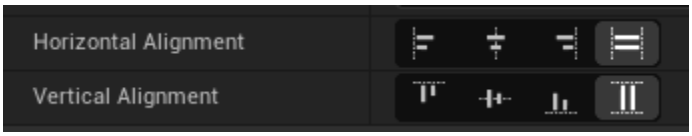
Current Limitations and Future Additions to the Statistics

The statistics system only currently defaultly supports the statistics which are provided by it. Any further statistics which the user may want would have to be added by that user, and the logic applied to it and the widget. Currently, the "Vandalism" and "Theft" stats are not functional as they will be provided once the Security staff functionality is added.

Troubleshooting Solutions:

Widget is not showing up correctly

If you are at this point and wonder why your widget isn't showing up correctly. Or if it looks something like this. This is a very simple fix. This is caused because the new widget has not been set to fill the encompassing parent widgets size.



By setting the alignment to fill on both vertical and horizontal, you will notice the widget fills the entire area.

The new widget's product list doesn't stay open when clicking the Stay Open button.

This is caused because when hooking up your new widget you missed one important node. The Auto populate function as seen in the standard building widget.

Product List does not fill in

This problem is usually caused because the building name does not match with the products building in the data tables. Make sure the spelling is the same on both.

Visitors Walk in the Park and turn around and leave

Check the visitor thresholds and starting stats set in the visitor construction script. Make sure none of the base stats are below the threshold.

Visitors Aren't Showing Up

This is actually built in by design. Once you place the entrance down, no visitors will show up until you open the park. Clicking on the park entrance and clicking on the power tab, you will find the Open/Close button. This will allow people to show up or not. When the park is closed, no visitors will show, however, you will still pay the maintenance fee for every building present. You can change this feature however you like as well. It will require a little coding on your end to do this.

Visitors Won't Move

There can be a few reasons for this. The first and primary reason could be your nav mesh not being properly set up. Depending on if you will have visitors walking on ground or on platforms, you should adjust your nav mesh height to encapsulate only the area needed. If they are not moving, and you are not using the platforms, you may not have your nav mesh covering the ground. Same goes for if you are using platforms, and they run over the edge, make sure your nav mesh encapsulates the lowest point they will travel.

If this does not solve this. It may be an issue with your visitor AI script, or the buildings you made. Check over the document again and see if you made the buildings correctly, and that the AI script is similar/or the same as the initial project.

What's New For V1.5?:

Change Log V1.5

Version 1.5, the Vendor and visitor Overhaul, comes with many much needed changes. The addition and update of the queue and vendor systems bring a new level of complexity to the template.

- It now has fully functional Vendors which will automatically look for a work location to go to and improve work speeds. They can be assigned and unassigned from locations. Each worker at that location will reduce the queue/visitor purchase order time based on their training level.
- Entertainers will wander around the park increasing happiness and excitement for visitors nearby.
- Visitors stay close to the buildings and structures of the park so they do not wander off too far from the core of the park.
- Attractions can now have queues as well.
- Buildings can be renamed to anything you want.
- Visitors can be banned from the park.

Change Log V1.4

Version 1.4, the Queue, comes with a few updates to the core system, and inclusion of visitor queues for more orderly visitor purchases.

- Visitors now enter a lineup, which you design the standing locations on your own. Create a line in any way you want. Visitors will load back into the lineup on save and load of game.
- The game now remembers the previous state of the visitor's behaviour tree and loads the variables back.
- Power status of buildings will now produce an icon overtop of them to indicate if it is "closed/unpowered" for a visual representation.
- Buildings now have an option to require a queue or not.

Change Log V1.3

Version 1.3, the Statistics Update, now adds in a number of typical statistics that are logical for games of this type of genre. This includes monthly wages, number of sales per item, monthly expenses, visitor counts, and comparative statistics to the previous month. This update is smaller in size, but a much needed update for a more rounded and complete product.

- Purchased items are kept, and displayed from the structure of items to be sold.
- Additional statistics for each building are created
- Selecting employees now allows you to find them

- Staff House now displays the highest paid employee and allows you to quickly find them.
- Staff House has a number of employee related statistics
- Restaurants, Kiosk, and Gift Shops all display top selling items, highest and lowest income producing products, and all expenses associated with operation of that building.
- Attractions now show various statistics and will drop the satisfaction rate if the pricing is too high.
- Edge Screen Scrolling now added.
- Bug Fix: Child and Spouse Leader Role from loading saved game
- Framework for Employee expansions are set in place.
- Buildings' basic settings are now set to landscape detection instead of platform detection, but allows both, and can still be customized.

Change Log V1.2

Version 1.2, also known as the visitor overhaul, brings in a lot of changes to visitor interactions, new staff features, and a variety of other fixes and updated areas.

- Visitor Features:
    - Visitors have a pricing markup threshold, with some “rich visitors” purchasing expensive ranged items
    - Visitors produce garbage if they buy from kiosks. Restaurants do not produce garbage in visitor hands.
    - Visitors will throw garbage if no garbage cans are available.
    - Visitors react to garbage on the ground, reducing the park rating
    - Visitors have thoughts on the park (ie: the park is kept clean, prices are too high, it’s cheap meals here)
    - NPC animation method updated to use the ABP for them instead.
    - Visitors carry objects, such as garbage (aka drinks (currently set up for this only) but can be switched to add additional)
- Garbage cans can overflow.
- Staff Features
    - Staff can now be hired/fired into one of four different roles. Janitor, Security, Entertainer, and Vendor. (Note: Currently only Janitor has tasks performed)
    - Unhappy staff will quit their job.
    - Janitors will collect one piece of garbage, and then be required to return it to a new structure “garbage shed”.
    - Staff will take breaks on a frequency depending on their training level and workload. More tasks performed, means more frequent breaks and higher stress.
    - Can give staff members a raise to help keep them happy and work harder.
    - Staff training helps to keep them happy and more effective but costs more.
- BUG FIX: People will no longer sit in the air.

- Wages are included in the monthly expense deduction.
- Save system updated to include all these features
- Begun work on statistic displays (ie: visitor numbers, expenses, revenue, etc)
- New buildings functions available for placements.
- Custom Building Data Table.

Change Log V1.1

Version 1.1 has a plethora of new features, functions and updates to code in general.

- The control system now utilizes the enhanced inputs of UE5 to better allow for quick control changes.
- Path system added, and path meshes added.
    - Visitors prefer to follow paths, but will not strictly adhere to them. This allows them to have a little bit of perceived freedom.
- A full multi file Save/Load system, with no restrictions on how many files can be saved/loaded.
    - Includes deleting from game
    - Includes date/time extension on save for reference (can be customized to however you would like)
- Visitors now have a new stat "energy". This gets refilled from sitting or having a meal. Long paths will tire visitors out more.
- The system now allows for fine placement/removal of snap to grid features by holding Ctrl down while placing an actor.
- No need for temporary meshes of larger size anymore to detect collisions for small meshes.
- BIG changes to the placement efficiency and method
    - Now each building has its own quick changed tracing method to determine where or how it can be placed built into the data table. This removes the need for adding this by "tags" for traces for specific models.
- Third-Person (personal character) mode.
    - This allows the player to take control of their own character in third person.
    - This mode still allows for placement of objects and changing options to them while in this mode
- Changes in code to relocate much of the controller code into its specific pawns control code.

If none of these troubleshooting solves your issue, send a message in the **Odium Studios** discord help and I will help however I can.