

## UX UI HACKNPLAN — COMMENT ATTAQUER LE PROJET INTELLIGEMENT

HacknPlan est l'outil qui vous sert à **organiser, structurer, documenter** et **suivre** la production de votre jeu. Dans le cadre d'un projet de game development, sa logique centrale repose sur des boards, des tasks, des user stories, des milestones, et sur un **Game Design Model**, c'est-à-dire une structure qui permet de relier le design et la production réelle.

### 1. Ne créez pas un projet vague

La pire erreur est de créer des tâches floues comme :

“faire l'UI”  
“faire le menu”  
“faire le gameplay”  
“faire l'art”

Ce type d'intitulé n'aide personne.  
Ce n'est ni mesurable, ni répartissable, ni pilotable.

Il faut découper le projet en éléments compréhensibles, concrets et suivables.

### 2. Commencez par structurer le jeu en grands blocs

Dans HacknPlan, commencez par définir les grandes parties du jeu.

Par exemple :

#### Game Design / Conception

- concept du jeu
- boucle de gameplay
- pitch
- conditions de victoire et d'échec
- progression

#### Art Direction / Direction artistique

- moodboard
- style visuel
- palette couleur
- personnages
- décors
- props
- FX

#### Development / Développement

- 3C
- mécaniques principales
- IA
- systèmes
- UI
- contrôles
- debug

## UI / UX

- HUD
- menu principal
- menu pause
- prompts d'interaction
- navigation manette
- feedback
- accessibilité

HacknPlan permet précisément de relier ces éléments à de vraies tâches de production plutôt que de garder un GDD statique séparé du travail réel.

### 3. Utilisez le Game Design Model pour clarifier le projet

Le **Game Design Model** est utile pour organiser le jeu comme une structure de design, au lieu de simplement empiler des tâches.

Par exemple, vous pouvez créer des éléments comme :

- Core Loop
- Main Mechanic
- HUD
- Main Menu
- Pause Menu
- Inventory
- Tutorial Prompts
- Controller Navigation
- Accessibility

L'intérêt est simple : chaque élément de design peut ensuite être lié à des user stories, des tâches, des pièces jointes, et à l'avancement réel du projet. Cela permet de garder un lien clair entre l'idée et sa fabrication.

### 4. Travaillez avec des user stories, pas seulement des tâches isolées

Une **user story** décrit un besoin ou une fonctionnalité importante.  
Les tâches détaillent ensuite le travail concret.

Exemple :

#### User Story

Le joueur peut naviguer dans le menu pause à la manette sans confusion.

#### Tâches liées

- construire la structure du menu
- définir l'ordre du focus
- rendre le focus visible
- ajouter la logique de retour
- tester Escape / bouton B
- vérifier la lisibilité
- tester à différentes résolutions

C'est beaucoup plus utile que d'écrire seulement :  
"faire menu pause".

HacknPlan présente justement les user stories comme des éléments de niveau supérieur contenant des child tasks plus concrètes.

## 5. Pour la UI / UX, découpez toujours en éléments distincts

Ne laissez jamais toute l'interface dans une seule tâche.

Créez au minimum des éléments séparés pour :

- HUD
- Main Menu
- Pause Menu
- Settings
- Inventory
- Tutorial / Prompts
- Controller Focus
- Back Logic
- Feedback states
- Accessibility checks

Ensuite, chaque partie peut avoir ses propres stories et ses propres tâches.

Exemple :

### HUD

- définir les informations primaires
- tester la lisibilité
- vérifier l'ancrage écran
- vérifier la hiérarchie visuelle

### Controller Navigation

- définir le focus par défaut
- vérifier les directions
- vérifier la sortie de menu
- vérifier les cas d'erreur

### Accessibility

- vérifier taille du texte
- vérifier contraste
- vérifier lisibilité à distance
- éviter de dépendre uniquement de la couleur

## 6. Attachez vos références directement aux bons endroits

HacknPlan permet d'ajouter des pièces jointes et des références directement dans les éléments et dans les tâches.

Vous pouvez donc y lier :

- moodboards
- wireframes
- captures Unreal
- références visuelles
- iconographie
- tests d'interface
- screenshots d'UMG
- notes de navigation
- schémas de menu flow.

Cela évite que les références soient perdues dans des dossiers ou dans une conversation.

La règle simple :

**Chaque élément important du jeu doit avoir ses références au même endroit que ses tâches.**

## 7. Faites des sprints lisibles

Chaque sprint doit répondre à une question simple :

**Qu'est-ce qui doit être vrai à la fin de cette période ?**

Par exemple :

### Sprint préproduction

- concept choisi
- boucle claire
- moodboard validé
- première direction UI / UX
- premières tâches réparties

### Sprint prototype

- 3C jouable
- une ou deux mécaniques fonctionnelles
- HUD basique
- prompts de base
- premiers tests de contrôle

### Sprint alpha

- structure jouable complète
- placeholder UI en place
- navigation fonctionnelle
- premiers tests de lisibilité

### Sprint bêta

- intégration plus propre
- cohérence visuelle
- feedback
- correction des bugs
- amélioration de la navigation

## 8. Une bonne tâche doit être précise

Une bonne tâche doit être :

- claire
- attribuée
- limitée
- vérifiable
- liée à un objectif

Mauvaise tâche :

- "faire l'UI"

Bonne tâche :

- "Créer le widget de barre de vie"
- "Définir l'ordre du focus du menu pause"
- "Tester le HUD en 21:9 et 4:3"
- "Ajouter l'état Hovered / Focused / Pressed du bouton principal"
- "Vérifier la lisibilité du texte à distance"

## 9. Pensez production, pas seulement idée

HacknPlan doit vous aider à répondre très tôt à ces questions :

- Qu'est-ce qui est faisable dans le temps ?

- Qu'est-ce qui est trop ambitieux ?
- Qu'est-ce qui doit être prototypé d'abord ?
- Qu'est-ce qui peut être simplifié tout de suite ?
- Qu'est-ce qui est essentiel ?
- Qu'est-ce qui peut attendre ?

L'objectif n'est pas seulement d'avoir de bonnes idées.  
L'objectif est de transformer une idée en production réelle.

### **10. Pour la UI / UX, pensez système**

L'interface ne doit pas être pensée comme une série d'écrans indépendants.

Elle doit être pensée comme un système :

- composants réutilisables
- états cohérents
- feedback constant
- navigation logique
- comportement stable
- structure répliquable

Si vous changez un bouton important, ce changement doit idéalement améliorer tout le système, pas seulement une page.

### **11. Vérifiez très tôt les points qui cassent le plus souvent**

Pour éviter de perdre du temps plus tard, créez des tâches de vérification dès le début pour :

- anchors
- safe zones
- focus navigation
- back logic
- DPI scaling
- controller readability
- menu flow
- feedback states
- text contrast
- readability at distance

Ces points ne sont pas du polish final.  
Ce sont des bases structurelles.

### **12. Ce que HacknPlan doit vous apporter**

À la fin, HacknPlan doit vous permettre de voir clairement :

- ce que le jeu est censé devenir
- qui fait quoi
- ce qui a été fait
- ce qui bloque
- ce qui manque encore
- comment le design et la production sont liés

Si votre HacknPlan est clair, votre production le sera beaucoup plus aussi.

### **RÈGLE SIMPLE**

Ne créez pas un projet flou.  
Créez une structure claire.

Ne créez pas une tâche “faire le jeu”.

Créez des éléments concrets, des user stories utiles, et des tâches vérifiables.

Ne laissez pas la UI / UX devenir une pensée de dernière minute.

Intégrez-la dès le départ dans le design, dans la production et dans HacknPlan.