

Doors Technical Design Documentation

Overview

- This Documentation Details the implementation of the **Door feature**. For more information on the Design please see the **Door Feature Design Document**.
- If you have any questions please speak to: **Konan M (Technical Designer)**.

Content

Doors Technical Design Documentation.....	1
 Overview.....	1
 Content.....	1
 User Stories.....	2
 Visual Brief.....	2
 Door Types.....	3
 Door Variants.....	3
 Functional Overview.....	4
 Gameplay Flow.....	6
 Door States.....	6
 Asset List.....	7
 Parameter Cheat Sheet.....	8
 Folder Location.....	8
 Door variant.....	9
 Designer Editable Variables.....	9
 Future Problems / Considerations.....	11

User Stories

These are the User Stories the implementation of this feature must address (taken from the **Door Feature Design Document**).

- As a player I must to be able to open doors to help me navigate through the space
- As a player I must to close doors behind me to control the space and escape enemies
- As a player I must be able to understand how to interact with doors
- As a player the door must give me appropriate feedback from interacting with it so I understand what to do next.

- As a Level Designer I need to be able to use doors to control the flow point between rooms
- As a level Designer I need to quickly iterate door states and feedback.
 - As a Level Designer I should easily be able to tweak the door feedback from player inputs.
- As a Designer I need the ability to have multiple visual variations of doors to dress, prop, and build environmental functional narrative.

Visual Brief

Gameplay and Technical Considerations for the visuals of the Doors. (update this with the previous document needs on visual description)

Visual Description	Concept Art
Doors should always be the same dimensions (See Door Types)	N/A
Doors should block the players vision	N/A
Doors should be able to pivot from hinges to replicate a real door	N/A
Doors are indestructible	N/A
Doors should be attached to walls and seamlessly blend in with the their environment	N/A

Door Types

The main types of doors that will be used in the game are:

Measurement reference: 1m = 100 Unreal Units

Door Type	Dimension	File Location
Single Door	2 (H) x 1 (W) m	Plugins\GameFeatures\ShooterCore\Content\Blueprint\Doors\Meshes

Door Variants

Doors will have different variants depending on the environment. This is purely for visual purposes to signify which part of the environment the player is in.

All Variants need to remain the same dimension so they can be switched out within the same Gameplay Object.

Door Variants	Description	Use Case	File Location
Basic Door	Doors with no notable visual features. Basic and can be used in various places.	Hallways, standard offices, cupboards,	Plugins\GameFeatures\ShooterCore\Content\Blueprint\Doors\Meshes
Reinforced Door	Door which looks sturdy and signifies whether it is Locked / Unlocked (would need material swap for this)	Lab rooms, High Security Offices	Plugins\GameFeatures\ShooterCore\Content\Blueprint\Doors\Meshes
Fancy Door	A salient door that looks more high class	Meeting rooms, Bedrooms, Ballroom,	Plugins\GameFeatures\ShooterCore\Content\Blueprint\Doors\Meshes

Functional Overview

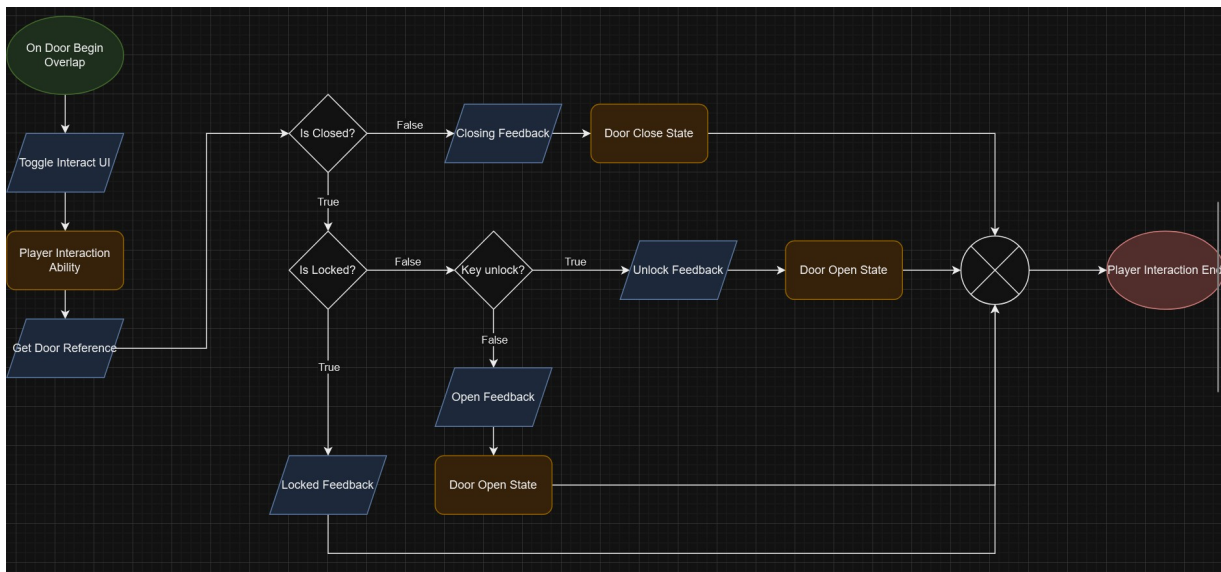
Priority	Function	Description	Completed
Must	Toggle Open / Closed States upon interaction	Doors should toggle from Open / Closed state when Interacted with <ul style="list-style-type: none"> • Gameplay ability interaction ability 	Completed
Must	Door Variants can be changed in Editor	Easily switch out Door Variant without having to change the actor. This should switch out: <ul style="list-style-type: none"> - Mesh - Audio - Material / Texture - Animations 	Completed
Should	Change direction doors open	Doors should be able to swing both ways to give Designers control on whether they should swing away from the player or be fixed to swing a certain position	In Progress
Could	Doors could have option to close on their own	Doors could slam behind a player to create tension / surprise. This slam time and speed should be adjustable for the Designers.	Backlog
Must	Door should be able to be Locked	When a door is locked it is a hard navigational obstacle. The player needs to trigger an event of sorts to open the door. <ul style="list-style-type: none"> • Pressure plate unlock placeholder • Doors have a boolean to toggle if it should be in a locked or unlocked state for designers to easily switch between the two. 	Completed
Must	Show interaction widget	Regardless of the door state, It must have feedback which shows if it's interactable <ul style="list-style-type: none"> • 3D Widget 	Completed
Must	Doors must be interactable close up	Players should have to walk close to the door to interact with it. <ul style="list-style-type: none"> • Line or sphere trace to detect door • Ability to trigger door functionality when trace registers a door. • Ability needs to identify the state of the door and trigger the right 	Complete

		sequence of events.	
Must	Doors must Feedback to a player when Locked	<p>Players should know when a door is locked, and therefore get audio feedback when interacted with.</p> <ul style="list-style-type: none"> • Line or sphere trace to detect door state. • Hit results provide necessary events for the door to trigger based on the state the door is in. 	Completed
Should	Some doors should have different visuals when Locked	<p>Depending on the environment Locked doors might have different visuals (e.g. a bathroom stall door or lab door)</p> <ul style="list-style-type: none"> • Toggle in the editor that disables the interactable door and uses the static variant instead. 	Backlog
Could	Doors could have option to close on their own	<p>Doors could slam behind a player to create tension / surprise</p> <ul style="list-style-type: none"> • Timer by event, If the door is in open state and not interacted with after X seconds, switch to closing state and play closing animation. 	Backlog
Should	Doors should stop during the opening and closing states if they collide with the player	<p>To prevent players getting stuck or weird collisions from happening, the doors should stop their animation when colliding with players during an opening or closing state.</p> <ul style="list-style-type: none"> • Placeholder Collision Overlap Event stopping animations. To start door animation, the player needs to interact with the door again. 	Completed
Should	Doors should have the option to start closed or open	<p>Provide designer exposed state events for door states. Door should be able to toggle between open and closed.</p> <ul style="list-style-type: none"> • Call in editor Door State change Event 	Completed

Gameplay Flow

Auto closing doors on flowchart?

[Flowchart Link](#)



Door States

Door Closed: This is an Interactable state where players can open the door manually.

Door Open: This is an Interactable state where players can close the door manually, or it will automatically close after the allocated time.

Door Opening: This is a non-interactable state. The player needs to wait for the animation to complete before proceeding for a new interaction.

Door Closing: This is a non-interactive state. The player needs to wait for the animation to complete before proceeding for a new interaction.

Door Idle: This state only toggles UI widgets. If a player overlaps its collision bounds the UI will toggle on/off with OnBegin and OnEnd overlaps.

Door Locked: This is an interactable state. The player gets a locked feedback indication when interacting.

Door Unlocked: This is an interactable state. The player gets an unlock feedback indication when interacting.

Asset List

Asset	Description	Departments Involved	Delivered
Static Mesh	Static Mesh for the different door variants	Env Art	Delivered
Concept Art	art reference for environment artists to model from	Concept Art	Not Delivered
FX	Sound effects for different door variants	Sound Designer	Delivered

Parameter Cheat Sheet

These are options available for LDs to customize the door gameplay.

Folder Location

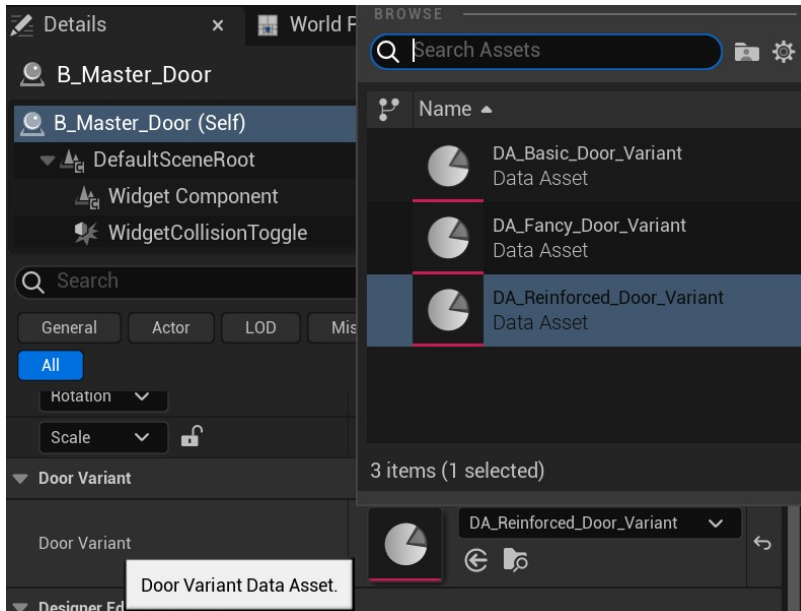
What and where	Image reference
<p>DataAsset: Create new door variants</p> <p>Location: Plugins\GameFeatures\ShooterCore\Content\Blueprint\Doors\DA</p>	 <p>DA_Basic_Door_Variant Data Asset</p>
<p>B_Master_Door: Modular door blueprint with multiple configurable settings.</p> <p>Location: Plugins\GameFeatures\ShooterCore\Content\Blueprint\Doors</p>	 <p>B_Master_Door Blueprint Class</p>
<p>B_DoorUnlock: Used to unlock and lock, locked door variants.</p> <p>Location: Plugins\GameFeatures\ShooterCore\Content\Blueprint\Doors</p>	 <p>B_DoorUnlock Blueprint Class</p>

Door variant

Easily add a door variant from a drop down menu with the pre-created variants.

- To create a new variant, duplicate the DA_Basic_Door_Variant and name it accordingly as DA_”variant name”_Door_Variant.

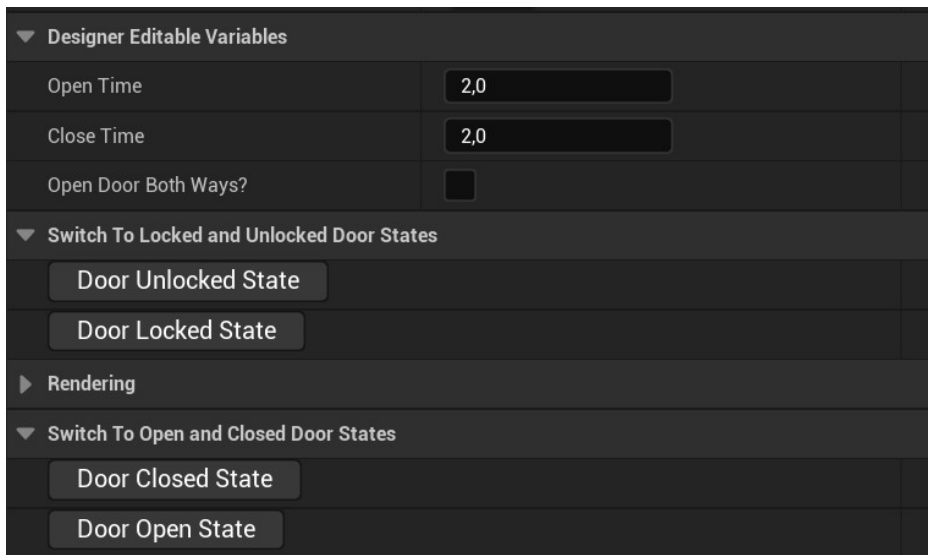
Now drag and drop the door blueprint into the world and within the editor details you can find “Door Variant” that includes all the variants to pick and choose from the list of.. Variants.



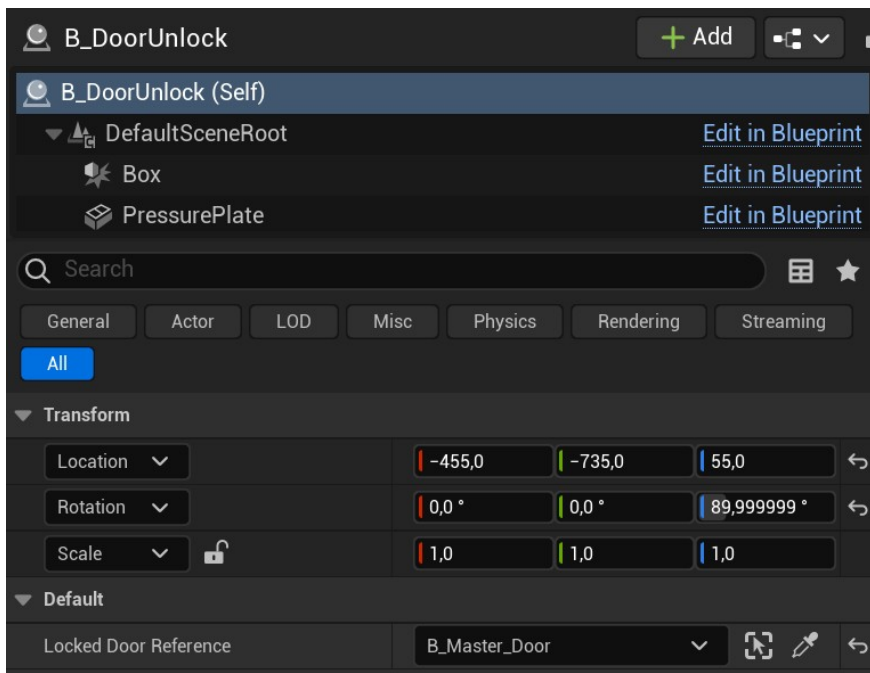
Designer Editable Variables

Easily tweak door gameplay functionality.

- Closing and opening speeds are in seconds.
 - It's a float variable. Both whole numbers and decimal values work, similar to the example...
 - Opening Time : 2,6
 - Closing Time: 5
- Open Door Both Ways? - Boolean
 - If true, doors can open both ways, away from where the player interacts with the door.
 - If false the door only opens in one direction, regardless of the player facing position.
- Switch To Locked and Unlocked Door States
 - Easily lock and unlock doors. The unlock state works similarly to the Closed State. Only difference with unlock state is the FX it plays.



- If a door is locked, the B_DoorUnlock is needed to unlock the door.
 - Make sure you are referencing the locked door actor within the designer config of B_DoorUnlock, "Locked Door Reference".
 - Player needs to walk on the pressure plate to toggle between locked and unlocked states for the specified door.



Future Problems / Considerations

Problem	Possible Solution	Disciplines Required	Comments
<p>Show interactable actors UI within trace range. Unreal's GAS requires a custom C++ interactable tick array.</p>	<p>The character can have an interactable trace running on tick that updates the array if an interactable actor is hit, otherwise an empty array if not within trace range. The trace triggers a 3D Widget on the actor if the actor is found and gets the actor reference as well. Once we trigger the interaction we already have all the necessary references needed.</p>	<p>C++ Programmer.</p>	<p>Workaround: Interactable actor has a begin and end overlap that toggles the 3D Widget. The Player Trace hit results only occur when the interaction button is pressed (not on tick), It only gets the actor reference (no widget toggle).</p> <p>The workaround works fine for the door but if we intend to use UI popup on other interactable actors within range, this feature needs to be fixed asap to support new interactable actors for future sprints.</p>
<p>No key item/pickup to use on the door.</p> <p>Workaround: Pressure plate that simulates the results and communicates with a specific door the designer needs to be locked.</p>	<p>Create an inventory system that supports the item pickups in the game.</p> <p>Must register item pickup and the stored pickup.</p> <p>Must be referenced with world interactables depending on item ability.</p> <p>Must be used/removed from inventory when used.</p>	<p>Technical UI Designer and C++ programmer.</p> <p>UI Designer?</p>	<p>The inventory system is another feature to consider apart from the door feature. Doing both at the same time is too difficult given the tight schedule. The inventory system may be viable to implement if we will reuse unlockable door items in the future.</p>

On door designer configurable state switches (locked, open state etc), the change is not saved when reloading the level.	Create a save game system that stores the door state changes in the level.	c++ programmer.	Unsure how the save game is handled, either within C++ code or blueprints.