

Unity

Les bases de Unity – Lexique

Table des matières

Unity.....	2
Moteur de Jeux.....	2
Unity Hub.....	2
Éditeur.....	2
Scène.....	2
Hiérarchie.....	2
GameObject.....	2
Inspector.....	3
Component/Script.....	3
Assets.....	3
MonoBehaviour.....	3
C#.....	3
Variables.....	3
Méthodes.....	3
Start().....	4
Update().....	4
Input.....	4
Sprites.....	4
Tilemap.....	4
Animator.....	4
Collider.....	4
Collider trigger.....	4
Rigidbody.....	5
SpriteRenderer.....	5
InputManager.....	5
GetComponent().....	5
Time.deltaTime.....	5
Tag (gameObject).....	5
Layer (gameObject).....	6
Gizmos.....	6
Build Settings.....	6
OnTriggerEnter().....	6
Waypoint.....	6
PreFab.....	6
SpawnPoint.....	7
Debug.Log().....	7
Canvas.....	7

Unity

Unity est un moteur de jeux vidéo très populaire, utilisé pour créer des jeux en 2D, 3D, réalité augmentée (AR) ou réalité virtuelle (VR). Il offre des outils et des services permettant de développer rapidement des jeux pour de nombreuses plateformes (PC, mobile, console, etc.), avec un environnement simple à prendre en main pour les débutants comme pour les professionnels.

Moteur de Jeux

Un moteur de jeux est un ensemble d'outils et de technologies qui facilitent la création de jeux vidéo. Dans Unity, le moteur gère la physique, les graphiques, l'intelligence artificielle et d'autres aspects essentiels du gameplay, permettant aux développeurs de se concentrer sur les éléments créatifs sans avoir à tout coder depuis zéro.

Unity Hub

Unity Hub est une application qui centralise la gestion des différentes versions de Unity et des projets que vous développez. Il permet de créer, ouvrir, et gérer vos projets, de télécharger différentes versions du moteur Unity, et d'accéder à des ressources et didacticiels. C'est l'interface de départ pour tout utilisateur de Unity.

Éditeur

L'éditeur Unity est l'environnement principal où le développement de jeux a lieu. Il inclut différents panneaux comme la scène, la hiérarchie et l'inspecteur, qui permettent de manipuler les objets du jeu, de créer des scripts, d'ajouter des effets, de tester le jeu en temps réel, et d'exporter le jeu une fois terminé.

Scène

Une scène dans Unity est comme une "page" dans un livre de jeu. C'est là où tout le contenu du jeu est placé : les personnages, les objets, la caméra, la lumière, etc. Un jeu peut être composé de plusieurs scènes, comme un niveau de jeu, un menu ou une cinématique.

Hiérarchie

Le panneau de hiérarchie affiche une vue organisée de tous les objets présents dans une scène. Les objets sont représentés sous forme d'arborescence, et chaque élément de la hiérarchie peut être un parent ou un enfant d'un autre objet, ce qui est utile pour organiser les éléments du jeu en groupes logiques.

GameObject

Un GameObject est l'entité de base dans Unity. C'est un conteneur qui peut représenter n'importe quoi dans une scène, comme un personnage, une caméra, une lumière ou un simple cube. Chaque GameObject peut contenir différents composants qui ajoutent des fonctionnalités, comme des scripts pour définir son comportement ou des rendus pour afficher des images.

Inspector

Le panneau Inspector montre les propriétés et paramètres du GameObject sélectionné. Il permet de modifier les composants attachés à l'objet, comme les scripts, la physique, ou l'apparence visuelle. C'est l'endroit où vous ajustez les détails de chaque GameObject pour influencer leur comportement dans le jeu.

Component/Script

Un composant dans Unity est un module qui ajoute des fonctionnalités à un GameObject. Un script est un type particulier de composant, qui est un bout de code écrit en C# et qui définit le comportement spécifique d'un GameObject. Par exemple, un script peut indiquer à un objet de bouger, de détecter des collisions, ou d'interagir avec le joueur.

Assets

Les assets sont les ressources utilisées dans le jeu, comme les modèles 3D, les textures, les sons, les animations, ou les scripts. Unity les stocke dans un dossier spécial du projet, et ils peuvent être importés ou créés directement dans l'éditeur. Les assets sont ensuite assignés à des GameObjects pour créer les éléments visuels et interactifs du jeu.

MonoBehaviour

MonoBehaviour est la classe de base à partir de laquelle tous les scripts Unity dérivent. Lorsqu'un script hérite de MonoBehaviour, il peut être attaché à un GameObject dans Unity et répondre à des événements de jeu comme le démarrage d'un niveau, les mises à jour à chaque image, ou les interactions avec d'autres objets.

C#

C# (prononcé "C-sharp") est le langage de programmation utilisé dans Unity pour écrire des scripts. C'est un langage moderne, orienté objet, qui permet de définir le comportement des objets dans un jeu, d'interagir avec les composants Unity et de contrôler la logique du jeu.

Variables

Les variables sont des conteneurs de données dans un script qui stockent des informations comme les scores, la position d'un joueur, la santé, ou les paramètres d'un objet. Dans Unity, les variables peuvent être définies et modifiées pour ajuster le comportement d'un GameObject.

Méthodes

Les méthodes sont des blocs de code qui effectuent des actions spécifiques. En C#, les méthodes peuvent être appelées pour exécuter une série d'instructions, comme déplacer un objet, jouer un son, ou détecter une collision. Elles permettent de structurer et d'organiser le code.

Start()

Start() est une méthode spéciale dans Unity, appelée une seule fois au début de la vie d'un script ou d'un objet dans une scène. C'est souvent utilisé pour initialiser les variables ou définir l'état de départ d'un GameObject.

Update()

Update() est une méthode appelée à chaque frame (image) du jeu, tant que le script est actif. Elle est utilisée pour gérer des actions qui doivent être mises à jour régulièrement, comme le déplacement d'un personnage ou la détection des entrées de l'utilisateur.

Input

Input fait référence aux interactions de l'utilisateur avec le jeu, comme l'appui sur une touche, le mouvement de la souris, ou l'utilisation d'une manette. Unity fournit des classes spéciales pour capturer ces interactions et permettre au jeu de réagir aux actions des joueurs.

Sprites

Les sprites sont des images 2D utilisées pour représenter des objets dans les jeux 2D. Ils peuvent être des personnages, des objets, ou des éléments de décor. Unity dispose d'outils pour gérer les sprites, comme les animer ou les modifier directement dans l'éditeur.

Tilemap

Tilemap est un système dans Unity qui permet de créer des mondes en 2D en disposant des tuiles (ou blocs) sur une grille. Il est couramment utilisé pour les jeux de plateformes ou les RPG, où les niveaux peuvent être construits rapidement à partir de petites images répétitives.

Animator

L'Animator est un composant qui contrôle les animations dans Unity. Il permet de définir des transitions entre différentes animations (par exemple, marcher, sauter, courir), et de les relier à des événements ou des conditions spécifiques dans le jeu.

Collider

Un Collider est un composant ajouté à un GameObject pour gérer les collisions physiques dans Unity. Il définit la forme de l'objet pour que le moteur de physique sache quand cet objet entre en contact avec un autre, mais n'ajoute pas de mouvement à lui seul.

Collider trigger

Un Collider Trigger est un type spécial de Collider qui ne bloque pas les objets, mais détecte simplement quand un autre objet entre dans sa zone. Il est souvent utilisé pour déclencher des événements dans un jeu, comme ouvrir une porte ou démarrer une cinématique.

Rigidbody

Le Rigidbody est un composant utilisé pour donner à un GameObject un comportement physique réaliste dans Unity. Il permet à l'objet d'être affecté par des forces comme la gravité et les collisions. Les objets avec un Rigidbody peuvent être déplacés de manière plus naturelle grâce à la simulation de la physique.

SpriteRenderer

Le SpriteRenderer est un composant qui permet d'afficher des sprites 2D dans Unity. Il définit comment un sprite est rendu à l'écran, y compris ses propriétés comme la couleur, le tri par ordre de rendu, et les effets de transparence. Grâce à ce composant, les développeurs peuvent facilement gérer l'affichage des graphiques 2D dans leurs jeux.

InputManager

L'InputManager est un système de gestion des entrées dans Unity, permettant de définir les actions que les joueurs peuvent effectuer via le clavier, la souris ou une manette. Il permet de créer des mappages d'entrée personnalisés pour gérer les différentes commandes et de centraliser la gestion des interactions utilisateur.

GetComponent()

GetComponent() est une méthode utilisée dans les scripts Unity pour accéder à un autre composant attaché à un GameObject. Par exemple, si un script a besoin d'accéder aux propriétés d'un Rigidbody ou d'un SpriteRenderer, il peut appeler GetComponent<Rigidbody>() pour obtenir ce composant. Cela permet une meilleure organisation et modularité dans le code.

Time.deltaTime

Time.deltaTime est une propriété qui renvoie le temps écoulé depuis la dernière frame, en secondes. En utilisant Time.deltaTime, les développeurs peuvent rendre les mouvements et les animations indépendants de la fréquence d'images, assurant ainsi que le gameplay reste fluide et cohérent sur toutes les plateformes, peu importe le matériel.

Tag (gameObject)

Les Tags sont des étiquettes attribuées aux GameObjects dans Unity, permettant de les identifier facilement dans le code. Par exemple, un GameObject peut être étiqueté comme "Player" ou "Enemy". Les développeurs peuvent utiliser ces tags pour effectuer des opérations spécifiques, comme détecter des collisions ou interagir avec d'autres objets basés sur leur étiquette.

Layer (gameObject)

Les Layers sont des catégories utilisées pour organiser les GameObjects dans Unity. Chaque GameObject peut être assigné à une Layer, permettant ainsi de contrôler la manière dont les objets interagissent avec d'autres objets dans le jeu, en particulier pour les systèmes de collision et le rendu. Par exemple, on peut créer des Layers spécifiques pour les objets qui doivent être ignorés par certaines caméras.

Gizmos

Les Gizmos sont des outils visuels qui aident les développeurs à visualiser des éléments dans l'éditeur Unity. Ils ne sont pas visibles lors de l'exécution du jeu, mais permettent de dessiner des formes ou des lignes pour représenter des objets, des trajectoires, ou des zones de détection. Les Gizmos facilitent la conception et le débogage des scènes.

Build Settings

Build Settings est une fenêtre dans Unity qui permet de configurer les paramètres de construction pour un projet. Les développeurs peuvent choisir la plateforme cible (comme Windows, macOS, Android, etc.), ajouter des scènes à inclure dans la construction, et configurer d'autres options pour personnaliser le comportement du jeu lors de son déploiement.

OnTriggerEnter()

OnTriggerEnter() est une méthode qui est appelée lorsqu'un Collider avec le paramètre "Is Trigger" activé entre en contact avec un autre Collider. Cela permet aux développeurs de déclencher des événements lorsqu'un objet entre dans une zone définie, comme activer un événement ou lancer une animation.

Waypoint

Les Waypoints sont des points prédéfinis dans un espace de jeu que les objets, comme les ennemis ou les véhicules, peuvent utiliser pour naviguer. En programmant un GameObject pour se déplacer d'un waypoint à un autre, les développeurs peuvent créer des chemins de mouvement intelligents, comme des patrouilles d'ennemis.

Prefab

Un Prefab est un modèle de GameObject sauvegardé dans Unity, permettant de créer des instances de cet objet facilement. Les Prefabs peuvent contenir des composants, des propriétés et des comportements, ce qui permet aux développeurs de réutiliser des objets sans avoir à les recréer à chaque fois. Cela simplifie la gestion et l'organisation des assets dans un projet.

SpawnPoint

Un SpawnPoint est un emplacement spécifique dans le jeu où un nouvel objet ou un personnage peut apparaître (ou "spawner"). Les SpawnPoints sont souvent utilisés pour faire réapparaître des ennemis, générer des items, ou introduire des personnages au début d'un niveau.

Debug.Log()

Debug.Log() est une méthode utilisée pour afficher des messages dans la console d'Unity. Cela aide les développeurs à déboguer leurs scripts en leur permettant de suivre l'exécution du code, de vérifier les valeurs des variables ou de comprendre le flux logique de leur programme. Les messages apparaissent pendant le mode de jeu et peuvent être très utiles pour le diagnostic.

Canvas

Le Canvas est un composant essentiel dans Unity pour le développement d'interfaces utilisateur (UI). Il sert de conteneur pour tous les éléments UI, tels que les boutons, les images, et les textes. Le Canvas gère le rendu et l'organisation de ces éléments dans l'espace de jeu, et il permet aux développeurs de concevoir des interfaces réactives et adaptatives.