CHECKLIST DE DÉCISION IMPRIMABLE

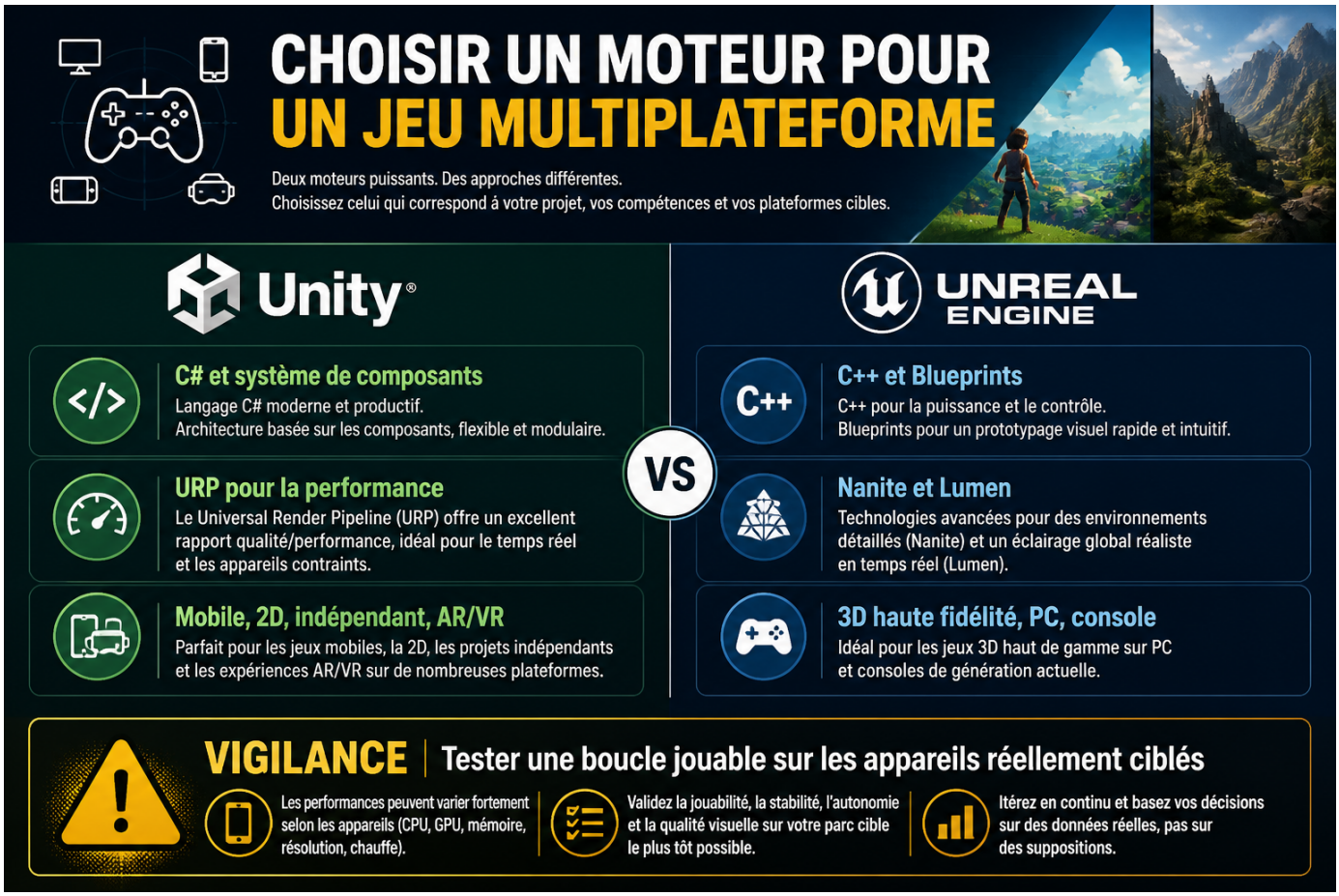
Choisir Unity ou Unreal Engine sans compromettre votre portage

Utilisez cette fiche avant de verrouiller votre moteur de jeu. Elle vous aide à confronter votre ambition créative aux contraintes réelles du multiplateforme : appareils, performances, équipe et maintenance.

Au sommaire

1. Définir le jeu avant de choisir le moteur

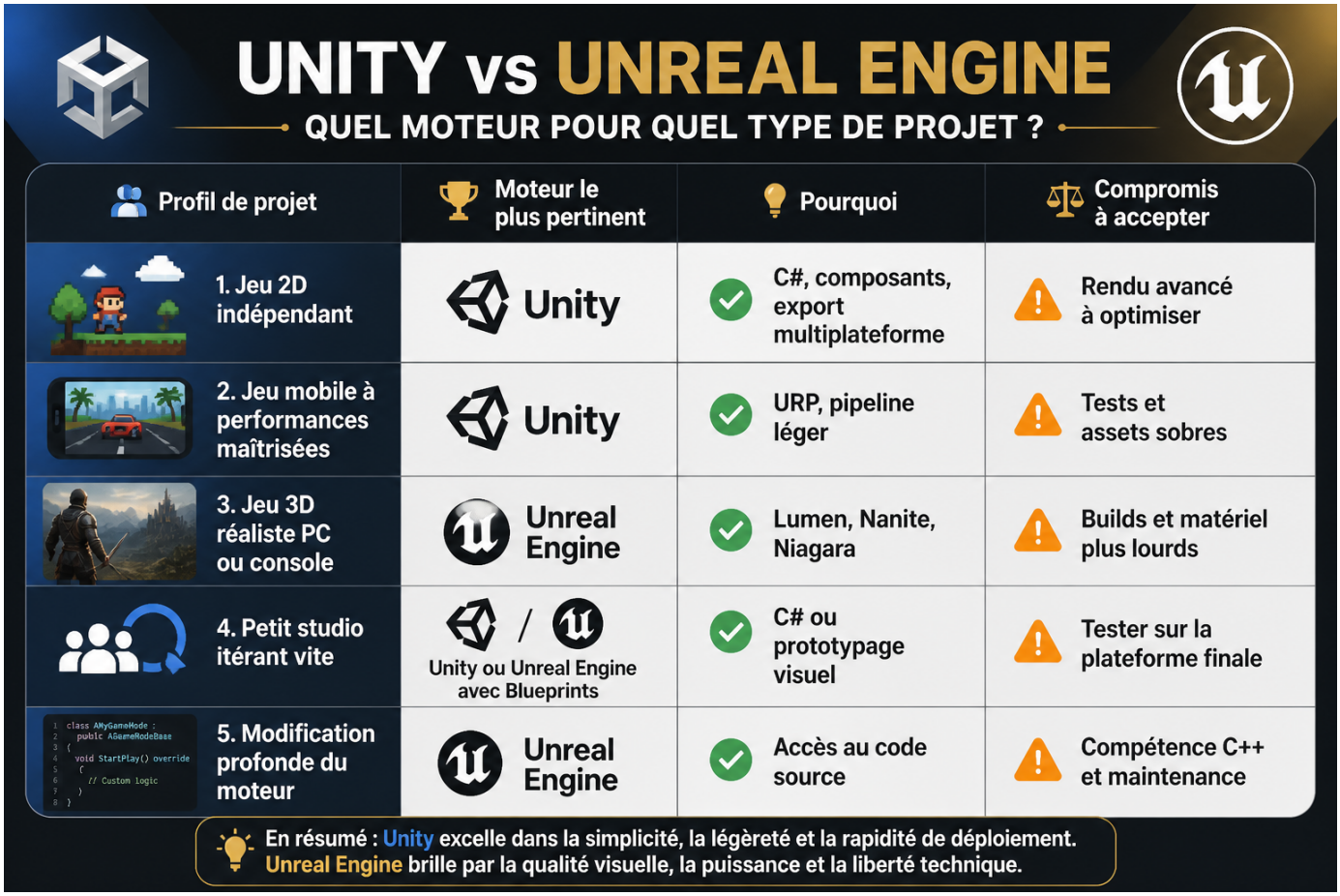
- [ ] **Décrire la boucle de jeu principale en quelques lignes**
  Le moteur doit servir le gameplay prioritaire, pas seulement l'apparence d'une démo.
- [ ] **Lister les plateformes visées au lancement et après lancement**
  PC, mobile et consoles n'impliquent pas les mêmes contrôles, interfaces ni contraintes techniques.
- [ ] **Identifier le niveau visuel indispensable au projet**
  Distinguer une 3D stylisée ou fonctionnelle d'un rendu réaliste qui porte l'expérience.
- [ ] **Évaluer les compétences disponibles dans l'équipe**
  C#, C++, Blueprints, optimisation, production d'assets et outils internes.
- [ ] **Prévoir la maintenance après publication**
  Correctifs, builds, mises à jour, tests et évolutions doivent entrer dans la décision.



INFOGRAPHIE *Infographie comparaison Unity et Unreal Engine pour publication de jeu multiplateforme*

2. Repère rapide selon le profil de projet

Votre priorité	Moteur souvent pertinent	Point de vigilance
Jeu 2D indépendant	Unity	Prévoir une direction artistique et une optimisation explicites pour un rendu avancé.
Jeu mobile sur des appareils variés	Unity	Tester régulièrement les appareils modestes et garder des assets sobres.
Jeu 3D réaliste sur PC ou console	Unreal Engine	Anticiper des builds, des assets et une configuration de travail potentiellement plus lourds.
Prototype à faire évoluer rapidement	Unity ou Unreal Engine avec Blueprints	Valider le prototype sur la plateforme finale avant de figer le choix.
Modification profonde du moteur	Unreal Engine	La personnalisation du moteur demande des compétences C++ et une maintenance durable.



INFOGRAPHIE *Repères visuels pour relier le type de projet à l'orientation de chaque moteur.*

3. Vérifier la réalité du multiplateforme

- **Tester les contrôles sur chaque mode d'entrée ciblé**
  Tactile, manette et clavier-souris demandent des adaptations dédiées.
- **Adapter l'interface à chaque écran**
  Ne pas supposer qu'une interface PC fonctionne telle quelle sur mobile ou console.

Mesurer mémoire, temps de chargement et performances sur les appareils réels
Les budgets techniques doivent être définis et vérifiés pendant la production.

Préparer des builds réguliers par plateforme
Un export ne remplace pas les vérifications sur le matériel ciblé.

Intégrer les exigences de sauvegarde, de boutique et de certification
Ces contraintes font partie du travail de publication, pas d'une étape secondaire.

4. Les compromis à assumer

Unity s'inscrit souvent dans un workflow plus léger pour les jeux mobiles, 2D, indépendants et les projets où l'itération rapide est centrale.

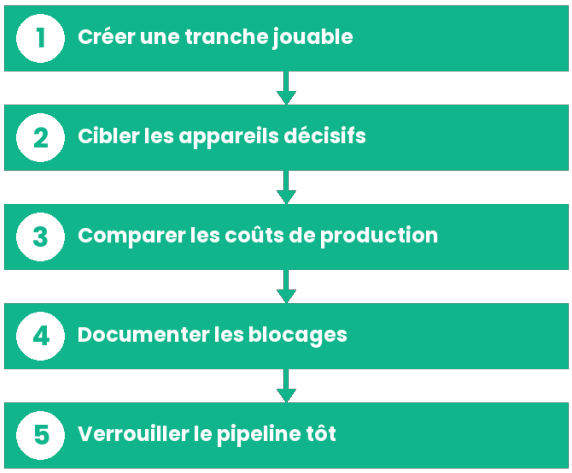
Les outils visuels d'Unreal Engine, comme Blueprints, accélèrent le prototypage mais ne suppriment pas les besoins d'optimisation ni de tests.

Le meilleur moteur est celui qui permet à votre équipe de livrer et de maintenir la boucle jouable prévue sur toutes les plateformes annoncées.

Unreal Engine est particulièrement adapté lorsque la 3D haute fidélité, les environnements détaillés, les effets avancés et la mise en scène sont au cœur du projet.

Les outils graphiques avancés ne dispensent jamais de gérer la consommation mémoire, la taille des assets et les performances sur les appareils limités.

5. Prendre une décision fondée sur un prototype



1 **Créer une tranche jouable**
Construisez une version courte avec un personnage, une interaction, une caméra, une interface et le niveau de rendu représentatif du jeu final.

2 **Cibler les appareils décisifs**
Faites fonctionner cette tranche sur les plateformes réellement visées, plutôt que de juger uniquement dans l'éditeur.

3 **Comparer les coûts de production**
Observez le temps de développement, le poids des assets, la fluidité, les temps de chargement et la complexité des adaptations.

4 **Documenter les blocages**
Notez ce qui ralentit l'équipe : programmation, pipeline artistique, outils, machines de développement ou déploiement.

5 **Verrouiller le pipeline tôt**
Une fois le moteur et l'orientation de rendu validés, évitez les changements tardifs qui alourdissent la production.

! **Licences : vérification obligatoire avant commercialisation**
Les conditions d'utilisation et de revenus évoluent. Vérifiez les conditions de licence officielles de Unity et d'Epic Games avant de publier ou de budgéter votre projet.

Cette fiche est un outil de cadrage : la validation finale doit reposer sur un prototype testé sur les plateformes réellement ciblées.