Bullethell Elemental Template Package

As it is a project that uses external dependencies such as Firebase, it is necessary to follow the step by step correctly, it is not difficult but requires attention to avoid errors.

This is a project inspired by the Swarm mode in League of Legends, where the player has to survive until time runs out and eliminate the boss from the map. I added other modes, such as Survive until time run out, Survive until time run out and eliminate the final boss and a last mode which is Eliminate the boss and win, where the boss is spawned after a wave.

The Showcase Shader is the Unity Toon Shader, but since it is an experimental package I cannot include it in the project files, so I added the SimpleURPToonLit under MIT, but there is a bug that the shader developer has not yet resolved, which is that the model reflected in Render Textures becomes invisible, affecting the project's Ranking menu. You can add the Unity Toon Shader yourself by activating the experimental packages in the project and downloading: **com.unity.toonshader**

To report bugs and improvements or something you think is essential to have in the project, send me a message on discord.

Do you want to add something different, like a battle pass or another system? You can make a paid request by contacting me on discord.

Discord: ikingalpha

Email: rafbizachi5@gmail.com

🚀 Quickstart

Setup and Run Instructions:

For developers who want a quick step-by-step guide and already know how to set up projects with Firebase:

1. Preferably use Unity Editor 2022.3.17f1 or newer.
2. Create a URP project and convert it to Android or IOS.
3. Install the Unity packages via PackageManager: Input System , AI Navigation, In APP Purchasing, Sprite 2D and TMP Text Pro.
4. Install the FirebaseFirestore_12.2.0 package or newer.
5. Install the FirebaseAuth_12.2.0 package or newer.
6. Install the BulletHell Elemental Template package.
7. If for some reason the Project Tags/Layers were not imported or you imported the Template into an existing project and chose not to import the Tags: Add Monster Tags to the monsters and bosses in the game, including the monster models from the Demo in BulletHellTemplate > Resources > Monsters and add the maps from the Scenes folder in Build Settings.
   The maps should be in the sequence Login / Home / other maps.
8. Set up the Project in the Firebase console, get the file generated at the end with the name google-services.json (without editing this name) and add it to the project in the assets folder.
9. Go to the Firebase console, set the "Authentication/Login Method" to accept email/password and anonymous login methods.
10. Set up the Firestore Database by creating a new collection called "Players".
11. Add a BattlePass collection outside of the Players collection, in the configuration window of this collection in Document ID put the SeasonInfo and just below add 2 fields one will be renamed Season with the type Number and value 1, and another field renamed as StartSeason with type timeStamp with the date and time that the first season begins.
12. Add the Basic Rules to the Database in the rules tab:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    match /Players/{userId}/{document=**} {

      allow read: if request.auth != null;

      allow write: if request.auth != null && request.auth.uid == userId;
    }

    match /BattlePass/{document=**} {
      allow read: if request.auth != null;
      allow write: if false;
    }
  }
}
```

For developers who have not yet set up a project with Firebase or who have any questions, below is a detailed step-by-step guide with images:

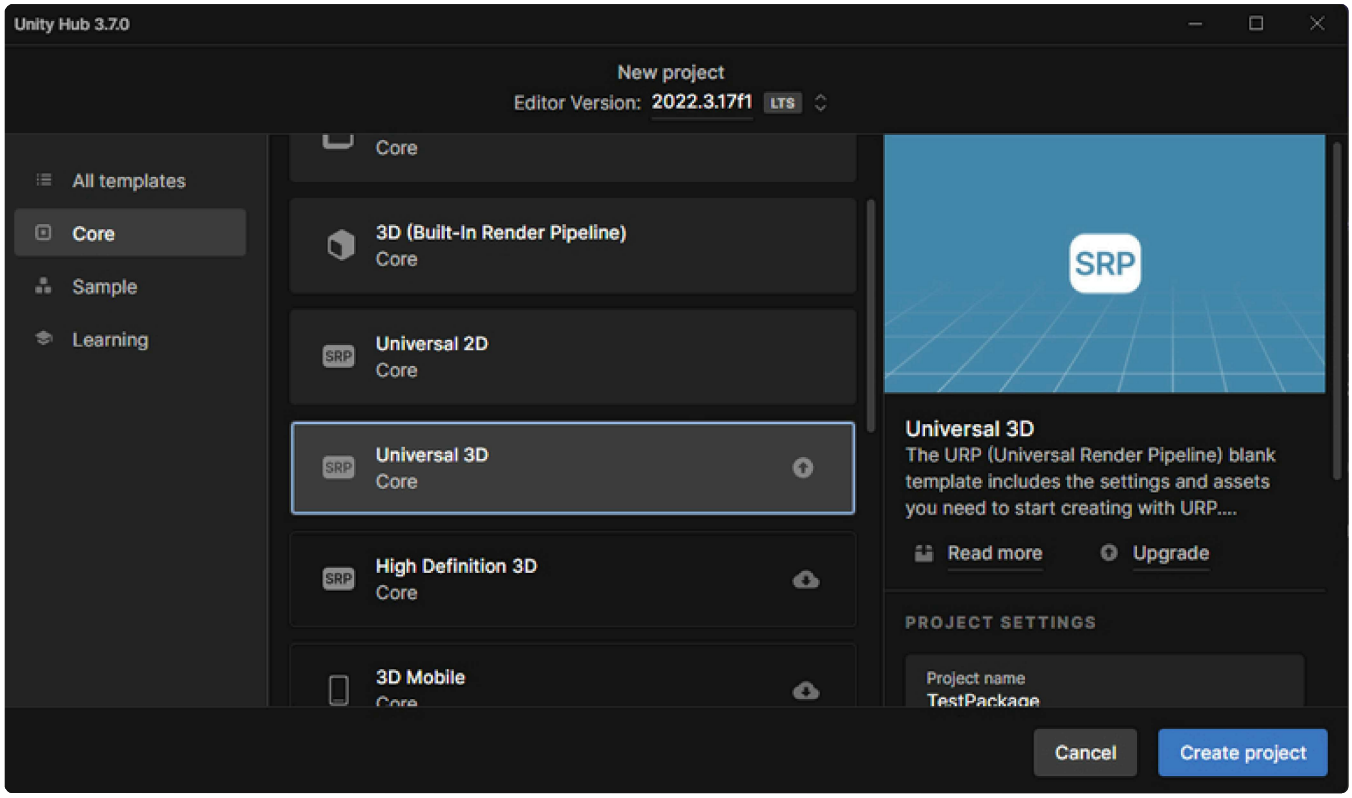
1. -Create a URP Project in the Unity Editor.

   -Preferably using Editor 2022.3.17f1

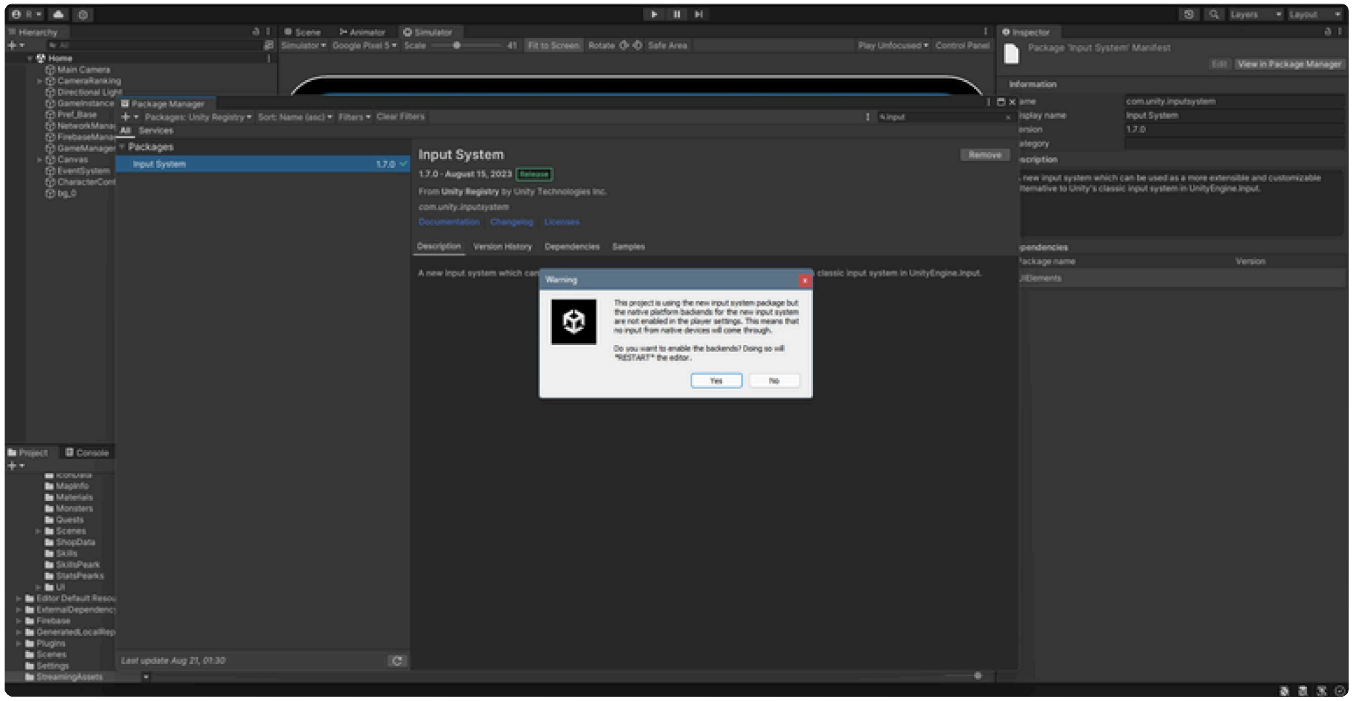
   -HDRP and integrated pipeline projects also work to create a game, but the demo textures must be converted or replaced.

   -Change the platform to Android or IOS (this requires that you have installed the Android and IOS Modules in the unity editor)

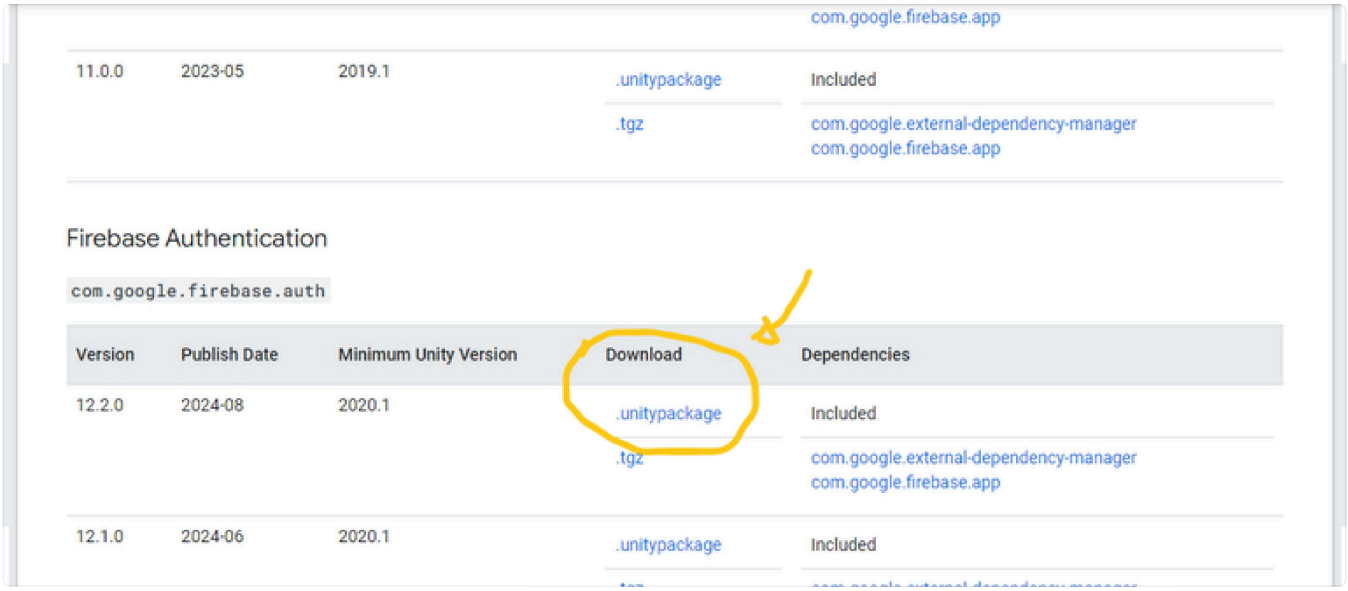
   -If you have not installed the modules, go to your UnityHub, click on the installs tab, click on the configuration gear of the editor you will use, click on Add Modules and then you can install the Android Build and its derivatives and IOS build and its derivatives, it is not necessary to add both but the firebase SDK will soon present an IOS or Android error and you will have to disable the "Validate References" of the corresponding dll.
   -If you have any questions, check the common errors section.



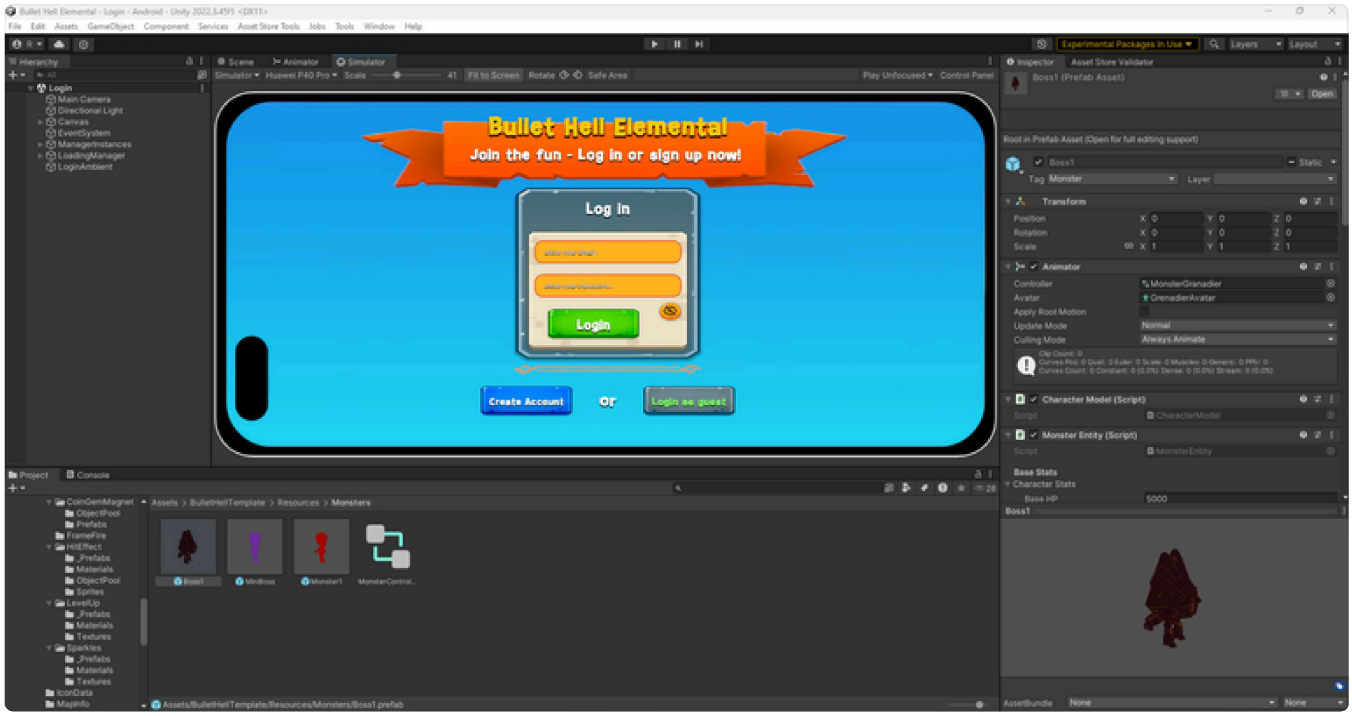
2. Install the Input System, AI Navigation, In APP Purchasing, Sprite 2D and TextMeshPro packages from PackageManager.
   Usually the project configuration file will install these dependencies automatically, but to be on the safe side, install them beforehand.

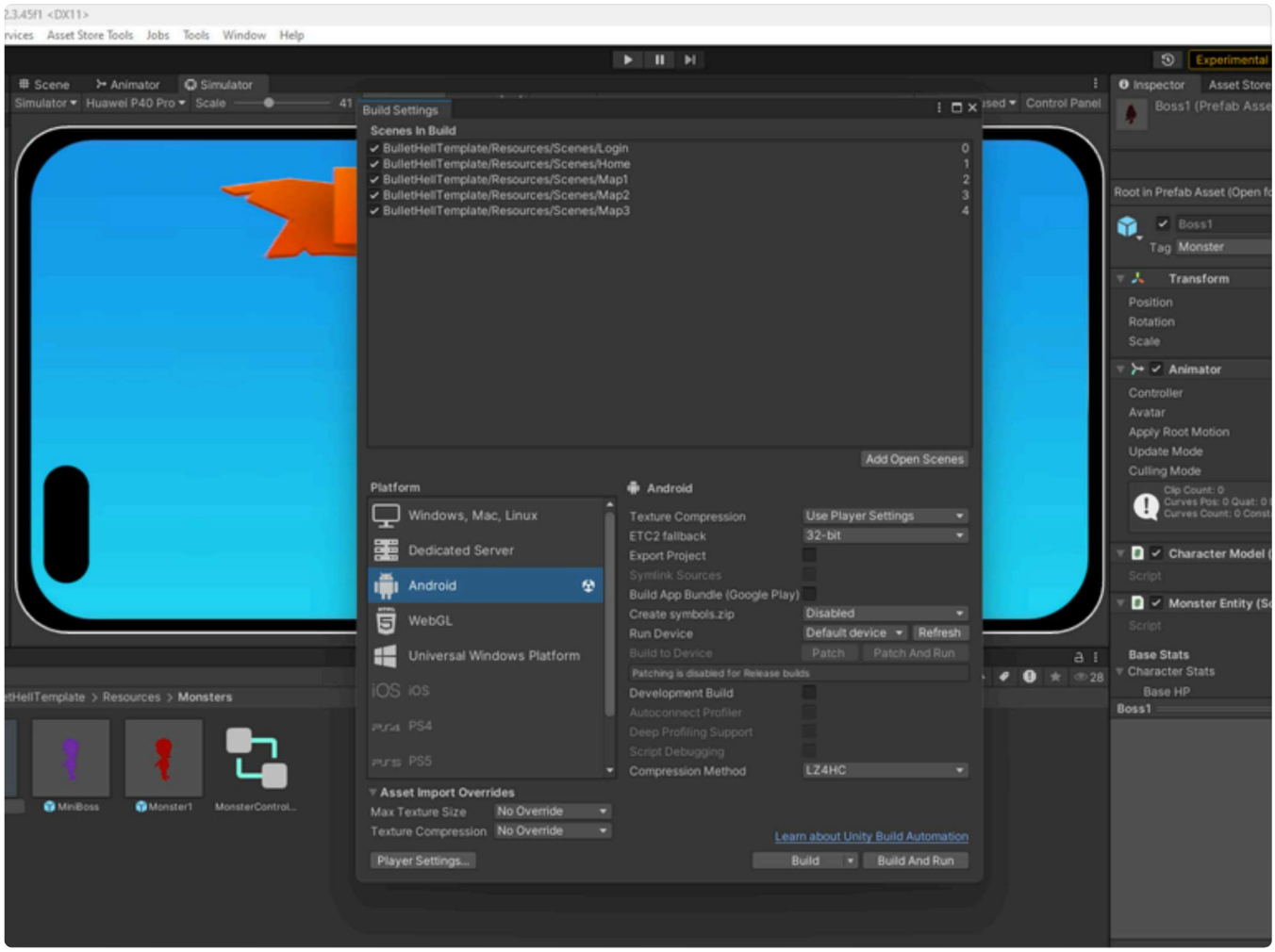


3. Install FirebaseFirestore_12.2.0 .
   Install FirebaseAuth_12.2.0 .
   Install Bullethell Elemental Template.
   Link: https://developers.google.com/unity/archive .
   Search for Firebase Authentication and Firebase Cloud Firestore for Firebase.

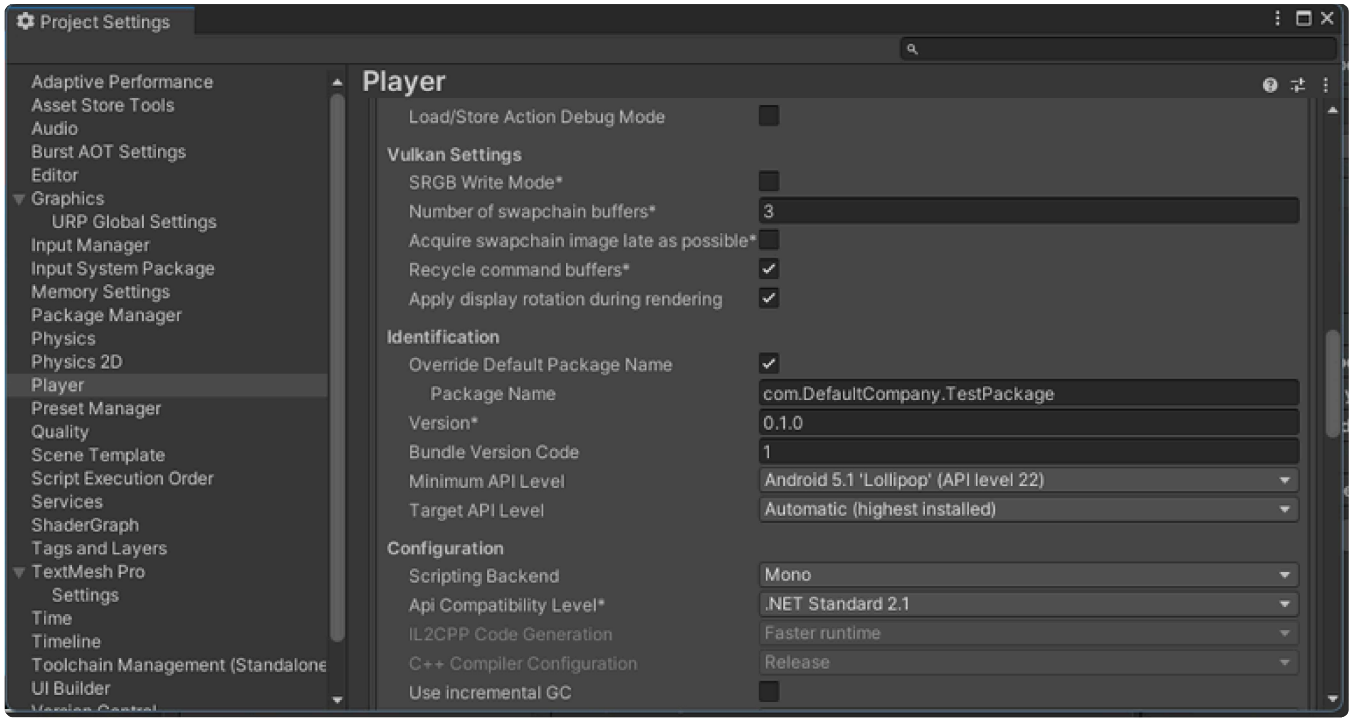


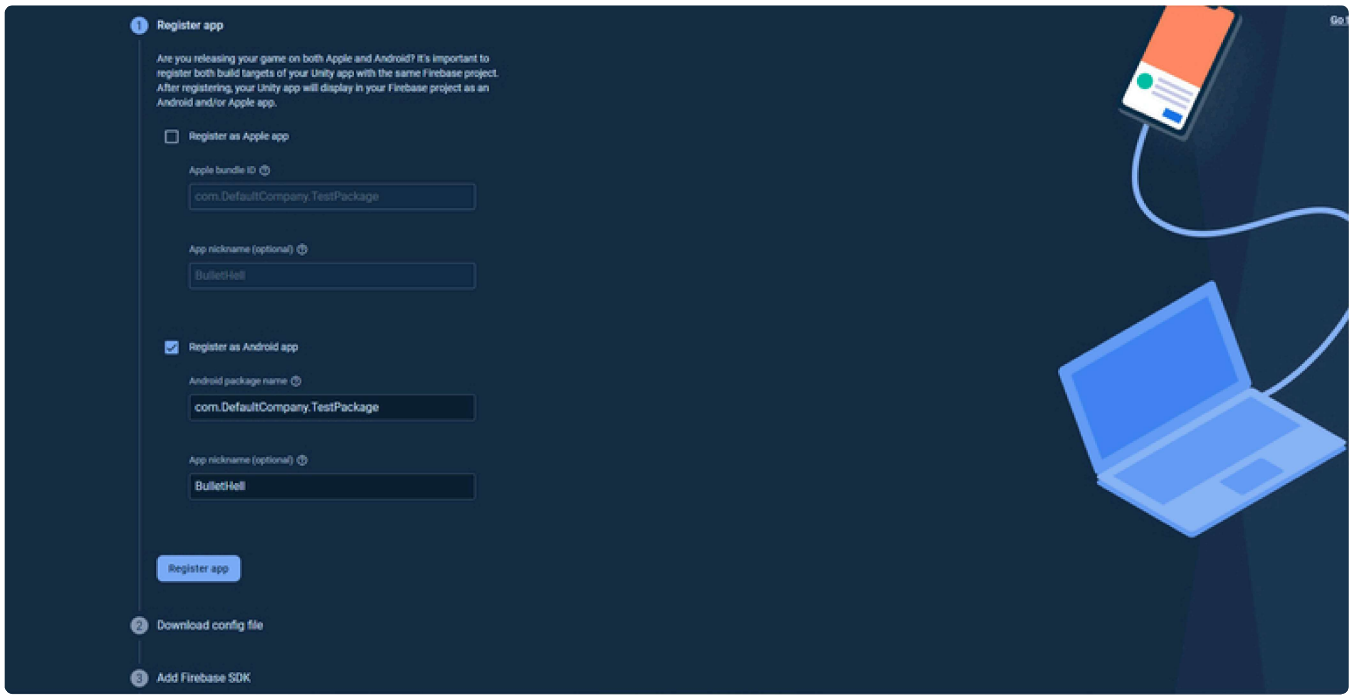
4. If you did not import the Tags/Layers and project settings:
   Add the Monster Tag to the project and apply it to the demo monsters and any monsters you create and add to the project.
   Add all maps from the Bullet Hell > Resources > Scenes folder
   The maps should be in the sequence Login / Home / other maps



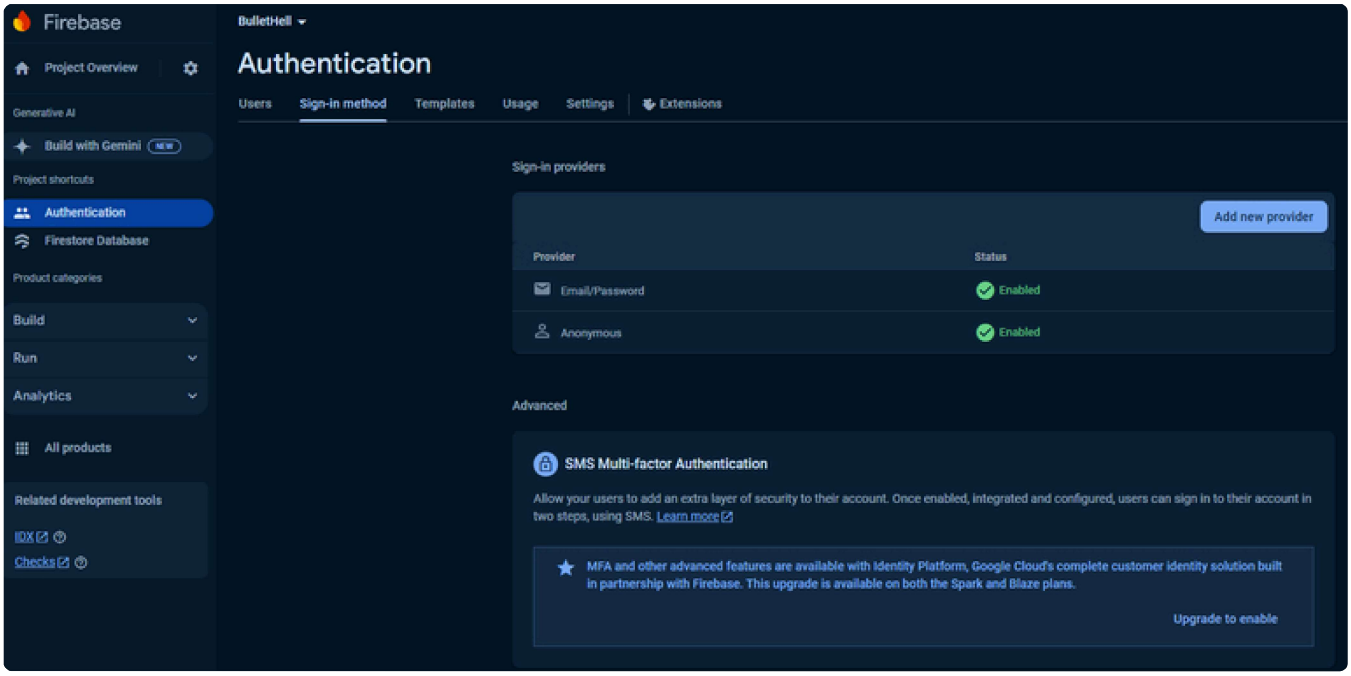


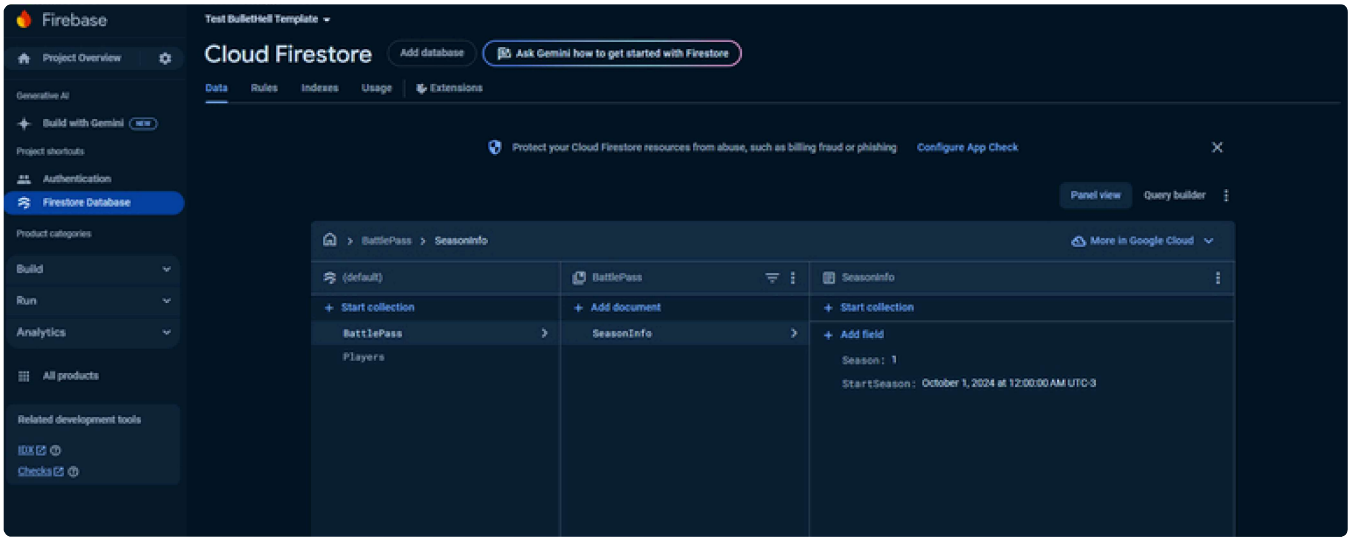
5. -Go to your account console on the Firebase website, create a new project and configure it. Remember to copy your project's identification name from Player Settings > Player > Other Settings > Identification > PackageName for Android and iOS to add it to your Firebase project configuration.
   -Grab the google-services.json file without changing its name, because if this is not the first time, it may be like "google-services.json(2)" for example and you will need to delete this (2) at the end and then place the file in the project's assets folder.
   -To be safe, create the StreamingAssets folder and place the google-services.json file inside it, because in some cases the Firebase SDK cannot find the file.





6. Now in the Firebase console, go to Authentication and add Email/Password and Anonymous as login methods.
   Next, go to the Firestore Database and create a new collection called Players and add an Initial document.
   Finally, for the Battle Pass to work correctly, add a BattlePass collection outside of the Players collection, in the configuration window of this collection in Document ID put the SeasonInfo and just below add 2 fields one will be renamed Season with the type Number and value 1, and another field renamed as StartSeason with type timeStamp with the date and time that the first season begins.
   Change the Season value when you want to start a new Season and when the players enter the game the pass will be reset.





7. In the Cloud Firestore menu, go to the Rules tab, copy and paste the basic security rules and click Publish. This is essential for players to be able to access the database and have the minimum level of security. Currently, the rules prevent disconnected users from accessing the database and allow the application to change only the document and the current user's information, but with free reading of the data so that the Ranking menus can work.

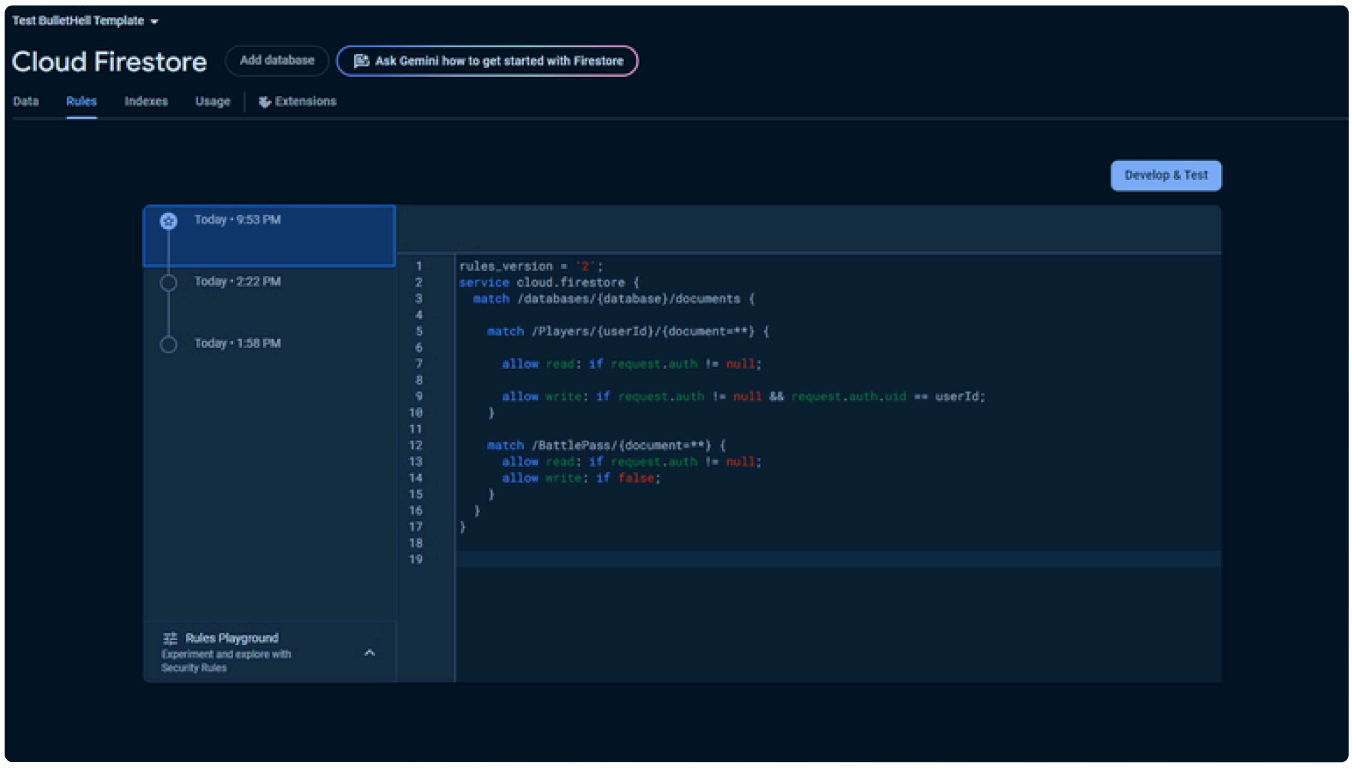
```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    match /Players/{userId}/{document=**} {

      allow read: if request.auth != null;

      allow write: if request.auth != null && request.auth.uid == userId;
    }

    match /BattlePass/{document=**} {
      allow read: if request.auth != null;
      allow write: if false;
    }
  }
}
```



With that, your project should now work, but if you have any issues, please contact us on our discord for support.
If you try to create an account using email/password for the first time and get an error message saying unknown error in the login menu, try to see if your password is strong enough for the authentication parameters (I recommend at least 8 characters with letters, numbers and one special character).
If the problem persists, try restarting the project

Note: you can add more security options, such as APP Check and other Firebase options.
The goal of the Template is to be practical and allow developers to create their games without upfront server expenses.
-For experienced developers, there is the possibility of moving the Load and Save logic completely to the server side using Firebase Cloud Functions, but this requires upgrading the Firebase plan and will also greatly increase the number of Reads and Writes on the server, also increasing the final expenses.
-In future updates I will implement an additional package that changes the database systems to CloudFunctions with its separate documentation, at no cost to those who already have Asset, just as an alternative for those who prefer to have more security in exchange for a higher server cost.

Previous
Bullethell Elemental Template Package

Next
Login Scene

Last updated 58 minutes ago

Login Scene

Quick explanation of the essential components of the Login scene.

AuthManager: Component responsible for login system and UI behavior.(It's in another GameObject in this scene)

FirebaseManager: Component responsible for loading and saving data in the firestore

GameInstance: Component that stores general game information in the Singleton pattern with characters, icons and more.

MonetizationManager: Component responsible for containing game currencies, receiving and spending coins, items, etc.

Battlepass Manager: Component responsible for configuring battle pass XP, season length, and battle pass value.

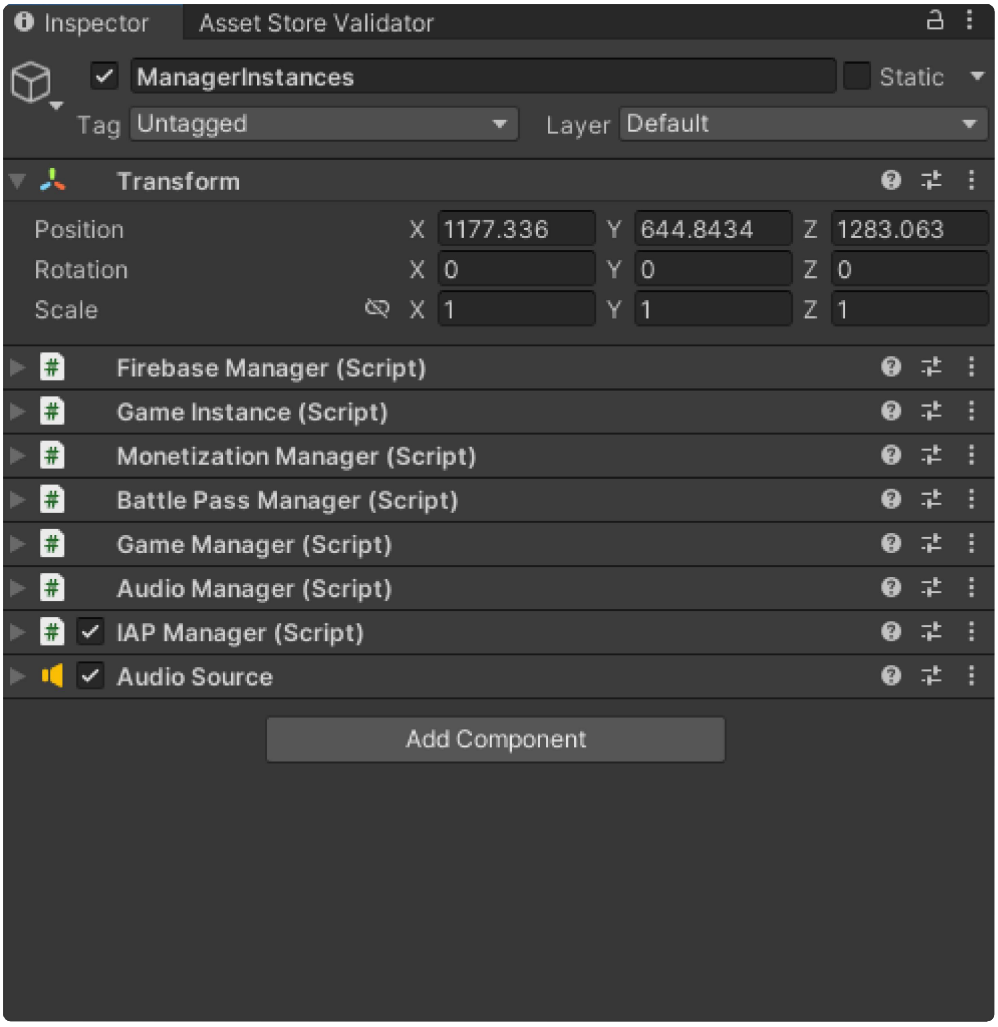
GameManager: Responsible for controlling the start and end of matches, as well as saving mission information and scores.

AudioManager: component responsible for managing sound volumes by tag, button and skill sounds, and limiting the number of sounds played for mobile performance purposes.

IAPManager: Simple shopping system using Unity's built-in IAP with configurable items to appear in the game UI, such as test and production modes (implemented but not 100% tested so I consider it in Beta test).

AudioSource: Unity Engine audio management component, reserved for audios with the Master tag.

LoadingManager: component responsible for the loading screen between scenes, with loading screen audio and animated texts(It's in another GameObject in this scene).

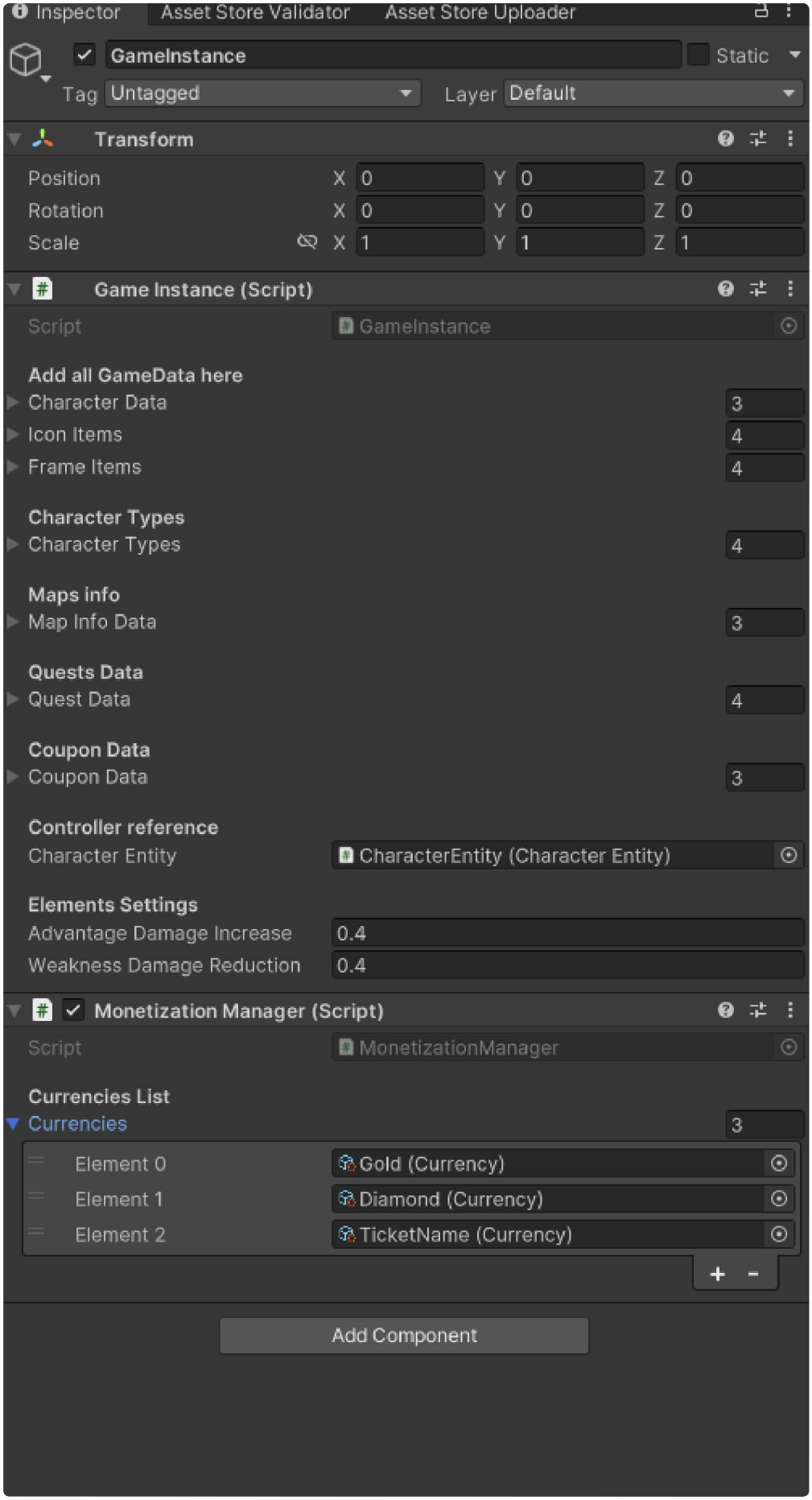


Previous
Quickstart

Next
GameInstance/Monetization/GameManager

Last updated 20 hours ago

GameInstance/Monetization/GameManager



CharacterData: Stores the character's scriptable Data, whenever you create a new CharacterData you need to add it here.

IconsItems : Stores the Scriptable data of the icons available in the game.

FrameItems: Stores the Scriptable data of the frames available in the game.

CharacterTypes: Store the Scriptable Data of Character Types, which are responsible for defining the weaknesses and advantages of game elements.

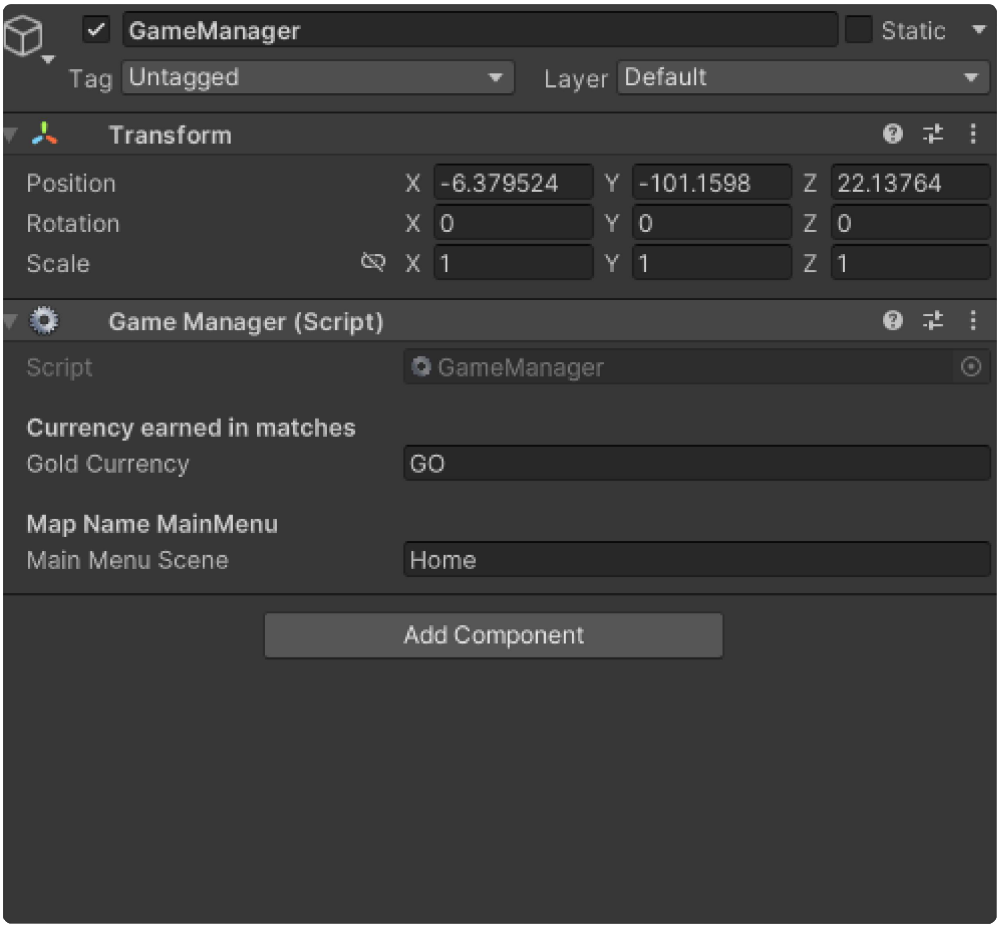
MapsInfo: Stores the ScriptableData of available game maps.

CouponData : Stores the ScriptableData of the coupons the player can redeem.

Controller reference: Set the default character controller.

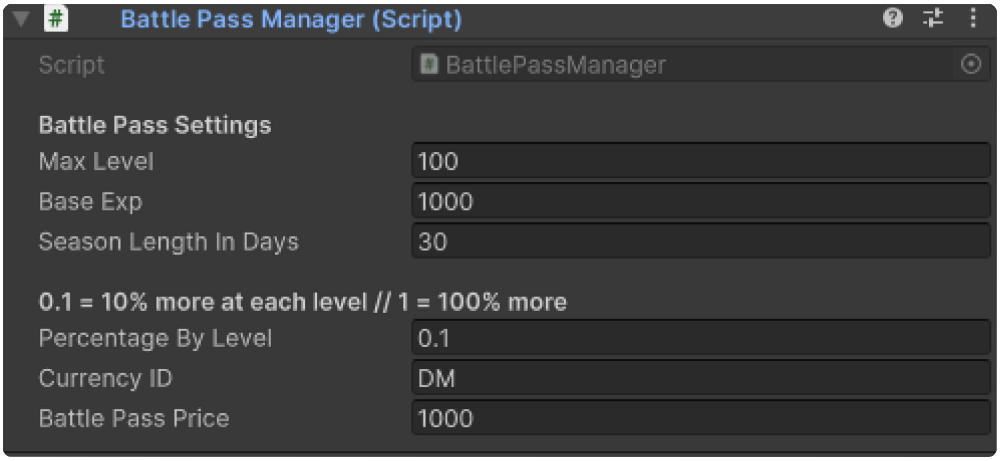
Elements settings : Defines the percentage of increase or reduction of the weaknesses and advantages of the elements.

MonetizationManager/Currencies : Stores the ScriptableData of in-game currencies.



Gold Currency: sets the ID of the currency the player receives when completing maps

MapName Main Menu: Sets the name of the main map the player will be taken to upon completing maps.



MaxLevel: Max Battle Pass Level, preferably use the last Battle Pass reward level.

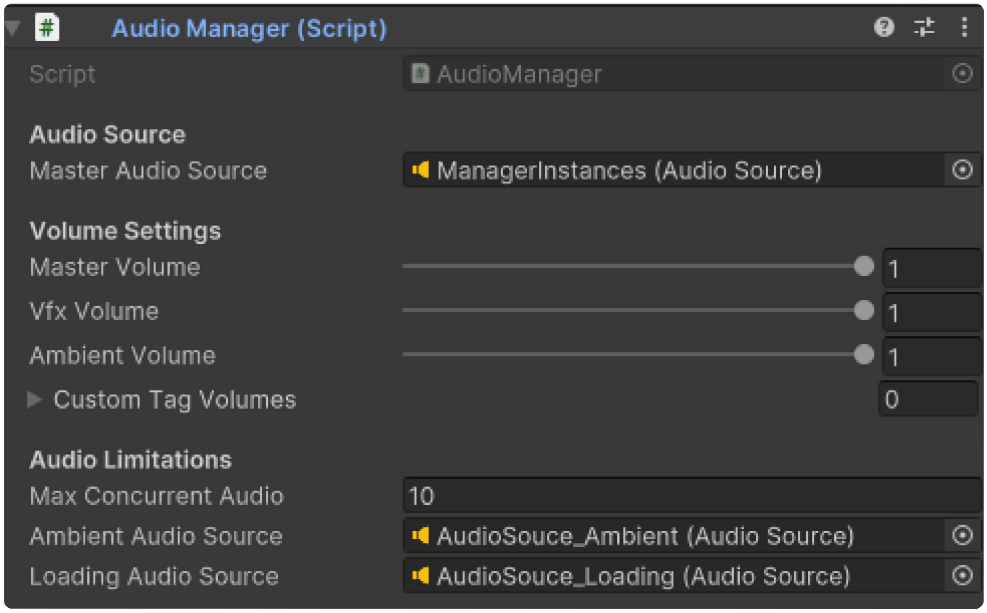
BaseEXP: Base EXP of the pass at level 1.

Season Length in Days: Duration of the current battle pass season, when starting a new season this number will be used by the system to indicate whether the current season is ongoing or has already ended.

Percentage by Level: Base XP percentage that will increase with each Battle Pass level.

Currency ID: ID of the currency used to purchase the pass.

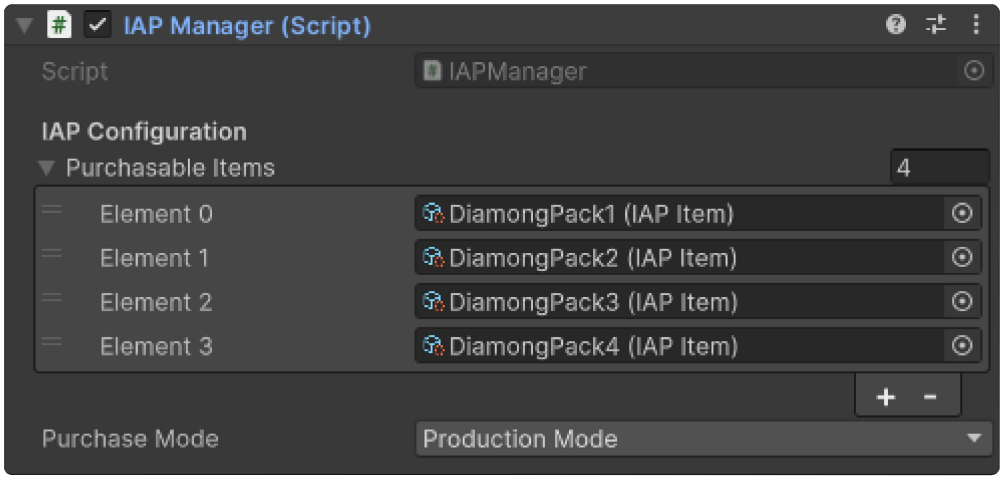
Battle pass price: Premium Battle Pass Price.



Master Audio Source: Audio source component responsible for Master, VFX and custom tags.

Volume Settings: Volumes of each tag or custom tag.

Max Concurrent Audio: Maximum number of audios played simultaneously for mobile performance purposes, leave it at 0 if you don't want limitations.



Purchasable Items: List of items purchased with real money (by default in dollars).

Purchase Mode: Select test mode (editor only) or production mode.

Previous
Login Scene

Next
Home Scene

Last updated 18 hours ago

Home Scene

Main Camera: Main scene camera.

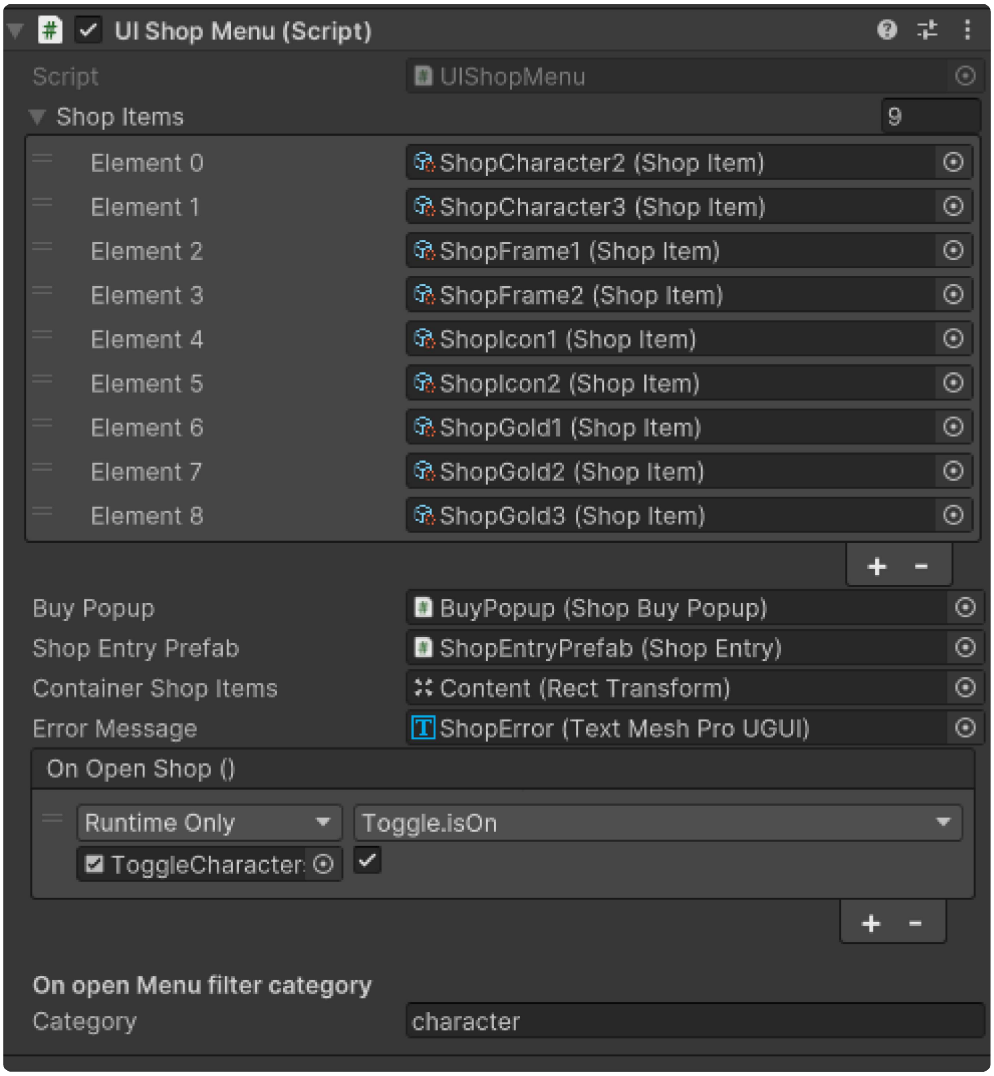
CameraRanking: Camera showing the characters spawned from the ranking selected in the ranking menu.

Rotate Character: Component that makes the selected character spin and do a flapping animation when clicked.

UI Main Menu: Component responsible for the main UI elements.

UI Character Menu: Component responsible for the main UI elements in the Characters menu, containing CharacterEntry that is created for each character unlocked on the account, prefabs for UI elements of skills, upgrades and stats.

UI Profile Menu: UI Components, UI Prefabs, and Nickname Change Settings.



UI Shop Menu: Component that has the Shop menu UI elements, UI prefabs and shop settings.

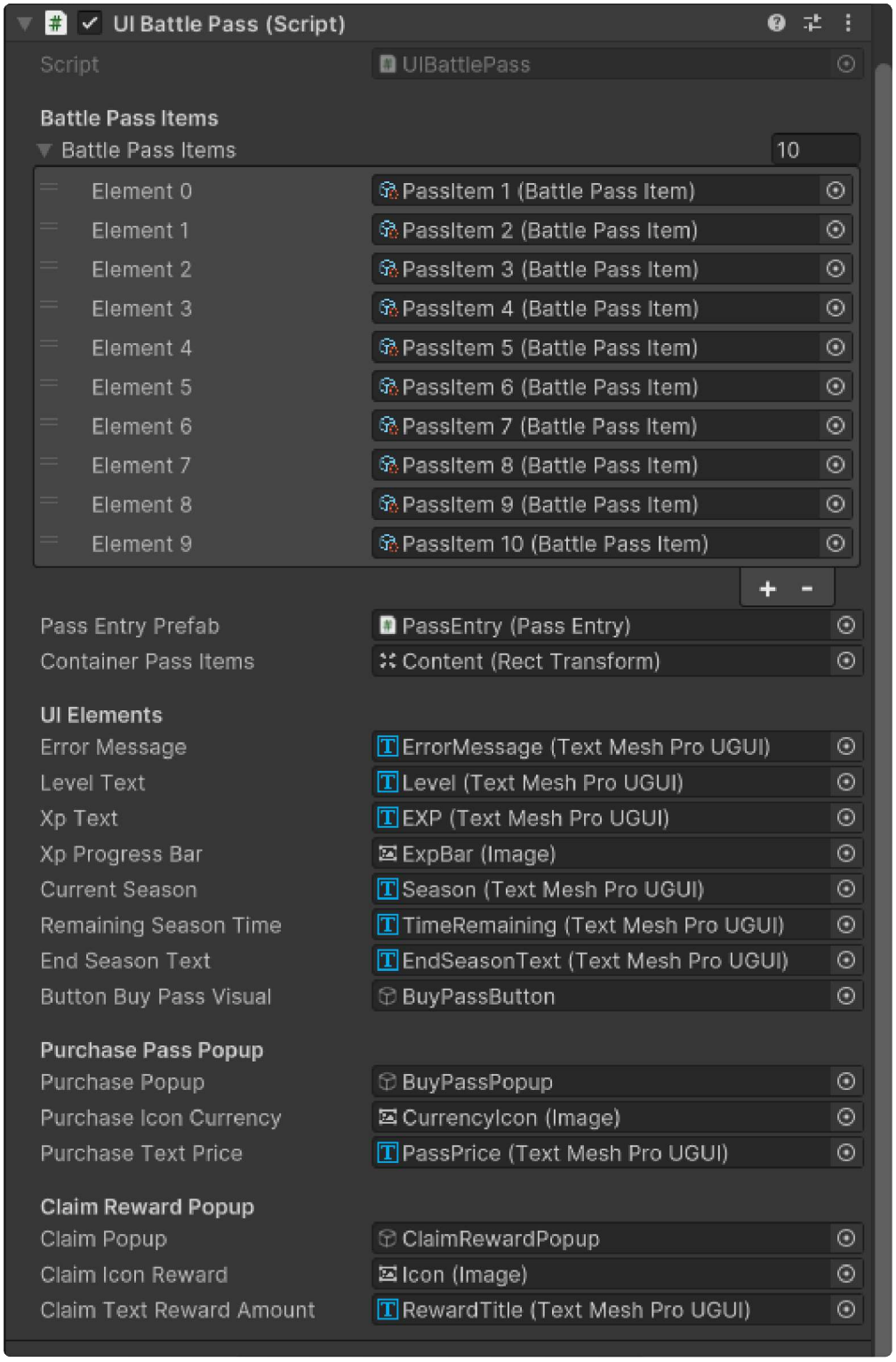
Shop Items: List of scriptable ShopItems that will populate the store menus.

UI Shop Menu/Category: category of items that will be initially filtered when opening shop.

UI Shop IAP: Component that holds the UI elements of the IAP store menu.

UI Maps Menu: Component that has prefabs and map menu UI configuration.

UI Quests Menu: Component that has prefabs and quest menu UI configuration and configuration to hide completed quests.



UI BattlePass: Component that contains Battle Pass menu UI prefabs and configuration, and a list of BattlePass Items scriptables.

UI Ranking Menu: Component that contains UI elements, UI prefabs .

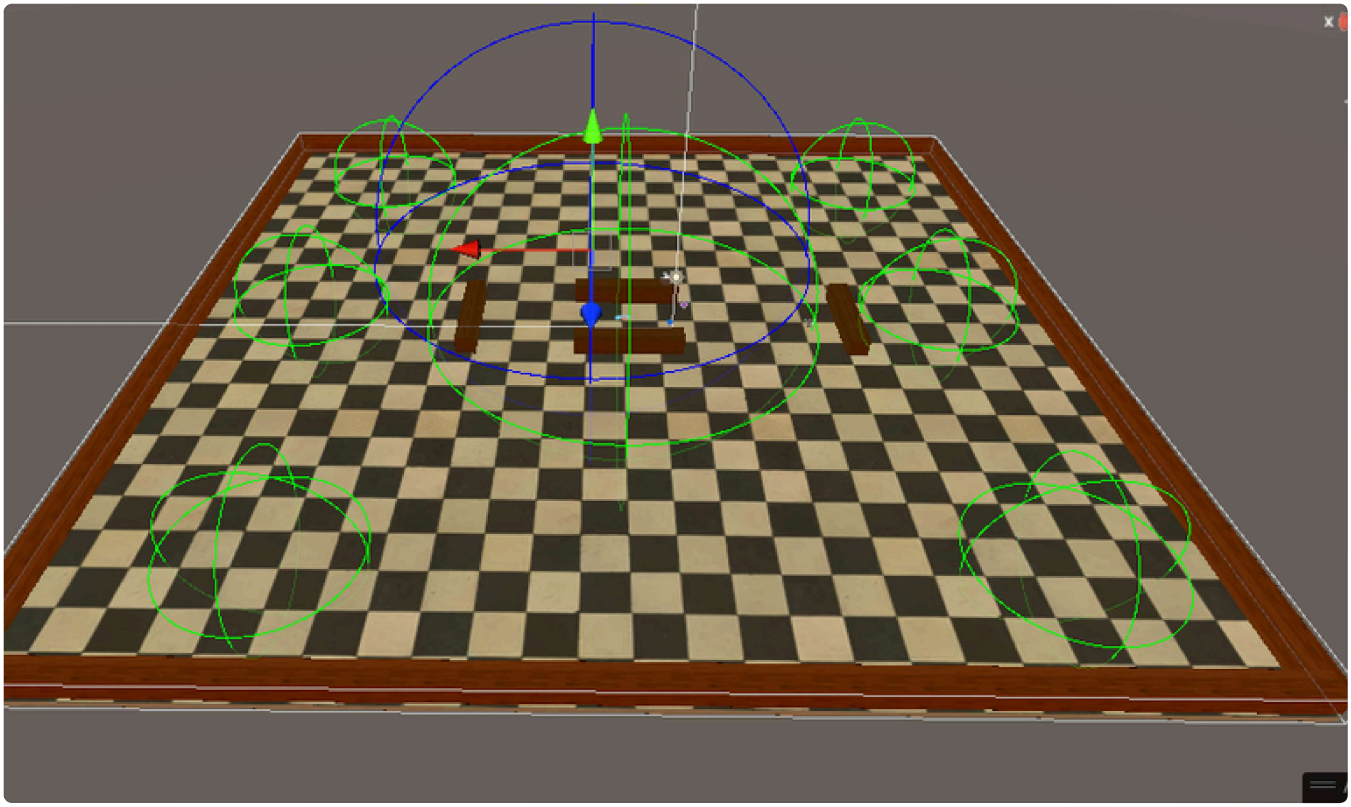
Previous
GameInstance/Monetization/GameManager

Next
Creating a new scene

Creating a new scene

Step by step to create a new map (scene)

Preferably duplicate an existing scene and edit it to make it easier, but it is also possible to create one completely from scratch with a few components.



1. Create your map with obstacles and everything, add the NavMeshSurface component and Bake it, configuring it as you prefer .
   The NavMeshSurface is essential for the functioning of Mobs and Bosses as they use NavMeshAgent to move around.
2. Change or add the UI Input System component from the Unity Engine New Input System package to a new object if it is not configured .
   This is essential for UI commands and clicks to work on this map.
3. Add the UIGameplay Prefab that is already ready inside the BulletHellTemplate/Resources/UI folder and just drag it to the Hierarchy window.
   This is essential so that the map has the UI with the joysticks and battle buttons, as this is not automatically instantiated like the player.
4. Create one or more objects with the SpawnPoints component, place them in strategic locations on the map where you want mobs or bosses to spawn, and the system will

randomly choose one of them to spawn.
Add Layer Wall on walls that you want spawning to ignore, to prevent mobs from spawning on or inside walls, and also add NavMeshObstacle on walls that don't adjust the navmesh to ignore them.

5. Add the GameplayManager component to an object and configure the match types, waves and bosses, match time, XP for each level, max stats and perks the player can choose (changing this may require changing the UIGameplay to receive the new stats, abilities and perks), add your SpawnPoints and configure the match time for Survival and Survival + Kill boss game cases.
6. Add the TopDownCameraController component to the scene's mainCamera and configure it.
7. Optional: You can add the BoxSpawner component to an object and configure the spawn radius and the BoxEntity that you want this manager to drop. You need to configure the boxes and create the BoxEntity prefabs with the power up and gold exp drops as desired. There are already BoxEntity prefabs created as an example in the Resources/Box folder.

See scenes Map1, Map2, and Map3 as examples to set up waves, spawns, and everything else in your scene.

You can use an automatic XP table generation tool:
Inside the current map Select Tools in Unity Engine
Select XP Progression Generator
Set starting and ending XP and amount of levels

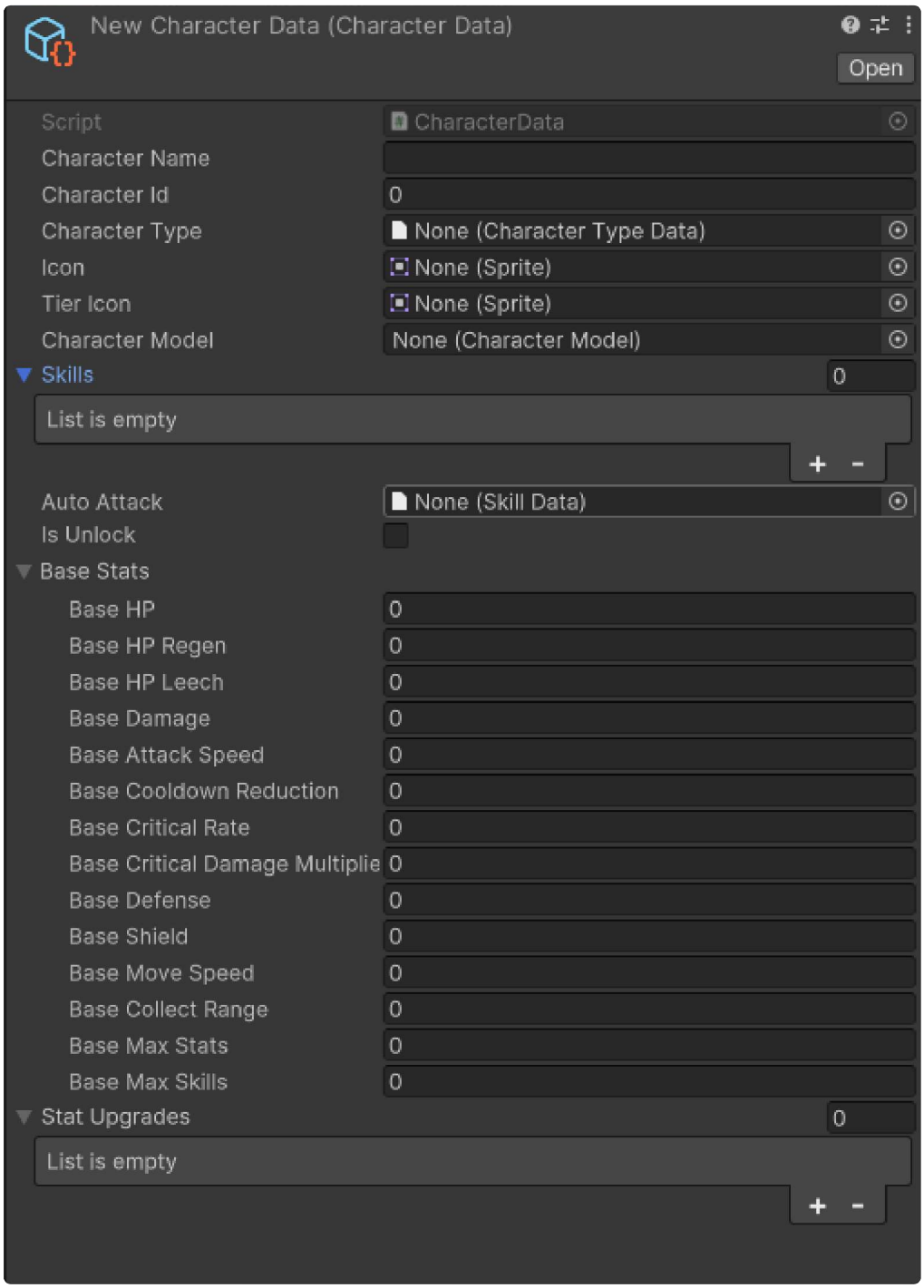
Previous
Home Scene

Next
Create new character

Create new character

To create a characterData, right-click inside any project folder, go to create and select CharacterData.

Remember that this item has to be added to the GameInstance in the Login Scene.



Character Name : Set the name of the character that will appear in the menus

CharacterId : Set a unique numeric ID for the character, the first character must have ID 0 (zero) and the isUnlocked box checked.

CharacterType: Main element of the character, defines reductions and increases in damage that the character will receive.

Icon: Character icon that appears in menus.

Tier Icon: Character Tier icon that appears in menus (visual only).

CharacterModel : Here you will add the configured character model (instructions are below)

Skills: Stores the skills usable by the attack joystick in the game (I recommend having 2 skills, otherwise you will need to edit the UIGameplay adding more joysticks).

Auto-Attack: Default character skill that is automatically used by the character and targets the nearest enemy.

IsUnlocked: Leave this box checked for characters that the player has already unlocked on the account (at least 1 character must have this box checked and the unique ID 0 (zero).

Base Stats: Default character stats before upgrades.

Stats Upgrades: Stores all Upgrades this character can have.

Statistics information:

Shield: amount of shield that will be removed before changing the character's HP (there is no visual part showing this yet, but it will be implemented in the future along with buffs for this).

Defense: Negates a fixed amount of any type of damage, but cannot be lower than the minimum damage declared in each map's GameplayManager.

HP leech: Drains part of the damage dealt as hp for the character.

HP Regen: amount of HP regenerated per second.

Cooldown Reduction: Reduces the cooldown of usable and auto skills (except auto attack).

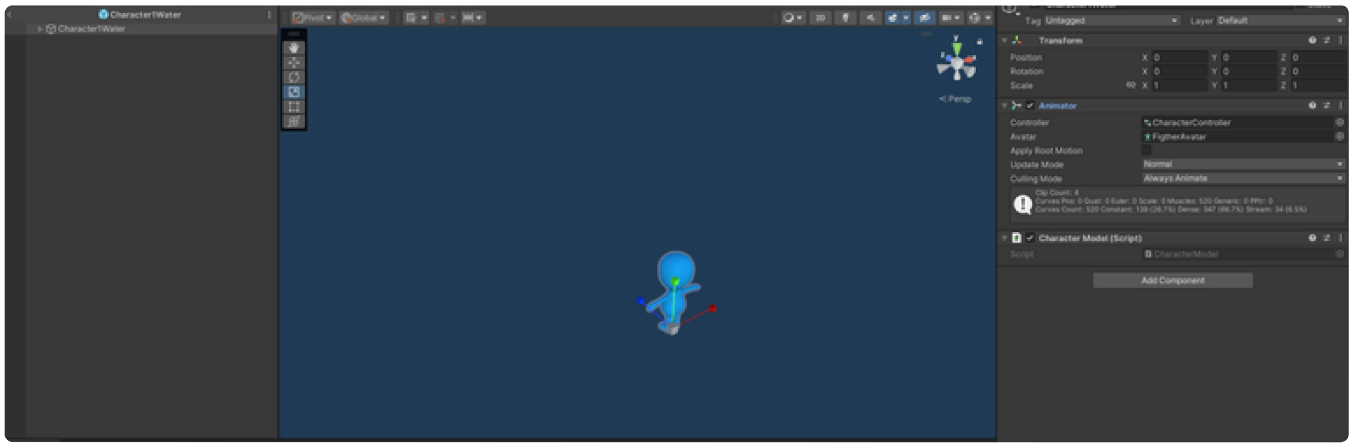
Attack Speed: Reduces auto-attack cooldown.

Collect Range: The distance the player needs these drops marked as auto-collect to collect.

MaxStats: Amounts of different StatsPerk that the player can choose (recommended not to exceed 5 counting baseMaxStats + Upgrades, otherwise it will be necessary to change the UIGameplay by adding more slots).

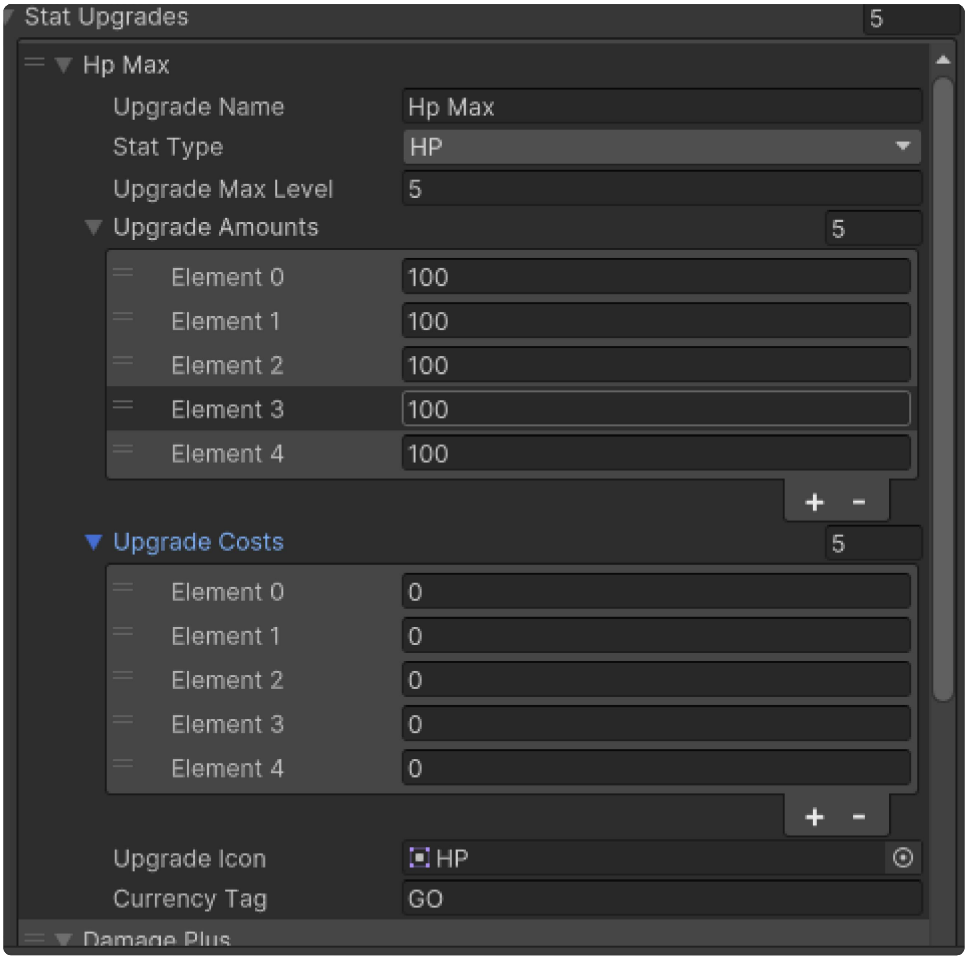
MaxSkills: Amounts of SkillPerk that the player can choose (it is recommended not to exceed 5 counting baseMaxSkills + Upgrades, otherwise it will be necessary to change the UIGameplay by adding more slots).

CharacterModel:



Setting up the CharacterModel is very simple: go to your 3D model and add characterModel and in the animator add AnimatorController = CharacterController and you can add this prefab to your CharacterData. The animations can be changed by opening the controller and changing the animations. You can also duplicate this AnimatorController and edit different animations for each character.

Stats Upgrades:



Upgrade Name: Name of the upgrade that appears in the menus.

Stat Type: Select the character stats that will increase.

Upgrade Max Level: Maximum level to upgrade.

Upgrade Amounts: Amount that each level of improvement will increase on the character (both amount and cost must go up to the maximum level).

Upgrade Costs: Cost of each level to upgrade.

Upgrade Icon: Upgrade icon that will appear in the menus.

Currency Tag: ID of the currency that will be needed for the update (by default it will be the GO currency (gold), but you can edit it).

Previous
Creating a new scene

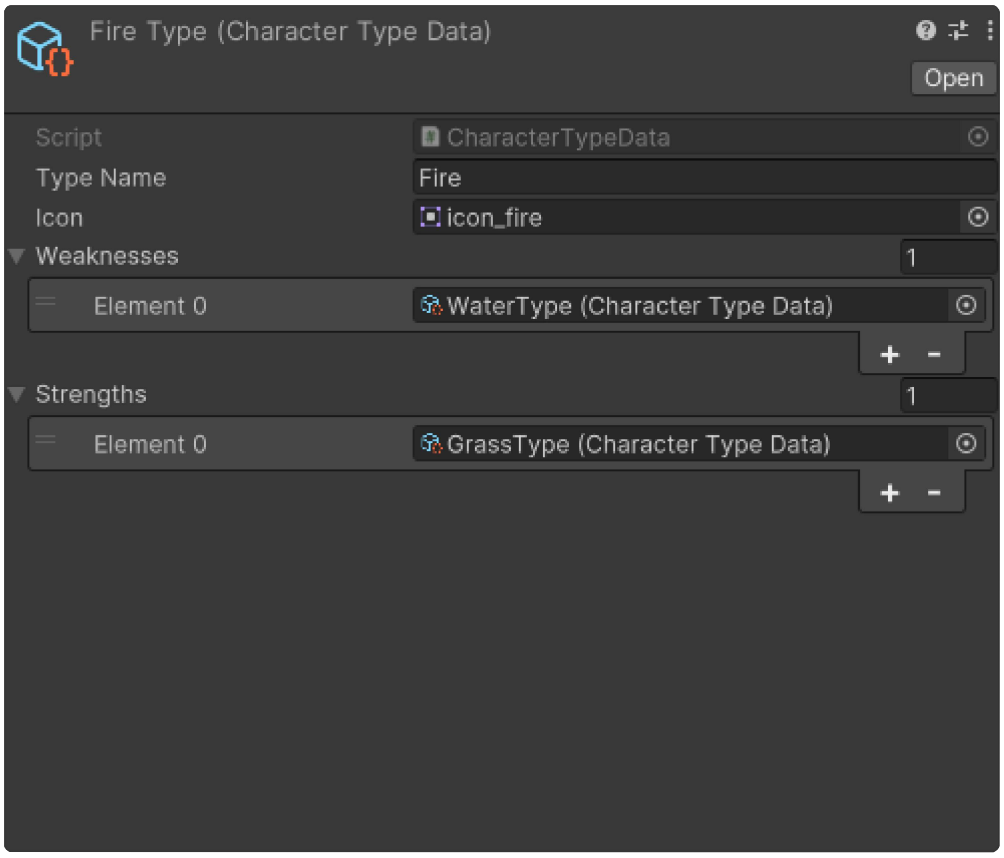
Next
Create new Character Type

Create new Character Type

Component responsible for defining the advantages and disadvantages of this element (fire, water, etc.).

Remember that this item has to be added to the GameInstance in the Login Scene.

To create a CharacterType, right-click inside any project folder, go to create and select CharacterTypeData.



Type Name: Name of the element that appears in the menus.

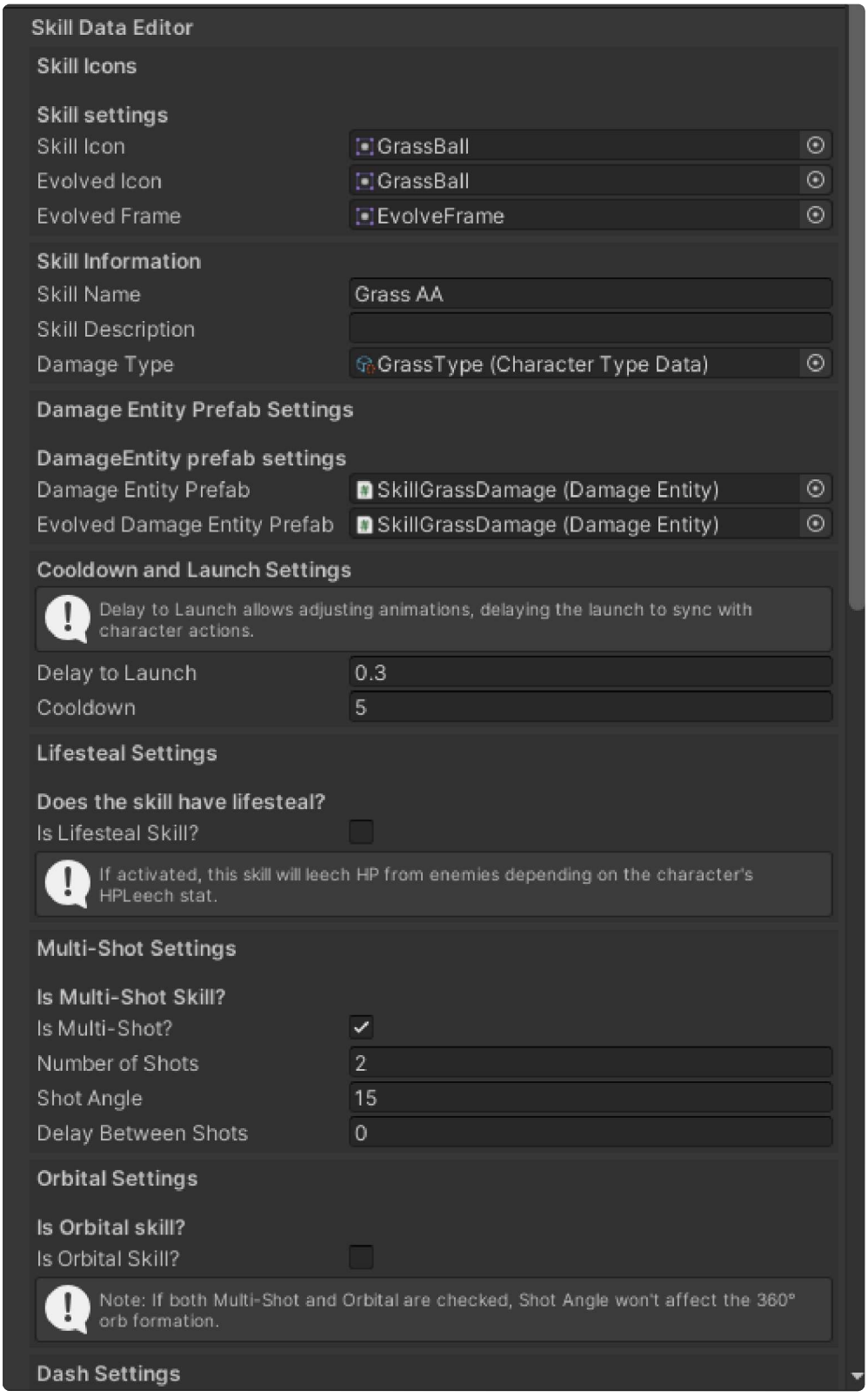
Icon: Icon of the element that appears in the menus.

Weaknesses: Add other character types to set as weaknesses (will take more damage from those elements)

Strengths: Add other character types to set as Advantages (will take less damage from these elements)

Create New Skill

To create a SkillData, right-click inside any project folder, go to create and select Skill Data.



Icon: Skill icon.

Icon Evolved: Skill icon when reaching maximum level.

Frame Evolved: Skill frame in the options menu when it is the card for the maximum level of this skill.

Skill Name: Name of the skill that appears in menus.

Skill Description: Description of the skill that appears in the menus.

DamageType: Damage element of this skill (will define whether the damage will increase, reduce or be neutral on the enemy hit).

Delay to Launch: Time before casting a skill, use this to delay the skill casting to match the animation.

Cooldown: Skill cooldown.

isLifestealSkill: Check this box if you want the skill to apply HP drain to enemies.

DamageEntityPrefab : Add the default DamageEntity prefab for this skill.

EvolvedDamageEntityPrefab: Add the DamageEntity prefab for when this skill reaches max level (both skill prefabs are required, even if they are the same).

IsMultiShot: Check this box if you want the skill to drop multiple DamageEntities.

Shots: Amount of skill shots (only if isMultiShot is checked).

Angle: change the angle between one shot and another, leave it at 0 (zero) if you want them all to go in the same direction.

Delay: Delay between one shot and another, leave it at 0 (zero) and it will fire all the shots at once.

Evolve Changes: Check this box if you want to change the multi-shot settings when evolving.

Shots Evolved: Number of shots the skill will fire when at max level.

Angle Evolved: Angle between shots when this skill is at maximum level.

Delay Evolved: Delay between shots when skill is at max level.

IsOrbital: Check this box if you want the ability to orbit the player until their LifeTime runs out. This works with multiShots, but the angle does not affect the ability as it will attempt to form a 360 degree circle with the orbs.

Orbital Distance: Distance at which DamageEntities orbit the player.

is Dash Skill: Check this box if you want to set the skill to dash, it works well for melee and ranged, basic attacks cannot use dash as they always target the closest target.

is Shield: Check this box if you want the character to gain a shield when using the skill, set the amount of shield the character gains and the duration of this shield.

is Melee: Spawn the DamageEntity inside the character making it become your son and follow you, for better benefits leave the speed at 0 (zero) in skillsLevels making it 100% melee, this skill works well with dash and KnockBacks.

Rotate to Enemy: Check this box if you want the character's 3D model to rotate in the direction of the joystick and return to its original position over time. In Auto Attack skills it will rotate in the direction of the nearest target.

ApplySlow: check the box if you want the skill to slow enemies, set the percentage and duration of the slow.

ApplyKnockback: check the box if you want the skill to push enemies, set the distance and time of the Knockback.

ApplyStun: check the box if you want the skill to stun enemies, set the stun time.

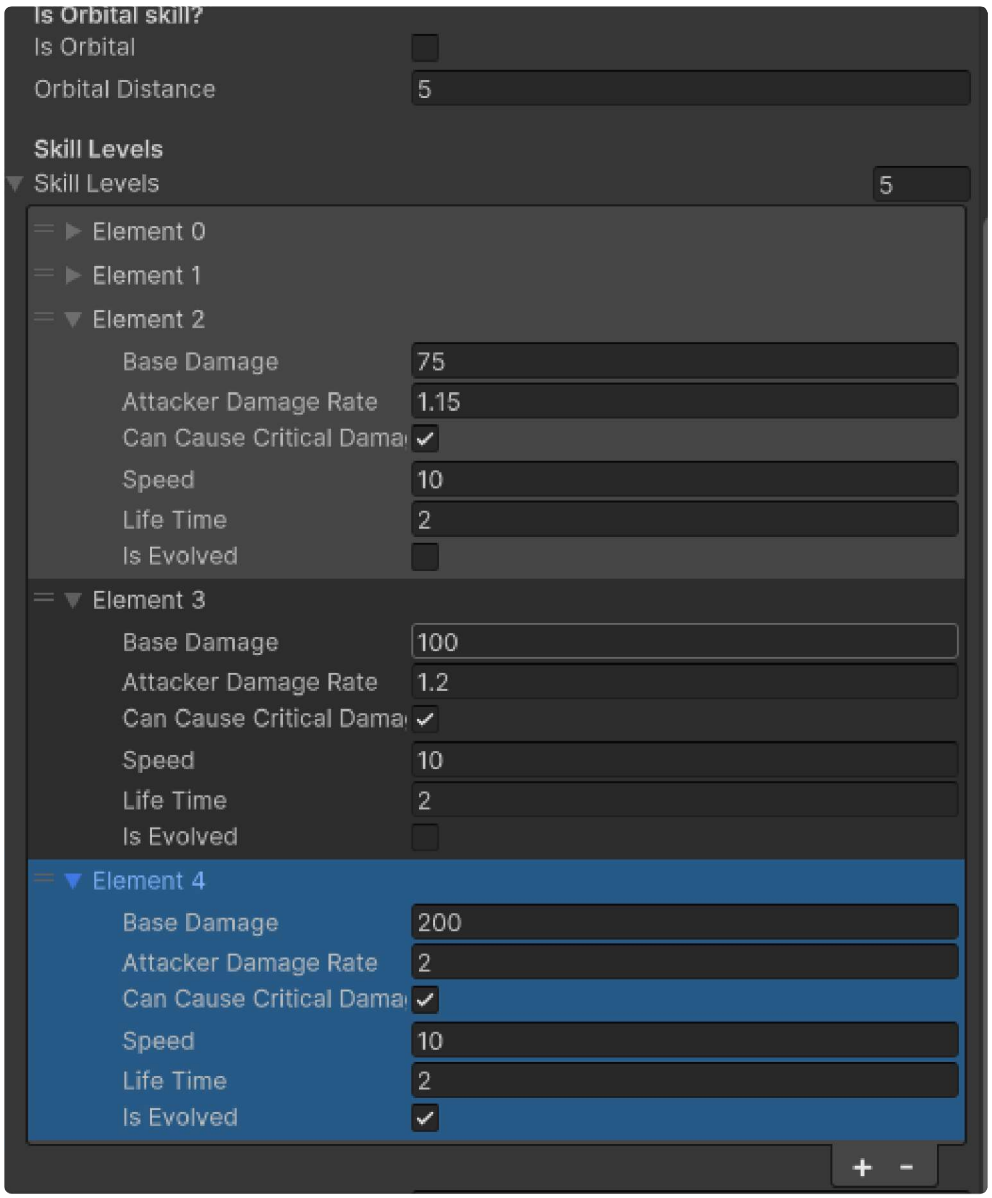
Apply Dot(damage over time): check the box if you want the skill to deal a certain amount of damage over a period of time, set the total damage and the period This is extra damage on top of the skill's Base Damage.

Skill Levels: Skill setup for each level.

MaxLevel: Max level of this skill (recommended level 5, otherwise you will need to edit the UIGameplay).

Requiere Stat for Evolve: Select a StatPerk that the player must have chosen to be able to evolve a skill, if he does not have that StatPerk at least lv1 the card to take that skill to levelMax will not appear.

Skill Levels



Base Damage : Base damage this skill deals before elemental reductions or increases.

Attacker Damage Rate: Percentage of the character's damage that increases the base damage. Example: 0.5 will add half of the base damage of the CharacterData to the skill 1 will add 100% of the base damage of the CharacterData to the skill
Use this so that the same skill can be reused on other characters with different damage.

CanCauseCriticalDamage: If a skill can deal critical damage.

Speed: Speed that the DamageEntity of this skill will have.

LifeTime: Damage Entity duration time.

isEvolved: Check this box only at the level that will be the maximum and will make the skill evolve.

Previous
Create new Character Type

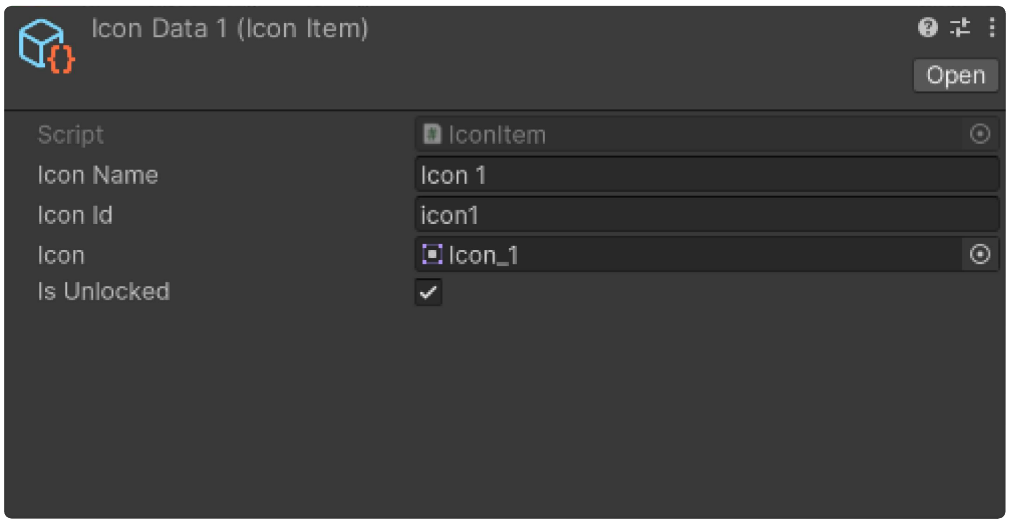
Next
Create New Player Icon

Last updated 15 hours ago

Create New Player Icon

To create a IconData, right-click inside any project folder, go to create and select Create Icon Data.

Remember that this item has to be added to the GameInstance in the Login Scene.



Icon Name: Name of the icon that appears in the store.

Icon ID: Unique player icon ID.

Icon: Image representing the icon.

IsUnlocked: Check this box for icons that the player will already have unlocked.

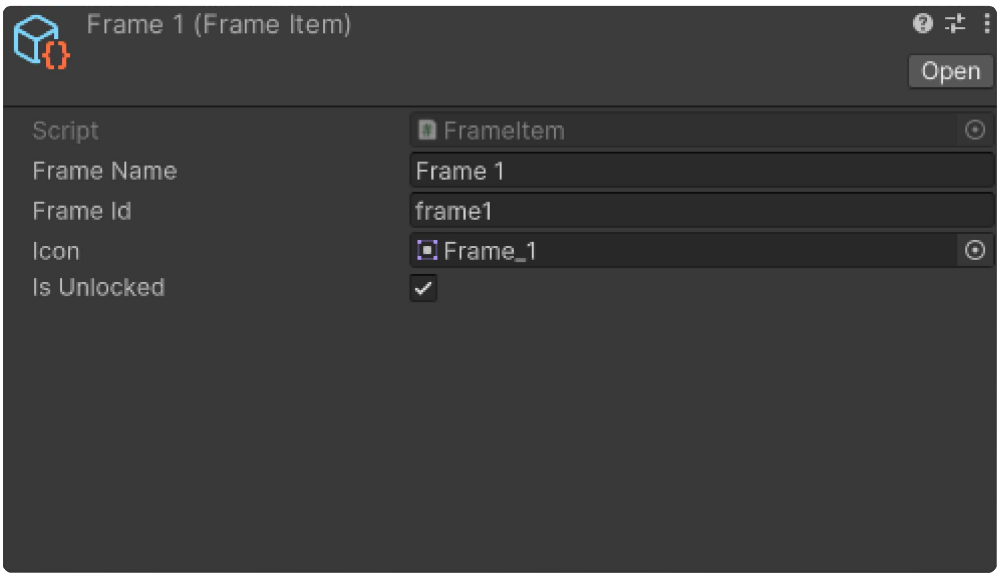
Previous
Create New Skill

Next
Create New Player Frame

Create New Player Frame

To create a FrameData, right-click inside any project folder, go to create and select Create Frame Data.

Remember that this item has to be added to the GameInstance in the Login Scene.



Frame Name: Name of the frame that appears in the store.

Frame ID: Unique player frame ID.

Icon: Image representing the frame.

IsUnlocked: Check this box for frames that the player will already have unlocked.

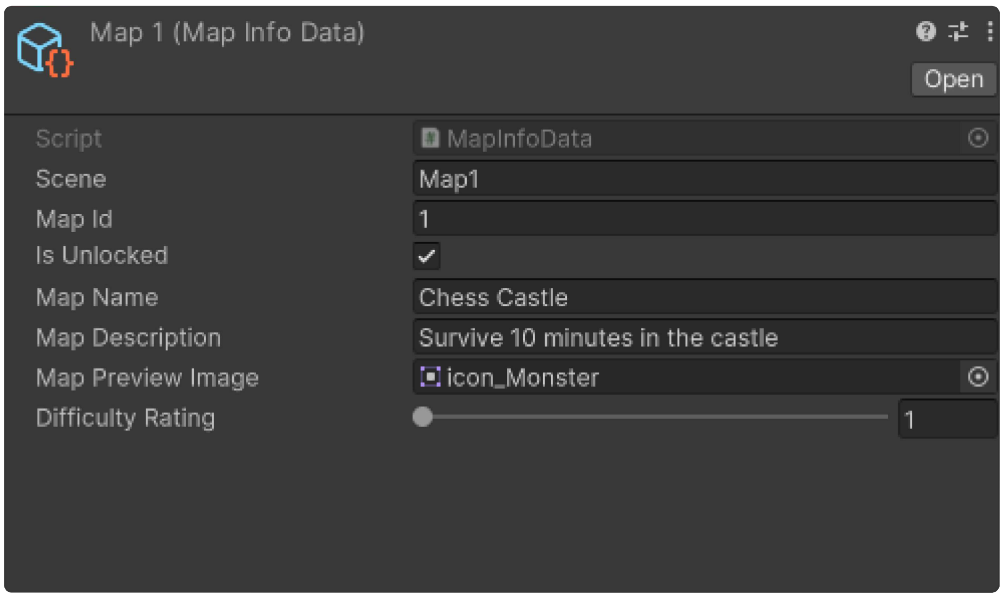
Previous
Create New Player Icon

Next
Create New Map Info Data

Create New Map Info Data

To create a MapInfoData , right-click inside any project folder, go to create and select Map Info Data.

Remember that this item has to be added to the GameInstance in the Login Scene.



Scene: Name of the scene that this MapInfo will load.

MapId: Unique ID for each map.

IsUnlocked: Check this box for maps that the player will have already unlocked (1 map in the game must have this box checked).

Map Name: Name of the map that appears in the menus.

Map Description: Brief description of the map that appears in the map selection menu.

MapPreviewImage: Map icon that appears in the selection menu.

Difficulty Rating: Map difficulty (visual only).

Previous
Create New Player Frame

Next

Create New Quest Item

To create a QuestItem , right-click inside any project folder, go to create and select Quest/QuestItem.

Quest ID: Unique QuestItem ID.

Icon: Quest icon that will appear in the menus.

Title: Mission title that appears in menus.

Description: Description of the map that appears in the menus.

Currency Reward: ID of the reward currency when completing this Quest, by default it is GO(Gold) but you can change it to any currency.

Reward Amount: Amount of coins the quest rewards upon completion.

IsDayleQuest: Leave checked for Quests that you want to be daily (every day at the Firebase server time set in the FirebaseManager the daily quests will be reset)

Requirement Type: You can choose the quest requirement from:
Kill Monster: Kill monsters to complete.
Complete Map: Finish a specific map to complete.
Kill Monster with Specific Character: Kill monsters with a specific character set.

Target Amount : Number of times you must complete the stage, or number of monsters to kill, depending on what is chosen in the requirement.

Target Map: MapInfoData of the map you want to be completed.**(only for complete map requirement)**

Target Character: Character that should be used to kill the monsters (only for the Kill Monster with Specific Character requirement).

Previous
Create New Map Info Data

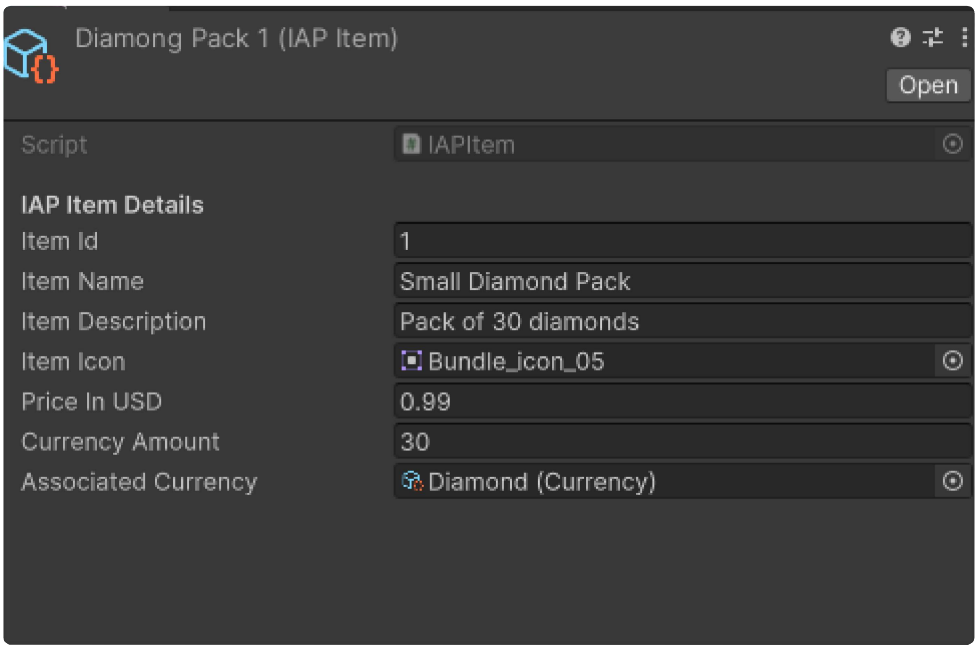
Next
Create New IAP shop Item

Last updated 15 hours ago

Create New IAP shop Item

To create a IAPItem , right-click inside any project folder, go to create and select IAP/IAP item.

Remember that this item has to be added to the IAPManager on the Login Scene.



item ID: Unique IAPItem ID.

Item Name: title that appears in menus.

Description: Description of the item that appears in the menus.

Icon: IAP item icon that will appear in the menus.

Price in USD: Item value in USD, Google Apple stores etc. will be converted automatically.

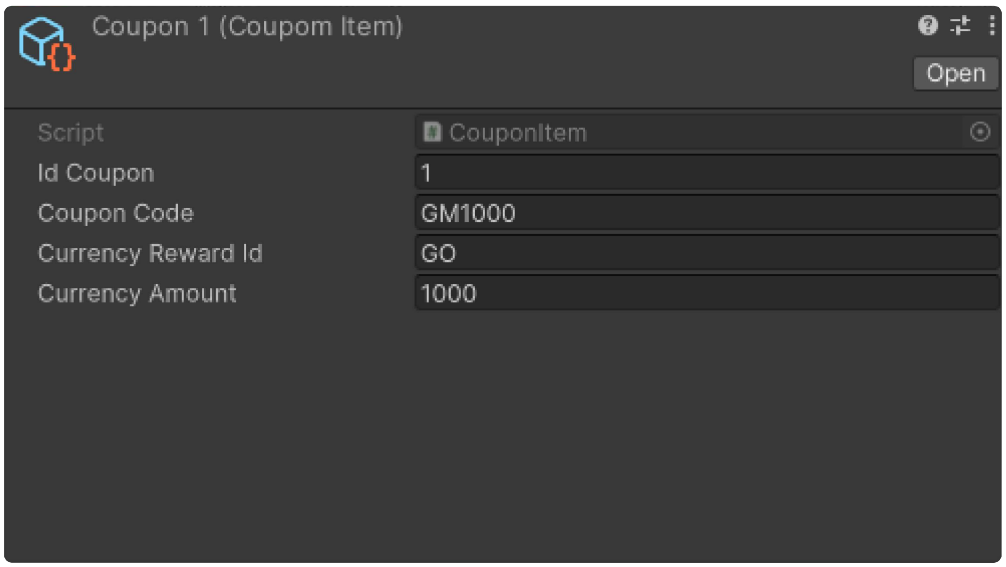
Currency Amount: Amount of coins that will be received if the purchase is successful;

Associated Currency: Currency that will be received upon purchase.

Create new Coupon Item

To create a CouponItem, right-click inside any project folder, go to create and select Coupon/CouponItem.

Remember that this item has to be added to the GameInstance in the Login Scene.



Id Coupon: Unique ID for each CouponItem.

Coupon Code: Code that the player must use to redeem Coupons.

Currency Reward Id: ID of the currency the player will receive.

Currency Amount : Amount of currencies the player will receive.

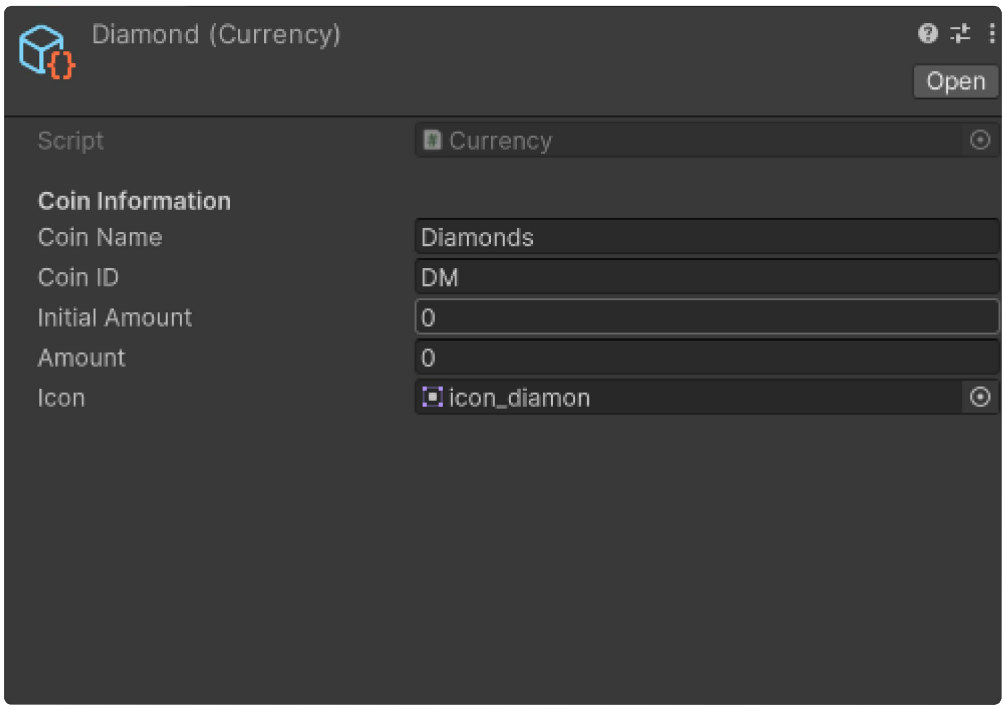
Previous
Create New IAP shop Item

Next
Create New Currency

Create New Currency

To create a Currency, right-click inside any project folder, go to create and select Monetization/Currency.

Remember that this item has to be added to the GameInstance in the Login Scene.



Coin Name: Name of the currency that appears in the menus, if necessary.

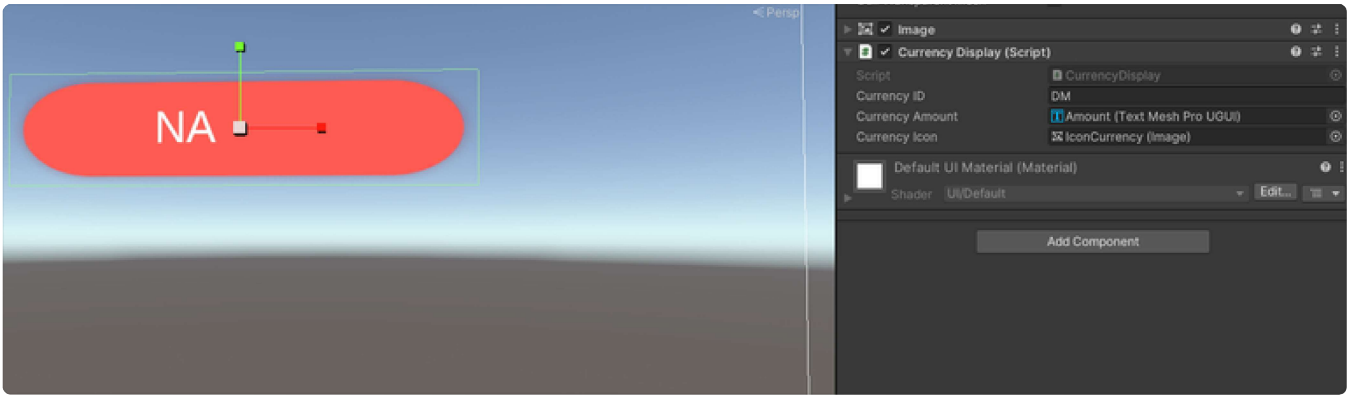
Coin Id: Unique ID that identifies the currency and will be used in reward items like quests and coupons.

Initial Amount: Initial value of this currency that the player will have.

Amount: Current amount of currency (I recommend leaving 0 (zero) and letting the Backend (firebase) take care of updating this value).

Icon: Currency Icon.

Display currency in game interface:



To display the amount of currency on the screen, create a CurrencyDisplay component. You can use the Display_CurrencyGold object from the home scene as an example to understand how to do this.

Currency ID : Currency ID you want to display on screen.

Currency Amount: Text that will display the amount of currencies.

Currency Icon: Image containing the Scriptable Currency sprite/icon.

Previous
Create new Coupon Item

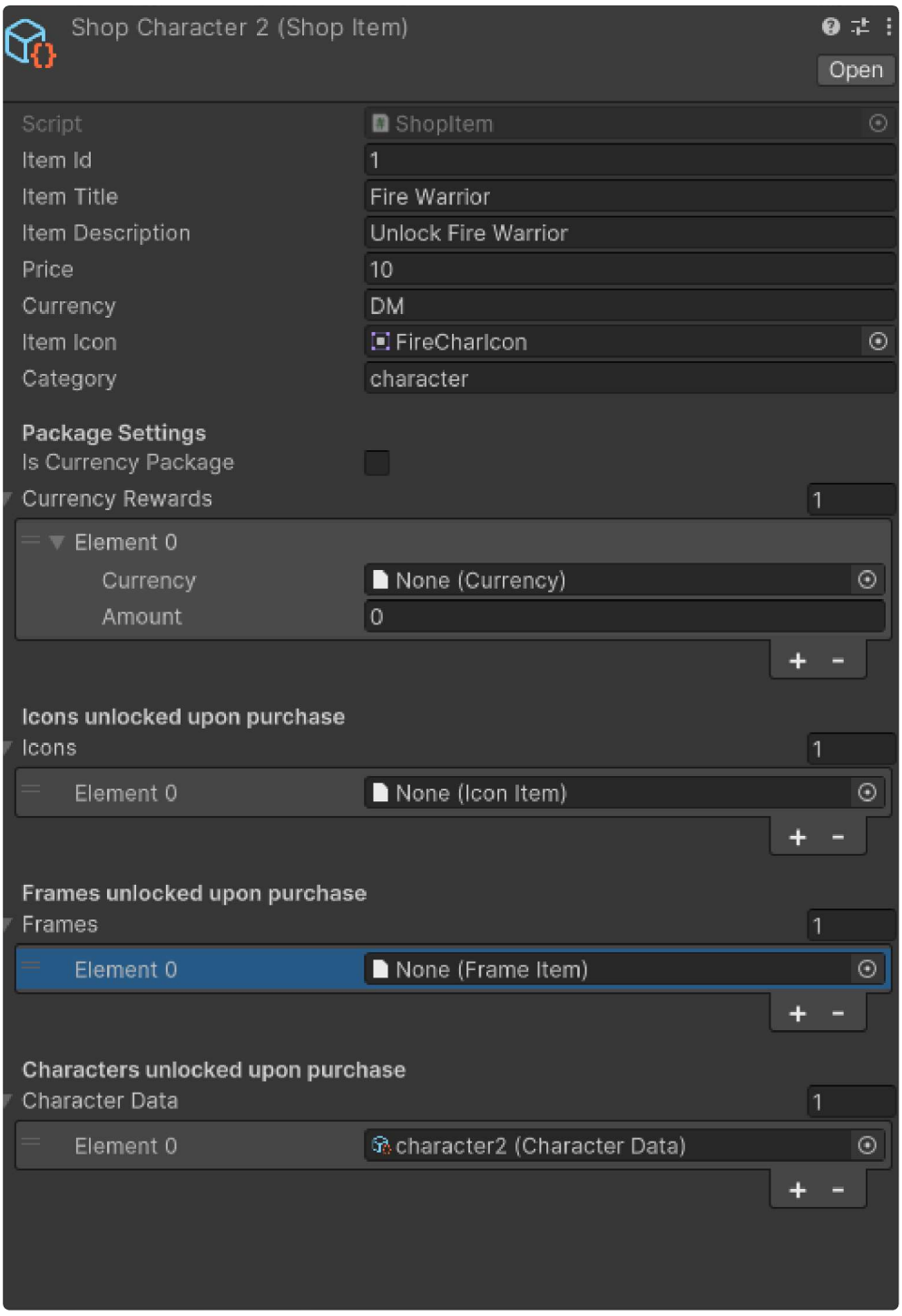
Next
Create New ShopItem

Last updated 15 hours ago

Create New ShopItem

To create a ShopItem, right-click inside any project folder, go to create and select Shop/Item.

Remember that this item has to be added to the UIShopMenu in the Home Scene.



Item Id: ShopItem Unique ID.

Item Title: Title of the Shopitem that appears in menus.

Item Description: Description of the item that appears in the menu.

Price: Price of the item or package.

Currency: Currency required to purchase the item.

Item Icon: ShopItem or package icon that will appear in the menu.

Category: Category to filter similar items.

IsCurrencyPackage: Check this box for items that only give gold and are not removed from the store after purchase.

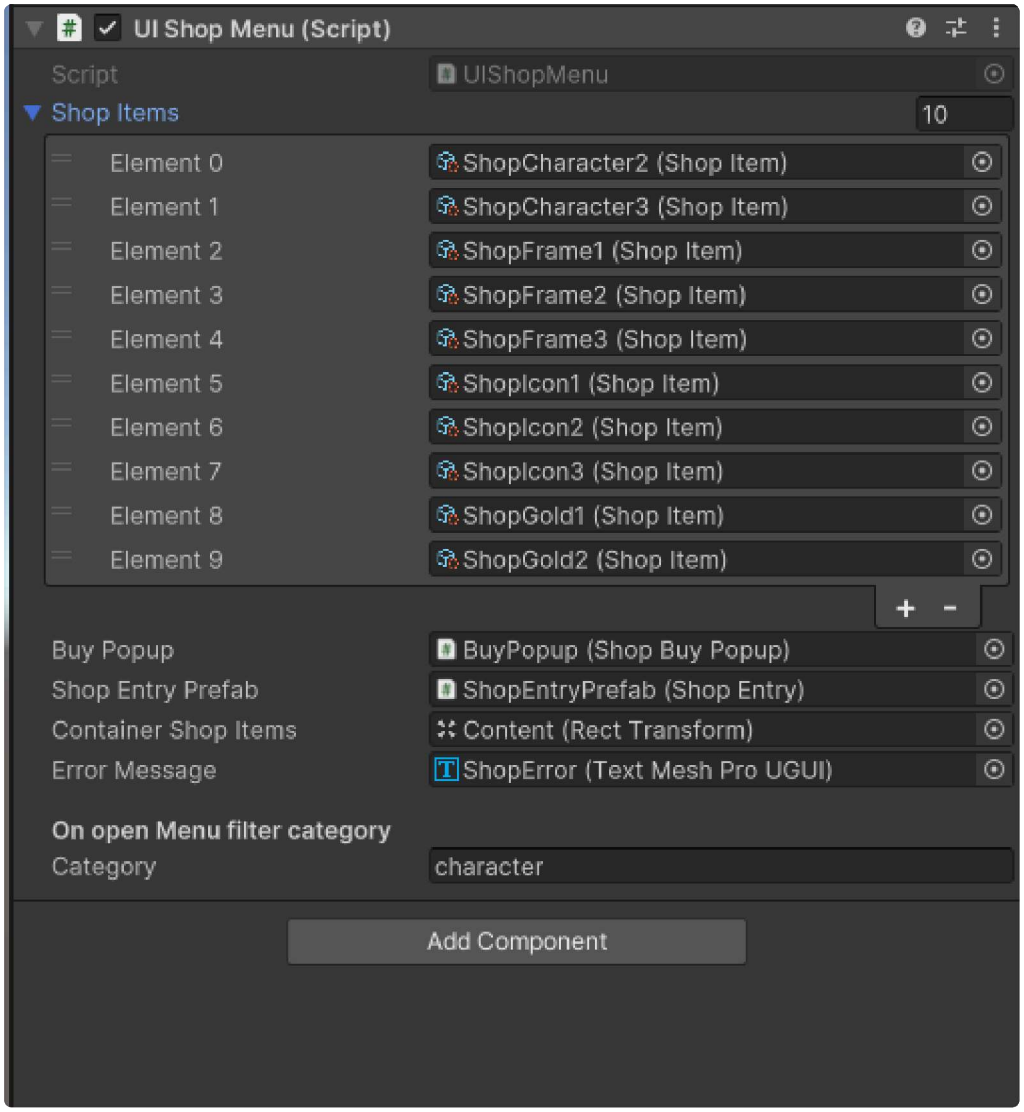
CurrencyReward: Configure the Currency that the player will receive when purchasing this item, remember to check the IsCurrencyPackage box and do not add other items to this ShopItem, that is, leave the item exclusively to deliver Currencies.

Icons: List of icons that the player will receive when purchasing this Shopitem.

Frames: List of Frame that the player will receive when purchasing this Shopitem.

CharacterData: List of character that the player will unlock when purchasing this Shopitem.

Shop Menu:



ShopItems: add all Shoptems that appear in the Shop menu.

Buy Popup: Popup that will appear when clicking on an item to confirm the purchase.

Shop Entry Prefab: Prefab that the menu will create for each ShopItem added.

Container Shop Items: Location that will instantiate ShopEntry prefabs.

Error Message: Text that displays store error messages, such as insufficient currency and others.

Category: Category of items that will be filtered when opening the store.

Previous
Create New Currency

Next
Create New BattlePassItem

Create New BattlePassItem

To create a BattlePassItem, right-click inside any project folder, go to create and select BattlePass/Item.

Remember that this item has to be added to the UIBattlePass in the Home Scene.

Pass Id: ShopItem Unique ID.

Item Title: Title of the Shopitem that appears in menus.

Item Description: Description of the item that appears in the menu.

Item Icon: ShopItem or package icon that will appear in the menu.

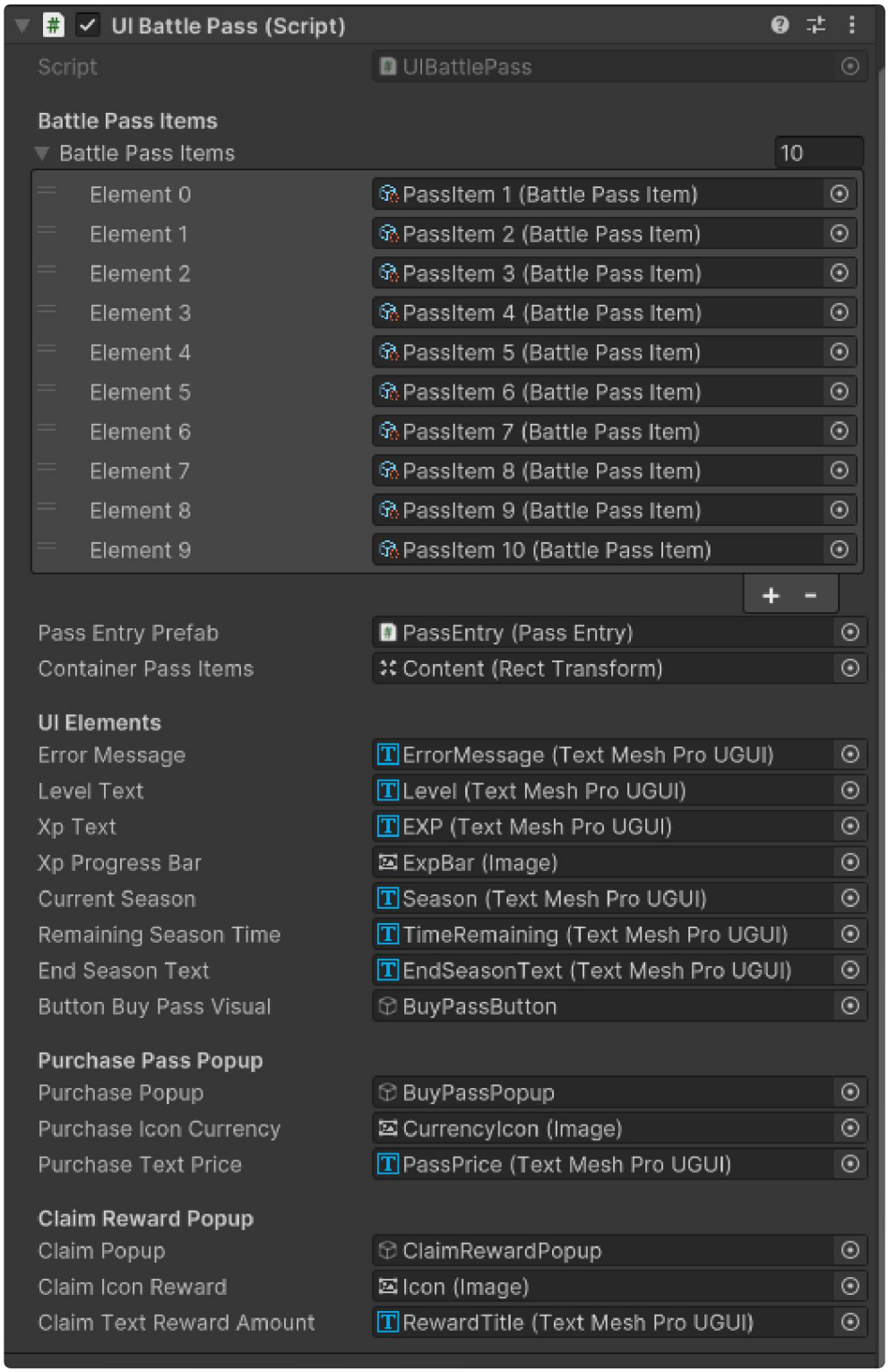
RewardType: Reward type when redeeming, you can only choose one.

RewardTier: Choose whether it will be free or paid, choosing paid will make it necessary to purchase the premium pass to redeem.

Currency Reward: Currency that the player earns when redeeming.

Icon Reward: Icon that the player earns when redeeming

Frame Reward: Frame that the player earns when redeeming



BattlePass Items: Battle Pass Scriptables List items that will appear in the UIBattlePassMenu, put them in the order you want them to appear.

PassEntry Prefab: Configured prefab that appears for each list item.

ContainerPassItems: Container where the PassEntry will be instantiated.

Error Message: Text where errors and important messages will appear.

Level text: Current pass level text.

Xp text: Shows current XP in numbers.

Xp progress bar: Shows current XP in a progress bar.

Current Season: Current season pass text.

Remaining Season time: season time remaining in days, hours, minutes and seconds format.

Button Buy Pass Visual: Button that appears only for players who have not yet purchased the pass and brings up the purchase menu.

Previous
Create New ShopItem

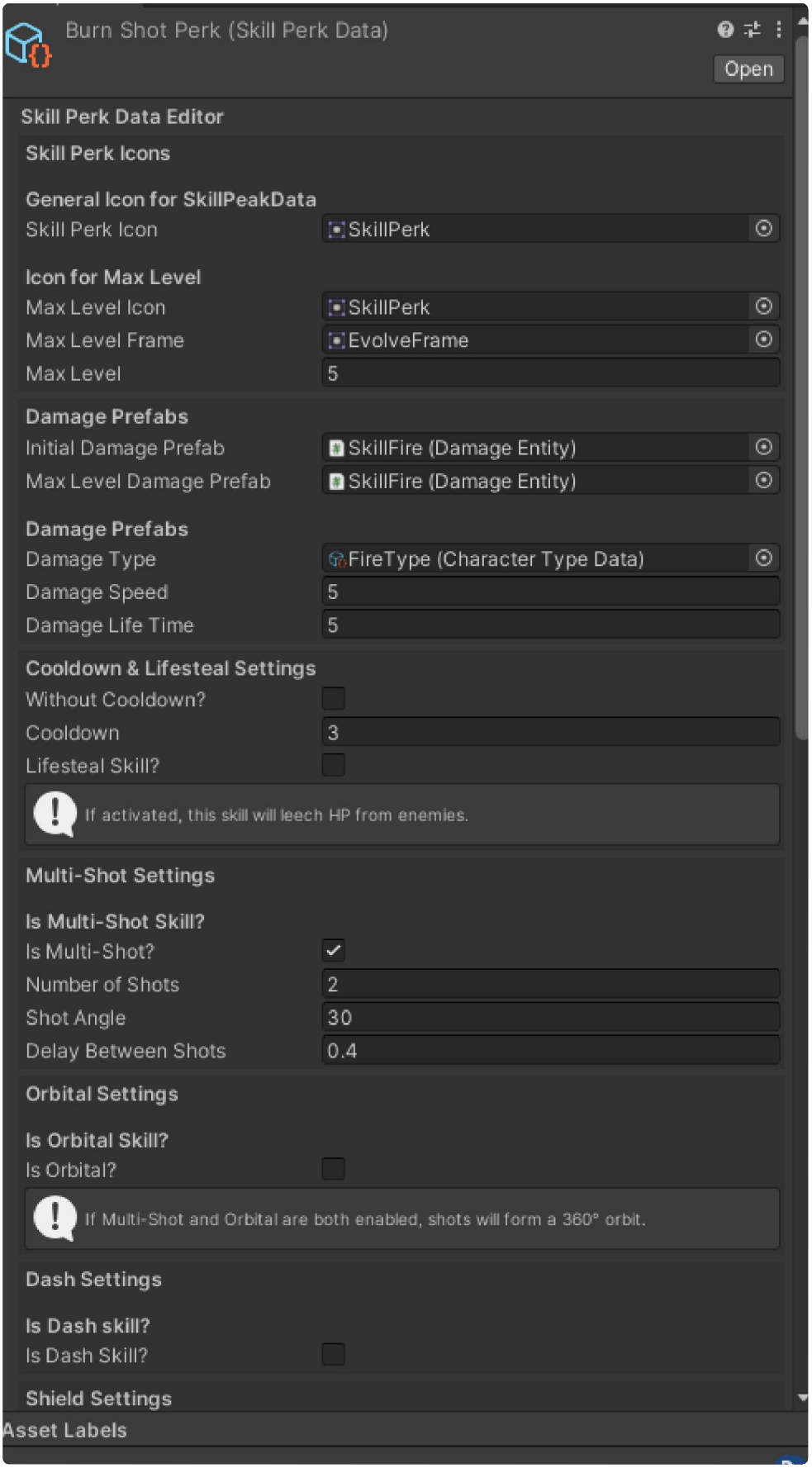
Next
Create New SkillPerkData

Last updated 15 hours ago

Create New SkillPerkData

To create a SkillPerkData, right-click inside any project folder, go to create and select skills/SkillPerkData.

Remember that this item must be added to the GameplayManager within each scene.



See the Create New Skill tab, as the settings are practically the same The difference is that this skill needs to be added to the GameplayManager of the scenes you want to be able to choose, and when the player levels up he has the chance to choose this Perk

Unlike player skills, SkillsPerkData are used automatically like the Auto Attack during the cooldown time of this skill, Cooldown reduction affects this type of skill

Previous
Create New BattlePassItem

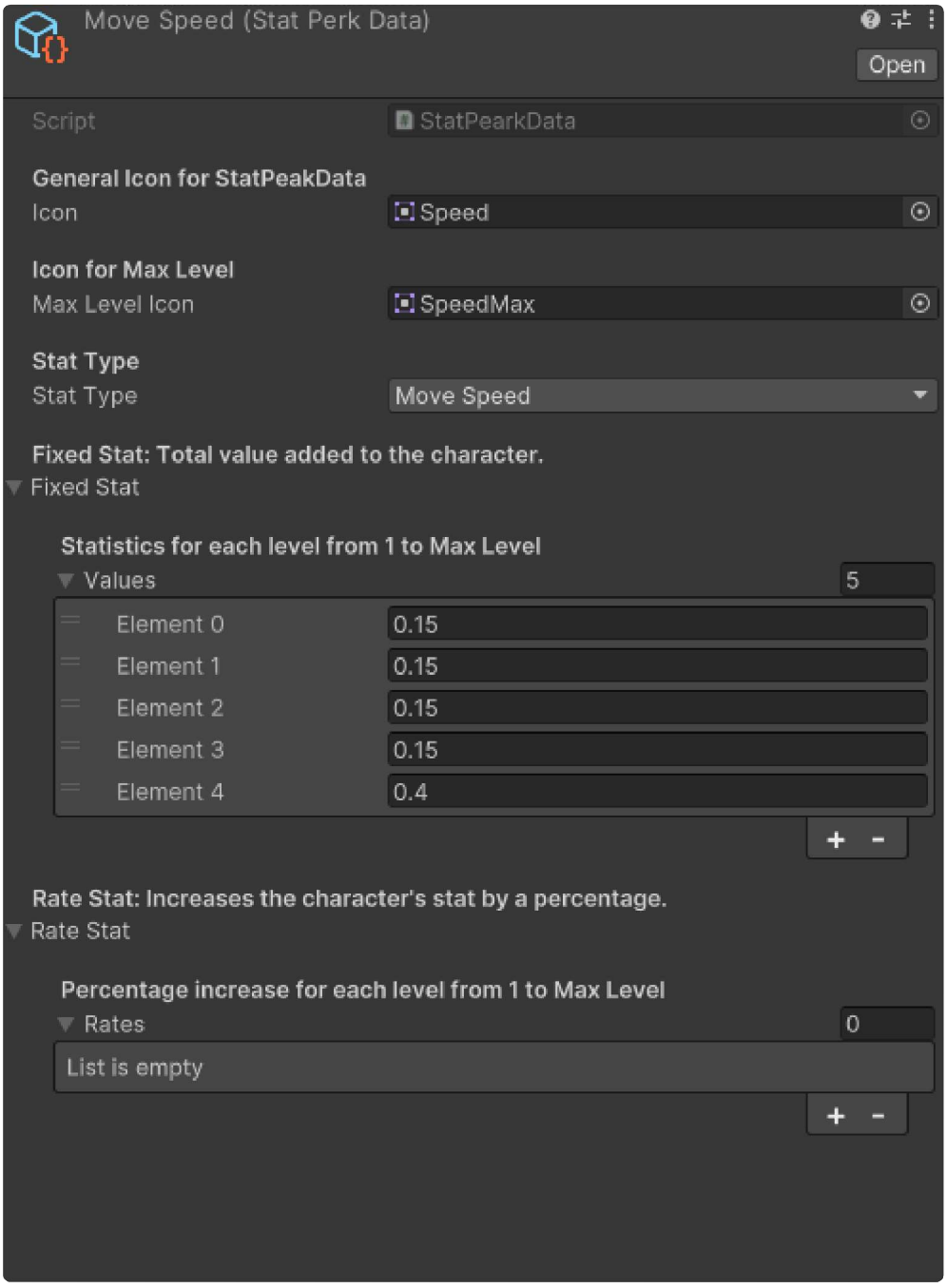
Next
Create New StatPerkData

Last updated 15 hours ago

Create New StatPerkData

To create a StatPerkData, right-click inside any project folder, go to create and select stats/StatPerkData.

Remember that this item must be added to the GameplayManager within each scene.



Stat Type: Type of status that will be increased when the player chooses this perk.

Fixed Stat: Fixed values that will increase the chosen stat based on the level (number of times the player chose this perk).

Rate Stat: Percentage of the base stat that will be increased when choosing this park based on the level. In cases where the player has a total of 0 of this stat, nothing will be increased. But you can add a fixed value and a percentage together.

Previous
Create New SkillPerkData

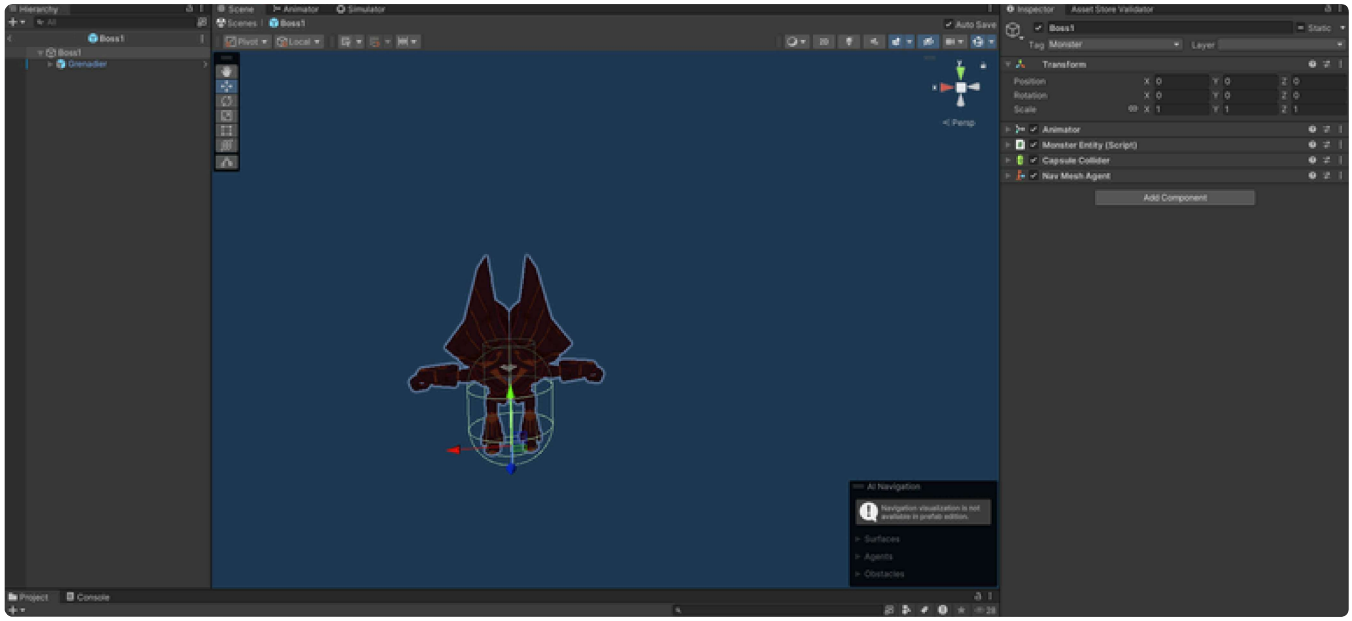
Next
Create New Monster/Mob

Last updated 15 hours ago

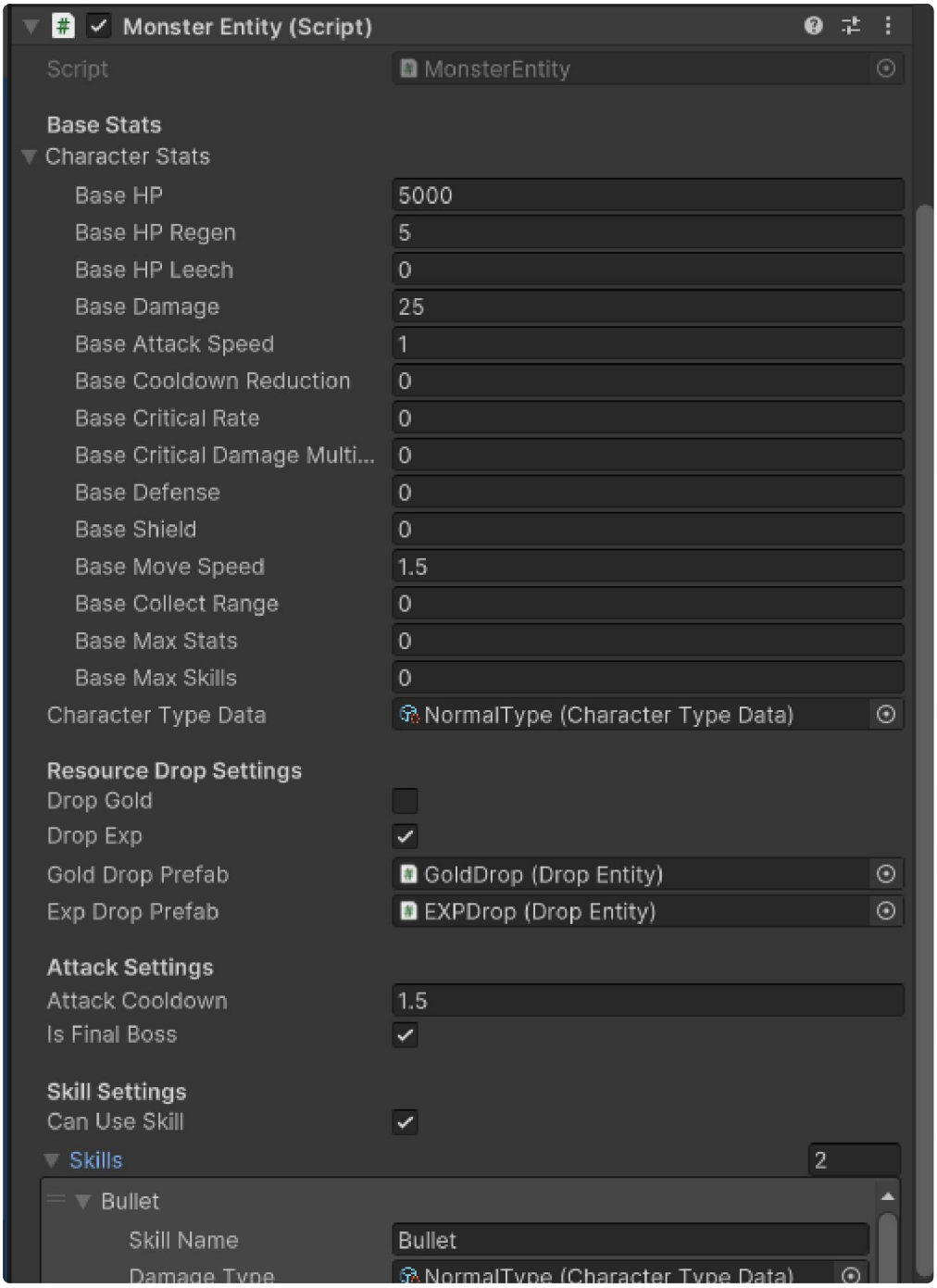
Create New Monster/Mob

To create a MonsterEntity no scriptable is needed, just open the monster prefab and add the components below.

Remember that this item must be added to the GameplayManager/Waves within each scene.



1. Add a capsule collider and configure.
2. Add a NavmeshAgent and configure.
3. Add the MonsterEntity component and configure



Base Stats: Set the Monster's basic stats to the same as the characters'
The following stats are not used by the monster and can be left as 0 (zero):
-Attack Speed (use the attack cooldown to determine the cooldown time to deal damage again while the player is colliding with the monster)
-Cooldown reduction
-Collect Range
-Max Stats
-Mas Skills

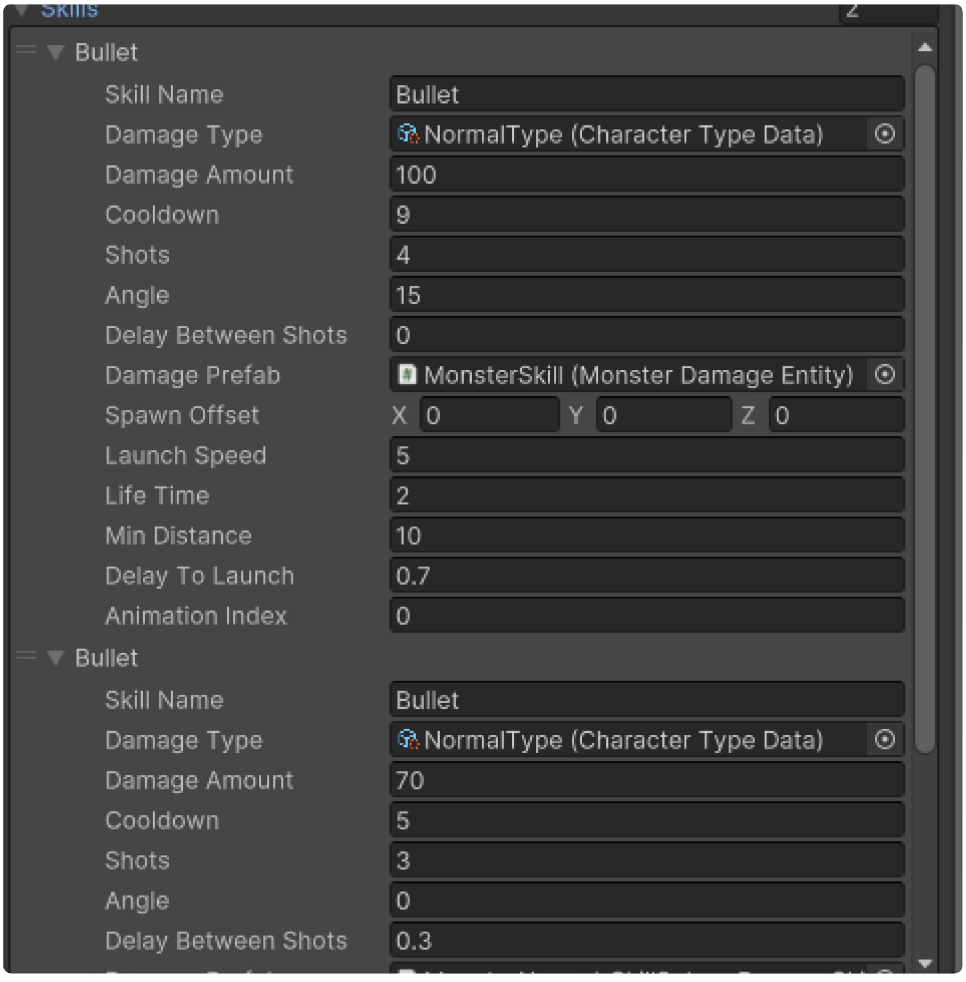
Character Type Data: Damage element that the monster causes when colliding with the player.

Drop Gold/Drop EXP: Check the boxes if you prefer the monster to drop exp or gold, you need to add the drop prefab. Leaving it unchecked makes the exp and gold be collected automatically without dropping on the ground.

Attack Cooldown: cooldown for the monster to deal damage again to the player who collides with it.

is Final Boss: Check this box to set the monster as the final boss, making its hp bar appear when it is spawned, there can only be 1 boss per scene.

Can Use Skill: Check an option for the monster to use skills.



SkillName: monster skill name, but currently it is not shown in any UI.

Damage Type: Damage element a skill deals.

Damage Amount: Damage amount before element reductions and increases.

Cooldown: Cooldown of this skill, monsters automatically use the skill when it comes off cooldown.

Shots: Number of skill shots.

Angle: Angle between one shot and another.

Delay Between Shots: delay between one shot and another.

DemagePrefab: Monster Damage Entity prefab that will be spawned.

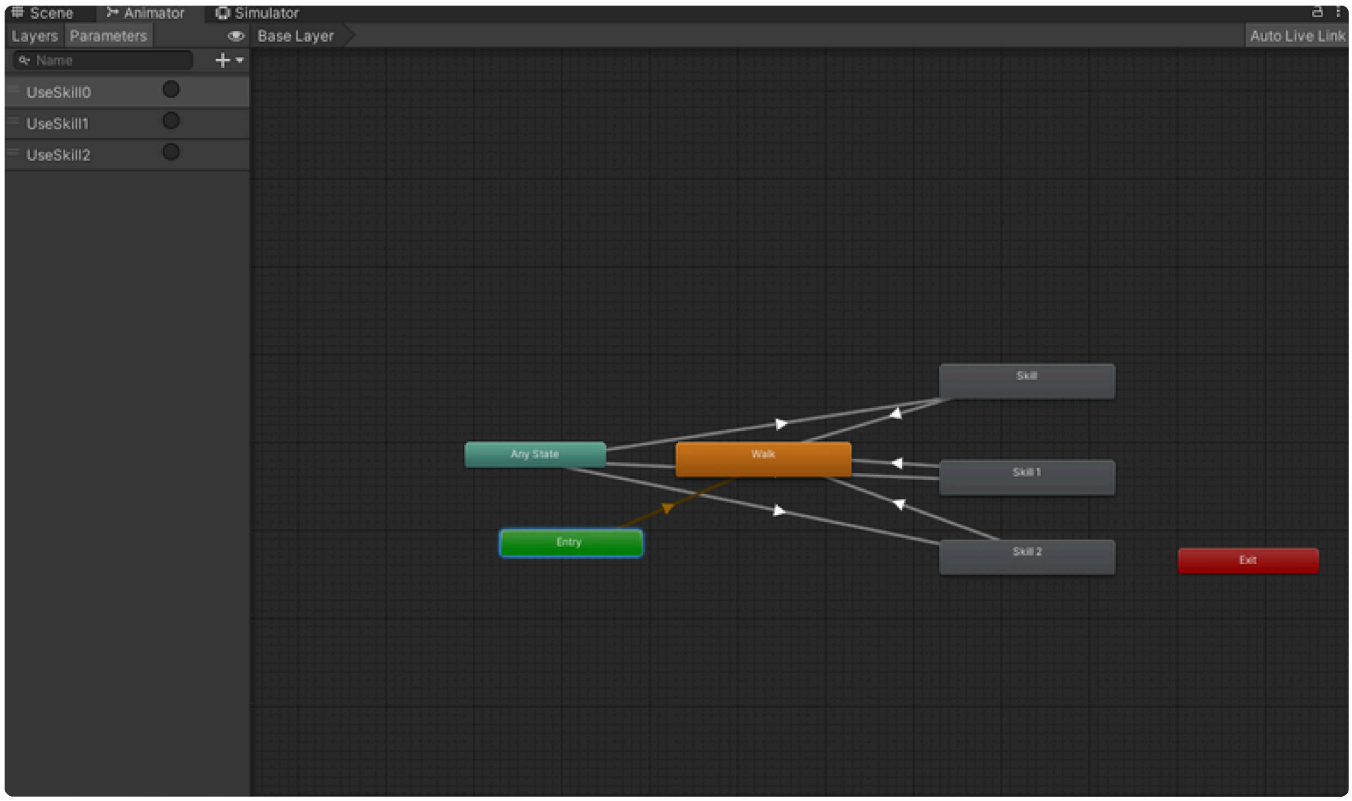
Spawn offset: Options to change the positions where the skill prefab will spawn from.

Launch Speed: Prefab speed.

Min Distance: Minimum distance the player needs to be from the monster for him to use the skill.

Delay to Launch: Delay before generating prefabs, use this to regulate the timing of the skill's animation and spawn.

Animation Index: Index of the animation that will play when using a skill.



The monster only has walking and skill animations, add UseSkill + index as a trigger in the animator If the Animation Index of the monster's skill is 0 it will activate the UseSkill0 trigger and so on.

You can duplicate the existing AnimatorController and configure the skills of a new monster and in the monster's Animator place this edited controller.

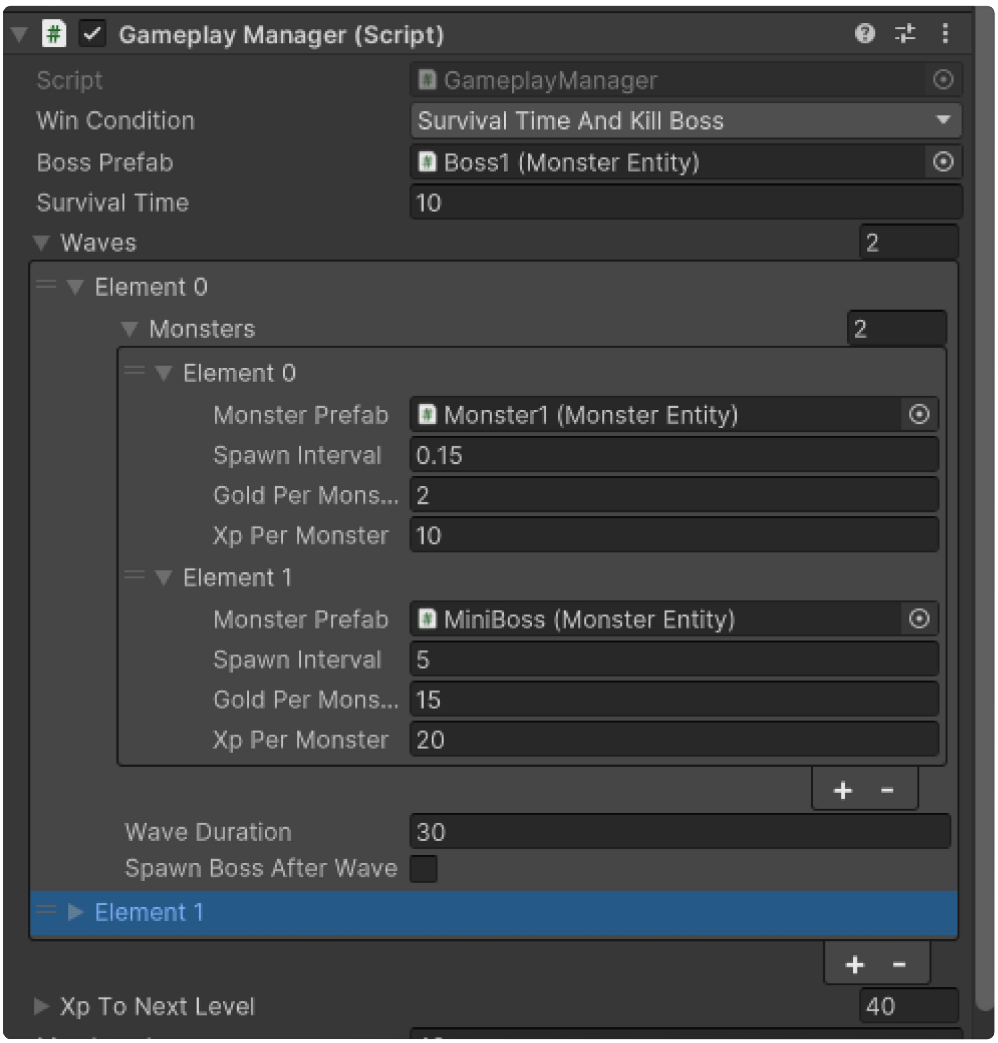
Previous
Create New StatPerkData

Next
Create New Monster wave

Create New Monster wave

To create a Wave no scriptable is needed, just select the GameplayManager

Remember that this item must be added to the GameplayManager within each scene.



1. Win Condition: Select the game type
   -**Survival time**: Win the match if you survive until time runs out.
   -**Kill Boss**: Win the match when the boss spawns after a selected wave and you eliminate him (time does not count in this game mode).
   -**Survival Time + Kill Boss**: Win the match if you survive until time runs out and eliminate the boss that will appear after time runs out.
2. BossPrefab: Final boss prefab with isFinalBoss box marked.
3. Survival Time: Survival time of this map.

Waves:
Each wave can be configured to spawn multiple types of mobs and for a specific duration.

Wave Duration: duration that this wave will spawn the configured mobs.

Spawn Boss After Wave: For "Kill Boss" game type check this box when you want the Boss to spawn after this wave.

Monsters:
Add and configure each monster that this wave will spawn

Monster Prefab: Add Monster Prefab.

Spawn Interval: time between one spawn and another of the same monster.

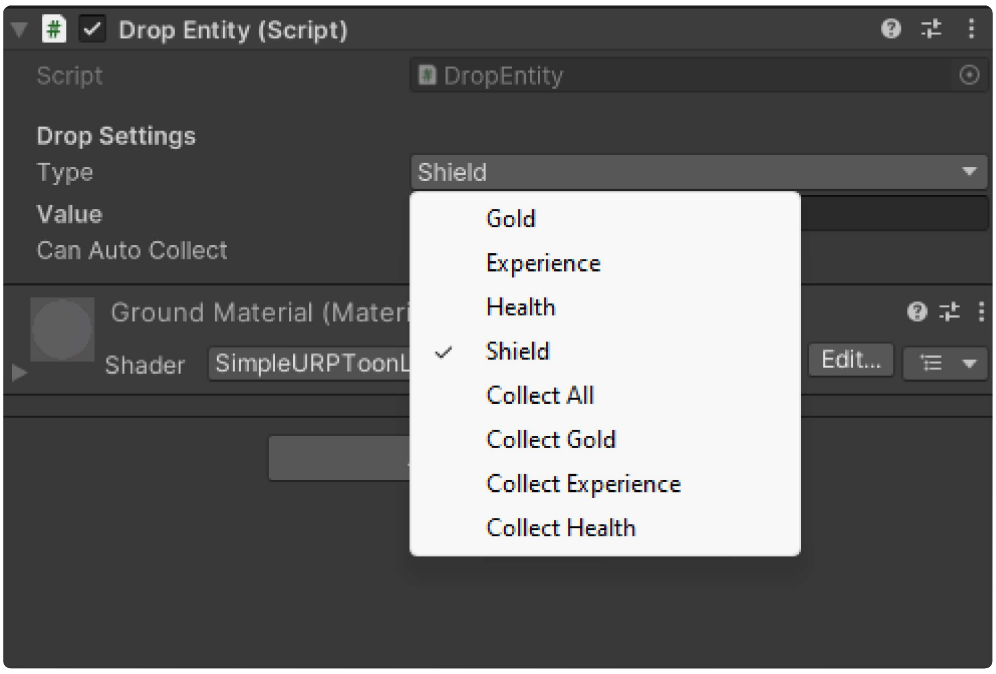
Gold per Monster: Gold that the monster will drop or automatically apply to the player

Xp per Monster: Experience that the monster will drop or automatically apply to the player.

Previous
Create New Monster/Mob

Next
Create New DropEntity

Create New DropEntity



Type:
Gold: Adds gold when collided or picked up.

Experience: Adds EXP when collided or picked up

Health: Heals an amount when collided with or picked up.

Shield: Applies a shield amount to the player that does not run out over time.

Collect All: Collect all other drops with the Can Auto-Collect option checked.

Collect Gold: Collect all other gold drops with the Can Auto-Collect option checked.

Collect Experience: Collect all other Exp drops with the Can Auto-Collect option checked.

Collect Health: Collect all other Health drops with the Can Auto-Collect option checked.

Value: This value is automatically set by entities that drop, but if you prefer to add a drop manually to the scenario you can set a value for it.

Can Auto Collect: Check this box to make the drop automatically collectible by CollectRange and other drops, such as Collect All. Unchecking this box means that the player has to collide with the drop to pick it up.

Previous
Create New Monster wave

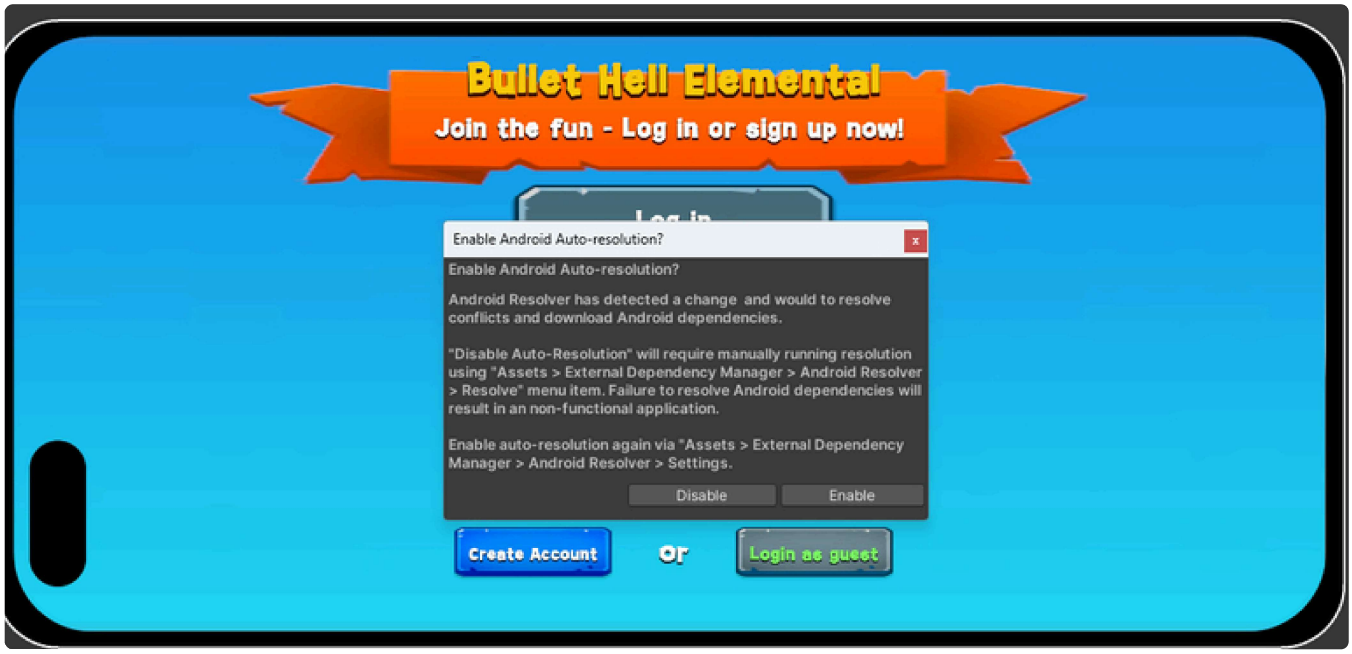
Last updated 25 minutes ago

😓 Common problems

Below is a list of common errors that the Firebase SDK can present and solutions that often resolve them.
Note: All errors below refer to the Firebase SDK and not to the BulletHell Elemental Template, that is, they are errors that you would have using Firebase in any application and would have to look for a solution with the Firebase documentation itself.

Most errors will appear when you open the project for the second time and play the scene as it will automatically install External Dependencies from Google or when you try to compile an APK or executable.

1 -



This screen usually appears the second time I open the project or when trying to compile the game apk

What to do: Click Enable unless you know what you are doing.

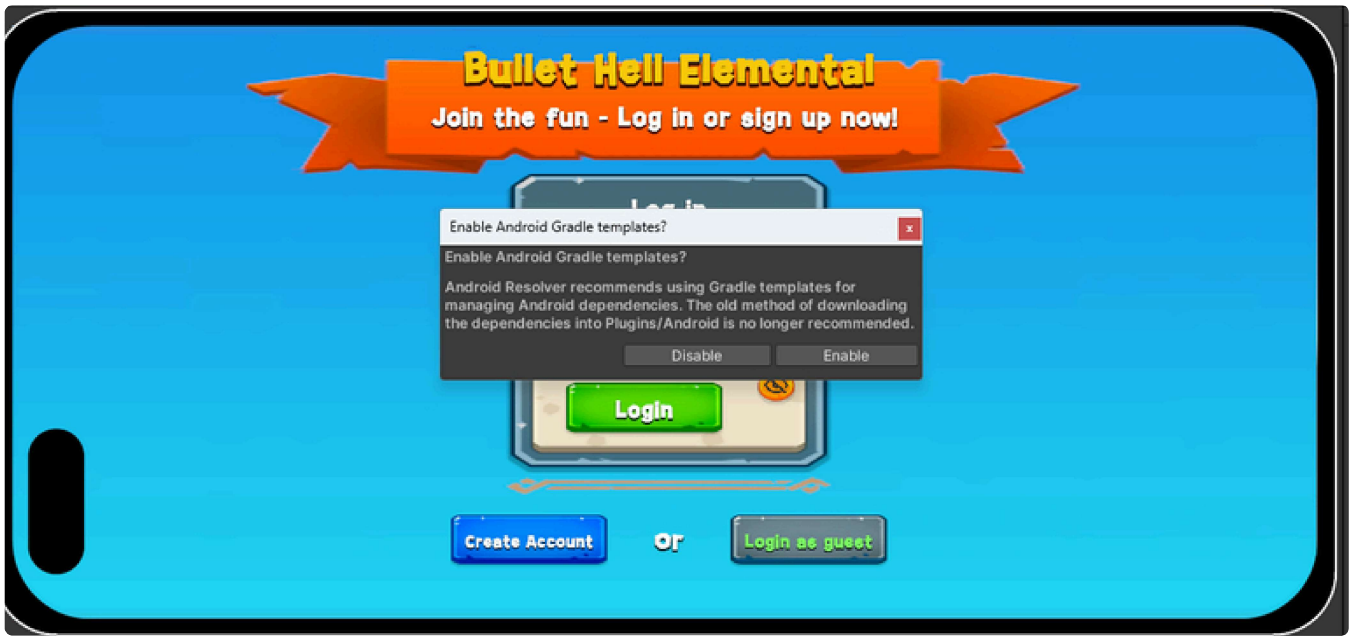
Common issues: Clicking Disable without having other plugins Android and even IOS can cause
-Infinite loop when trying to compile an APK

-Unity crash when trying to compile an APK

-Compiled APK does not connect to Firebase services

Solution: Click on the Unity Assets tab, look for External Dependency Manager, click on Android Resolve and then on Force Resolve and soon this window will appear again to click on Enable.

2-



This window usually appears right after accepting the previous Android Auto-Resolution window.

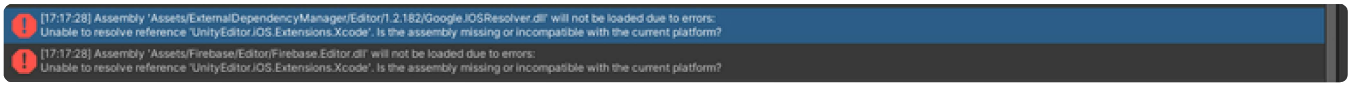
Common issues: Clicking Disable without having other Gradle templates can cause
-Infinite loop when trying to compile an APK

-Unity crash when trying to compile an APK

-Compiled APK does not connect to Firebase services

Solution: Click on the Unity Assets tab, look for External Dependency Manager, click on Android Resolve and then on Force Resolve and soon this window will appear again to click on Enable.

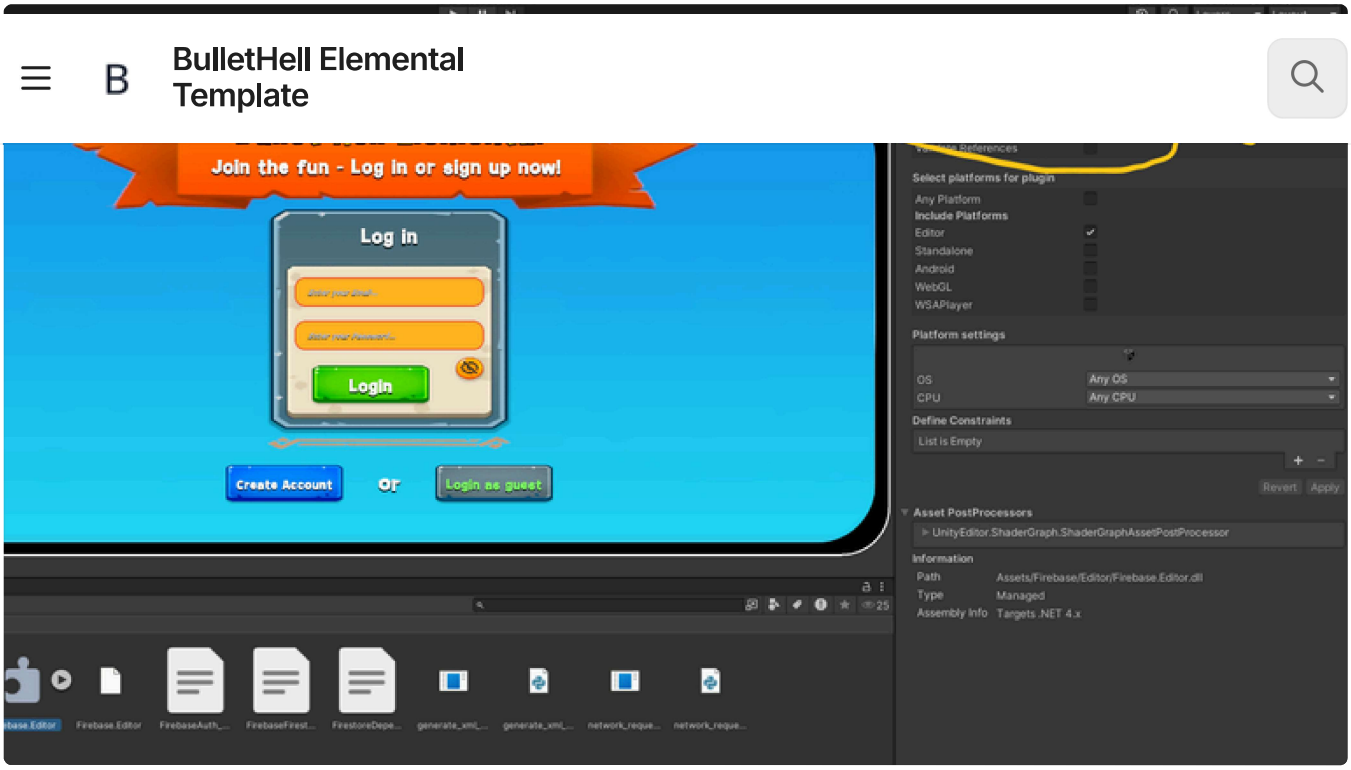
3-



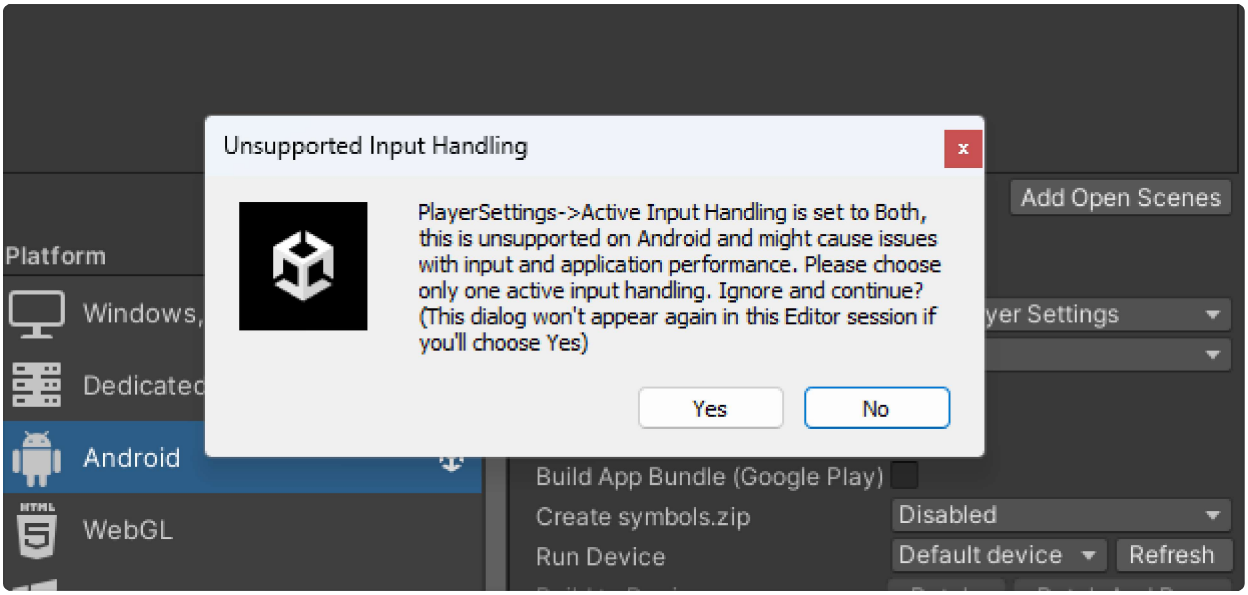
This error appears when one of the required modules is missing from the Unity Engine editor, be it the Android or IOS module.

If you are not going to use one of them, as in the case above the IOS module is missing, then you can either disable DLL checking or install the missing module.

Look for the DLL that is showing the error, in the case above where the IOS module is missing, the Google.IOSResolver.Dll and Firebase.Editor.Dll files are showing the error, select one of them and uncheck Auto Reference and Validate Reference, then wait for it to compile completely before going to the next one, the progress can be seen in a window in the lower right corner of the unit with the message Compiling After compiling, select the other files with error and do the same

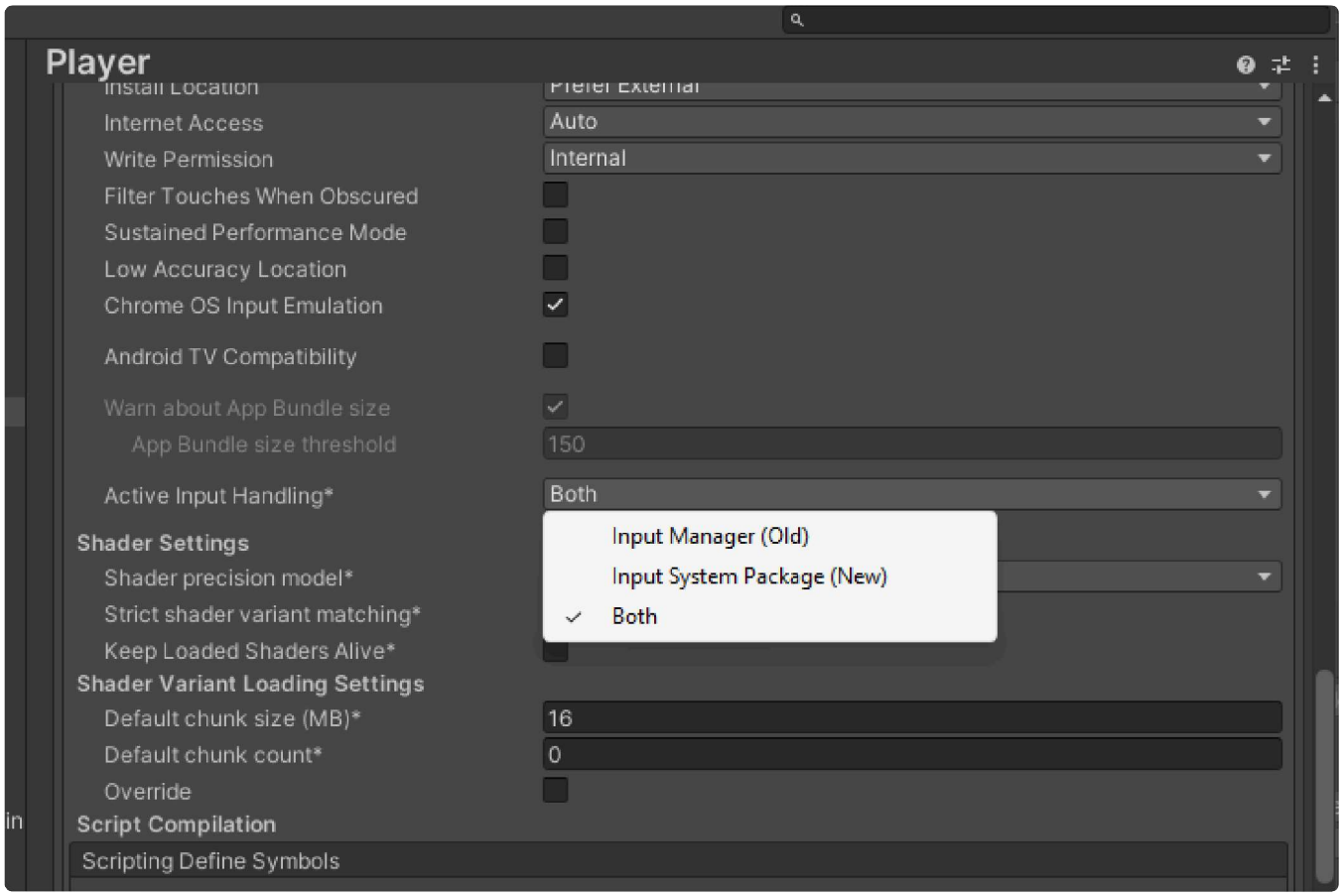


4-

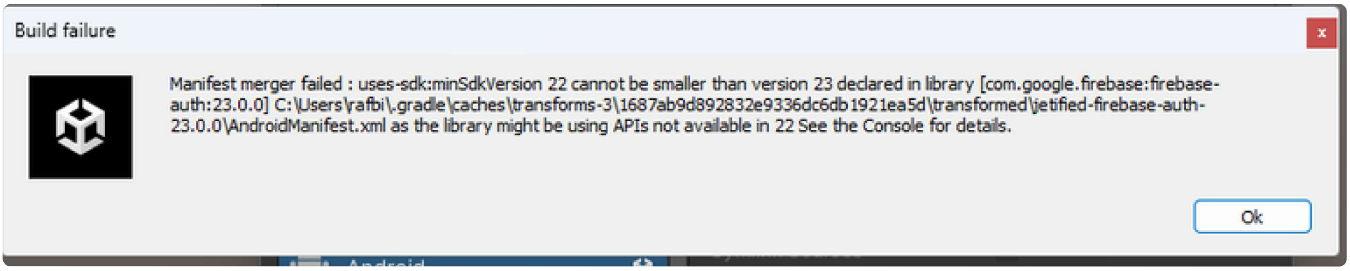


This error occurs because you have installed the Asset New Input System and the system applied both for compatibility, but in Android builds this can lead to errors so the engine issues this warning.

Solution: In the build settings change Both to Input System Package(new).

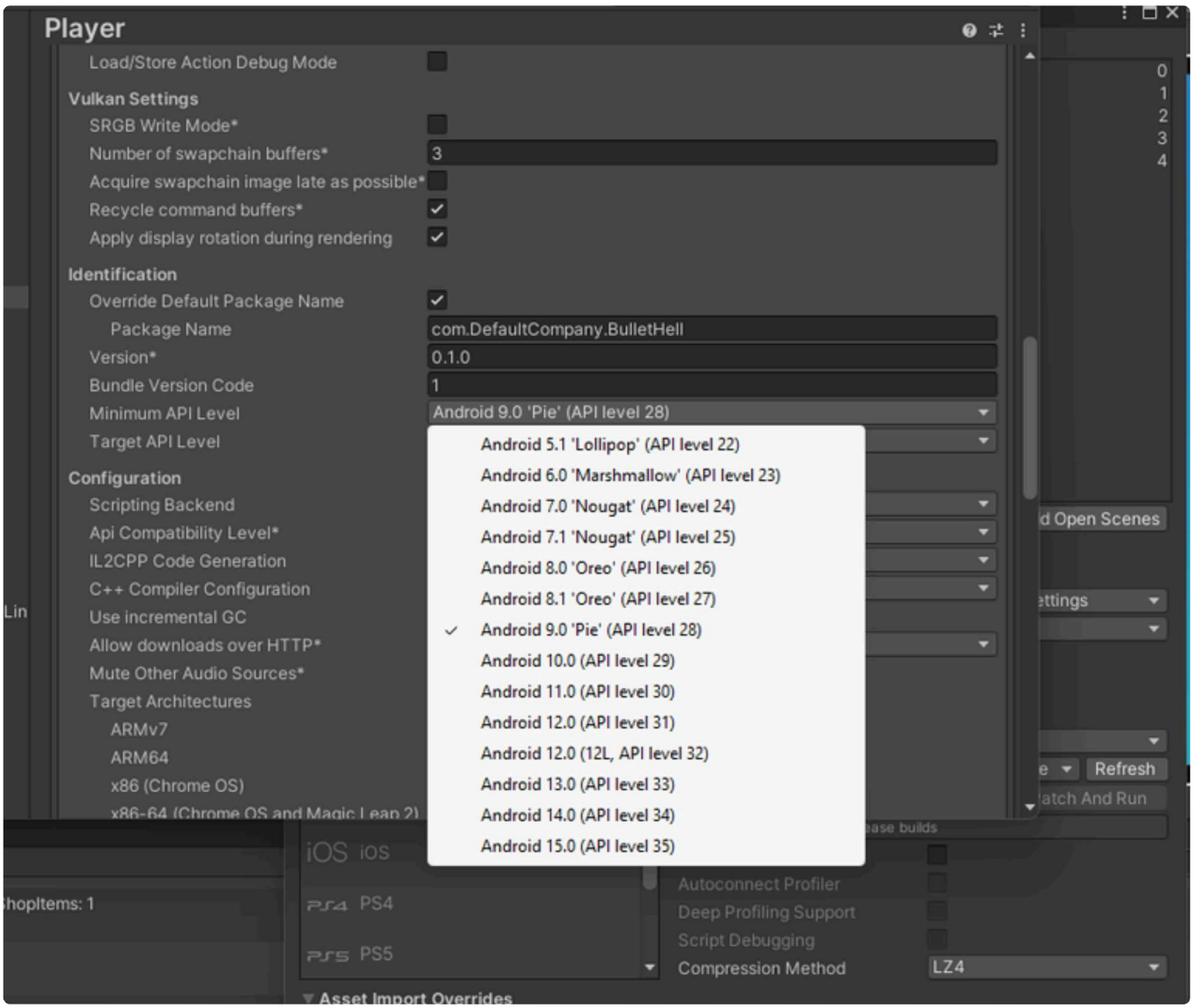


5-



This is the most annoying error and one that discourages new developers trying to use the Firebase SDK because there is no direct warning about what is actually causing it, and it always appears when trying to compile the game build. There are several build errors with large logs and little information about what actually caused the error.

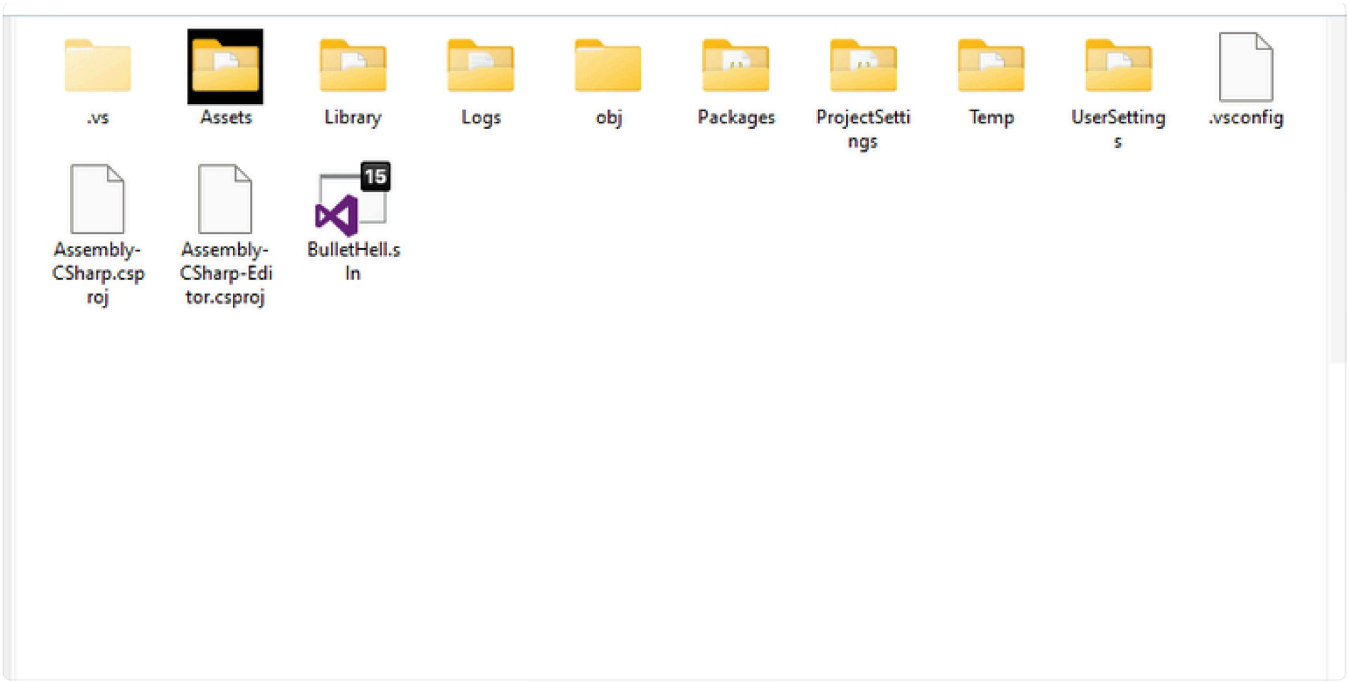
Main cause: The latest Firebase SDKs only support Android versions with a minimum API level of 23 or higher. I recommend using above 28. Change the minimum Android version in the build settings menu

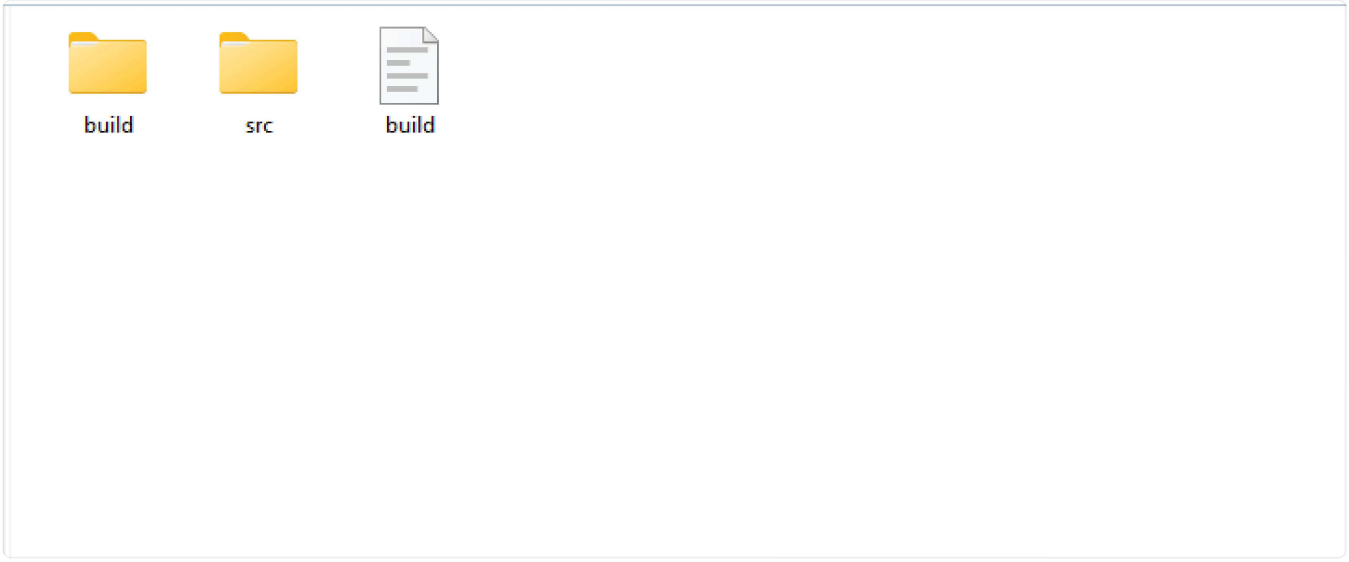


Second cause: When using Android's force resolve, some options in Publishing Settings/Build are automatically marked, but the file was configured for a minimum

Android version that Gradle/SDK does not support.

Solution: Access the root folder of your project (the parent folder of the Assets, library, etc. folders), search for Launcher and select the folder of the current build type of your project (Mono/IL2CPP) and change the Build.Gradle file to an accepted configuration, if you do not know the build type, access the 2 launcher folders if any and change them:





```
android {
    namespace "com.DefaultCompany.BulletHell"
    ndkPath "C:/Program Files/Unity/Hub/Editor/2022.3.45f1/Editor/Data/PlaybackEngines/AndroidPlayer/N

    compileSdkVersion 34
    buildToolsVersion '34.0.0'

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_11
        targetCompatibility JavaVersion.VERSION_11
    }

    defaultConfig {
        minSdkVersion 23
        targetSdkVersion 34
        applicationId 'com.DefaultCompany.BulletHell'
        ndk {
            abiFilters 'armeabi-v7a'
        }
        versionCode 1
        versionName '0.1.0'
    }
```

Other possible solutions: Make sure these boxes are checked in Publish/Build Settings. If not, try checking them and if you are unable to do so or get an error, force-resolve again, accept all previous prompts, go back to Build Settings and try again.