

# Rapport IHM

Critère	Electron	Godot	Unity
Facilité d'apprentissage	✅ Facile (JS, HTML)	✅ Facile (GDScript)	❌ Plus complexe (C#)
Affichage graphique	❌ Pas optimisé (pas de moteur natif)	✅ Adapté pour la 2D	✅ Adapté, mais plus lourd
Multijoueur	✅ WebSockets (Node.js)	✅ WebSockets / ENet	✅ Mirror / Photon (mais plus complexe)
Déploiement	✅ Web, Desktop	✅ Web, Mobile, Desktop	✅ Web, Mobile, Desktop
Performance	❌ Très lourd (Chrome intégré)	✅ Léger et rapide	🟡 Lourd pour un simple jeu de cartes

Après mes recherches en ligne j'ai trouvé que Godot est la meilleure option pour l'IHM. Son langage est GDScript (même possible d'utiliser C++). En exportant Godot on obtient des fichiers HTML5 pour le web et des applications pour le desktop. Donc Godot est facilement adapté et simple pour le web et le desktop. De même sa performance est légère. Il est meilleur que PyGame car PyGame est difficile pour le déploiement et sa performance est moins optimisée que Godot.

## GPT ANSWER

🎯 **Godot est le meilleur choix pour l'UI !**

Pourquoi ?

- ✅ **Tout est intégré** → Pas besoin de HTML/CSS/JS.
- ✅ **Éditeur visuel pour l'UI** → Facile à positionner les cartes et les boutons.

✓ **Adapté aux jeux** → Meilleure gestion des animations et des interactions qu'un simple site web.

✓ **Export facile** → Web (HTML5), Desktop (Windows/Linux/Mac), Mobile (Android/iOS).

---

## 🔥 1 Pourquoi Godot UI est meilleur que HTML/CSS pour ton projet ?

Si tu fais un jeu **avec Electron** (HTML + CSS + JS), l'UI ressemblera à un **site web**, ce qui n'est **pas naturel pour un jeu de cartes**.

Avec **Godot**, tu peux :

- 📌 **Positionner les cartes visuellement** avec le **CanvasLayer**.
- 📌 **Créer facilement des animations et transitions** entre les tours.
- 📌 **Gérer le multijoueur et les interactions dans le même environnement**.

💡 **Pas besoin de jongler entre HTML/CSS/JS et Python !**

---

## 🎨 2 Comment structurer l'UI de "6 qui prend!" dans Godot ?

Godot propose plusieurs **Nodes UI (Control Nodes)** :

Élément du jeu	Node UI dans Godot
Fond de l'écran	<code>TextureRect</code>
Plateau de jeu	<code>VBoxContainer</code> / <code>GridContainer</code>
Cartes	<code>TextureButton</code> ou <code>TextureRect</code>
Boutons ("Jouer", "Passer")	<code>Button</code>
Score des joueurs	<code>Label</code>
Liste des joueurs	<code>HBoxContainer</code> + <code>Label</code>

📌 **Les `Containers` permettent d'aligner automatiquement les cartes et les éléments de l'interface.**

---

🏆 **Conclusion : Godot est parfait pour l'UI de ton jeu !**

💡 **Pourquoi ?**

- ✓ **Éditeur visuel** = Pas besoin de CSS/HTML, tout est **positionné visuellement**.
- ✓ **Interface dynamique et responsive** avec `Containers`.
- ✓ **Animations et interactions fluides** sans code externe.
- ✓ **Support direct du multijoueur** pour gérer les tours des joueurs.
- ✓ **Déploiement Web, Desktop et Mobile en un clic**.
- 👉 Tu veux un guide pour structurer l'UI complète dans Godot ? 😊