Roadmap

Objectif final : Devenir game developer

Durée de la roadmap : 2 à 3 ans

Compétences clés à maîtriser :

1. **Programmation orientée jeu** (C++, C# avec Unity, optimisation)
2. **Moteurs de jeu avancés** (Unreal Engine, Unity)
3. **Systèmes d'intelligence artificielle (IA) et physique**
4. **Conception et architecture de jeux vidéo**
5. **Multiplayer et networking**
6. **Collaboration avec des équipes multidisciplinaires (artistes, designers, etc.)**
7. **Soft skills** : Gestion de projet, collaboration, créativité, résolution de problèmes

---

Roadmap détaillée :

0 à 6 mois : Consolidation des bases et spécialisation

1. **C++ (niveau débutant à intermédiaire)**
    - **Pourquoi** : Le C++ est le langage le plus utilisé pour le développement de jeux vidéo (notamment avec Unreal Engine). Il te faudra bien maîtriser la programmation orientée objet, la gestion de la mémoire, et l'optimisation des performances.
    - **Actions** :
        - Suivre un cours intensif en C++ (Codecademy, Udemy, ou OpenClassrooms).
        - Projets pratiques : Écrire des petits programmes en C++ pour comprendre la gestion de la mémoire et l'optimisation.
        - Développer un petit jeu en C++ avec un moteur comme **SFML** ou **SDL**.
    - **Objectif** : Avoir une base solide en C++ pour commencer à travailler sur des projets de jeu complexes.
2. **Approfondissement Unreal Engine**
    - **Pourquoi** : Tu as déjà des bases en Unreal Engine, il est temps de les renforcer. Unreal est très utilisé pour les jeux AAA.

- **Actions** :
  - Suivre des tutos avancés sur Unreal Engine (création de mondes 3D, optimisation, blueprints et scripting en C++).
  - Travailler sur des petits projets personnels (ex. jeux de plateforme, simulations simples).
- **Objectif** : Maîtriser les **blueprints** et commencer à coder en C++ dans Unreal Engine.

3. **Conception de systèmes de jeu**
   - **Pourquoi** : Il est important de comprendre la logique derrière la conception de systèmes dans les jeux (systèmes de combat, inventaire, quêtes, etc.).
   - **Actions** :
     - Lire des ressources ou suivre des cours sur la **game design theory** et la conception des systèmes.
     - Créer un prototype de système de jeu (inventaire, combat, etc.) dans Unreal ou Unity.
   - **Objectif** : Concevoir et coder des systèmes basiques pour des jeux vidéo.

---

6 à 12 mois : Développement de projets complexes et nouvelles technologies

1. **Unity + C# (niveau débutant à intermédiaire)**
   - **Pourquoi** : Unity est un autre moteur très populaire dans l'industrie, et C# est essentiel pour les projets Unity.
   - **Actions** :
     - Suivre des cours sur Unity et C# (Unity Learn, Udemy).
     - Réaliser un projet simple dans Unity (jeu 2D ou 3D basique).
   - **Objectif** : Avoir un projet fonctionnel sous Unity pour ton portfolio.
2. **Systèmes d'intelligence artificielle pour jeux vidéo**
   - **Pourquoi** : Les IA sont des composantes cruciales en jeux vidéos (NPC, ennemis, etc.).
   - **Actions** :
     - Apprendre les bases de l'IA dans les jeux (pathfinding, comportement d'ennemis, etc.) avec *A algorithm**, **FSM (finite state machines)**.
     - Créer un petit projet d'IA pour un jeu.
   - **Objectif** : Créer un système d'IA basique pour un NPC dans un petit projet Unreal ou Unity.

3. **Développement Multijoueur et Réseaux**
    - **Pourquoi** : Beaucoup de jeux AAA ont des fonctionnalités multijoueurs. Il est important de comprendre comment fonctionne le **networking** dans un jeu vidéo.
    - **Actions** :
        - Apprendre les bases du **networking** dans les jeux (sockets, client-serveur, synchronisation).
        - Suivre des tutoriels sur le multijoueur dans Unreal Engine ou Unity.
    - **Objectif** : Créer une démo multijoueur basique dans Unreal Engine ou Unity (par exemple, un simple jeu où deux joueurs peuvent interagir).

---

12 à 18 mois : Création de projets personnels et perfectionnement

1. **Création d'un jeu complet**
    - **Pourquoi** : Créer un projet de jeu complet te permettra de montrer tes compétences dans ton portfolio.
    - **Actions** :
        - Développer un jeu complet (du concept au produit final) en utilisant **Unreal Engine** ou **Unity**. Ce jeu pourrait être une simulation, un jeu de plateforme, un RPG simple, etc.
        - Travailler sur l'optimisation des performances, l'IA, le système de jeu, et les interactions multijoueur (si applicable).
    - **Objectif** : Avoir un projet de jeu prêt à être montré à des recruteurs.
2. **Systèmes physiques avancés (Physics Engines)**
    - **Pourquoi** : Comprendre les moteurs physiques est crucial pour les jeux modernes, surtout dans les simulations ou les jeux d'action.
    - **Actions** :
        - Suivre des cours ou lire des ressources sur les **Physics Engines** (moteurs de physique dans Unreal/Unity).
        - Appliquer les connaissances en créant des interactions physiques dans un projet personnel.
    - **Objectif** : Intégrer des systèmes physiques réalistes dans tes jeux.
3. **Collaboration en équipe (projets open-source ou game jams)**
    - **Pourquoi** : Les grandes entreprises cherchent des développeurs capables de travailler en équipe.
    - **Actions** :

        - Participer à des **game jams** (Ludum Dare, Global Game Jam).
        - Contribuer à des projets open-source ou collaboratifs pour développer des compétences de travail d'équipe.
    - **Objectif** : Acquérir une expérience collaborative en développement de jeux.

---

18 à 24 mois : Spécialisation et préparation à l'industrie

1. **Optimisation de jeux pour les performances (Profiling)**
    - **Pourquoi** : Les jeux AAA nécessitent des optimisations poussées pour tourner efficacement sur différents supports.
    - **Actions** :
        - Apprendre le **profiling** et l'optimisation dans Unreal ou Unity (optimisation des ressources, gestion de la mémoire, rendu, etc.).
    - **Objectif** : Optimiser un projet pour réduire les temps de chargement et améliorer les performances globales.
2. **Participation à des projets collaboratifs ou open-source**
    - **Pourquoi** : Montrer ta capacité à travailler dans une équipe et gérer un projet complexe.
    - **Actions** :
        - Participer activement à des projets **open-source** ou travailler avec une équipe sur un projet collaboratif (via GitHub par exemple).
    - **Objectif** : Avoir des contributions significatives sur des projets en équipe.
3. **Portfolio final et networking**
    - **Pourquoi** : Un bon portfolio est essentiel pour postuler dans des entreprises comme Ubisoft ou EA.
    - **Actions** :
        - Rassembler tous tes projets dans un **portfolio en ligne** (jeux, prototypes, systèmes, IA, multijoueur, etc.).
        - Participer à des événements de l'industrie du jeu (conférences, salons) et à des communautés en ligne (forums, LinkedIn, Twitter) pour élargir ton réseau.
    - **Objectif** : Avoir un portfolio attractif pour postuler dans des entreprises AAA et établir des connexions professionnelles.