

decharge

PIERRE HENNECART

MAXIME GINDA

BENJAMIN VILLA



SOMMAIRE

Introduction

Conception

Les Interfaces

Phase Exploration

Quêtes et Cinématiques

Phase de Combat

Conclusion

INTRODUCTION

Nos motivations et idées

NOS MOTIVATIONS

- Créer un jeu vidéo
- Apprendre à utiliser un moteur de jeu tel que Unity
- Collaborer avec des artistes sur le plan graphique et sonore

NOS IDÉES

- Point'n'Click d'aventure (à la base)
- Jeu de rôle d'action-aventure avec combat de cartes (au final)

SYNOPSIS

Plongé dans Oneiros pendant son sommeil, le joueur devra enquêter et survivre aux diverses entités maléfiques de ce monde afin de retrouver sa liberté

CONCEPTION

Game design, inspirations et outils

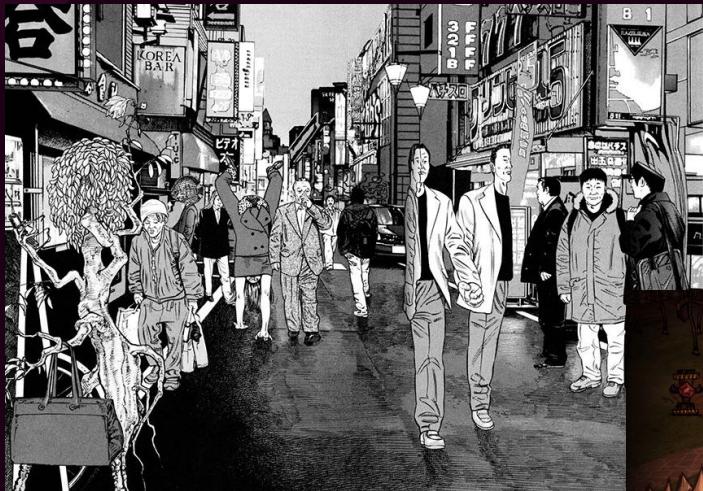
GAME DESIGN - PHASE EXPLORATION

- Dans Oneiros et le monde réel
- Déplacement et interaction avec l'environnement
- Possibilité de récupérer des objets

GAME DESIGN - PHASE COMBAT

- Dans Oneiros seulement
- Combat au tour par tour avec des cartes à effets
- Victoire lorsque l'ennemi n'a plus de vie

NOS INSPIRATIONS ARTISTIQUES



LES OUTILS UTILISÉS

UNITY



GITHUB



TRELLO



..ainsi que wombo.art

LES INTERFACES

Menus, inventaire et suivi de quêtes

PLUSIEURS INTERFACES

- Menu principal
- Menu pause
- Menu d'option
- Menu d'inventaire
- Menu de fin de partie
- HUD pour l'affichage des quêtes

LE MENU PRINCIPAL

- ▷ Jouer
- ▷ Options
- ▷ Quitter

Possède:

3 Animations et 1 musique



LE MENU PAUSE

- ▷ Reprendre
- ▷ Options
- ▷ Quitter

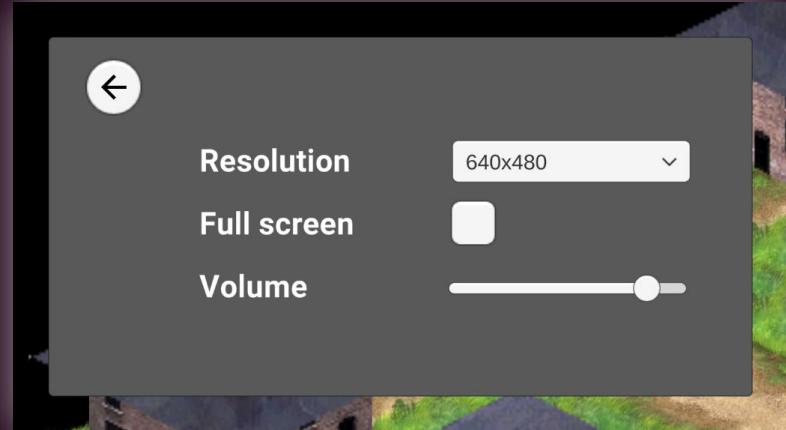
Timescale à 0

On retire les autres éléments de l'affichage



LE MENU D'OPTIONS

- ▷ Changer la résolution
- ▷ Passage en plein écran
- ▷ Modifier le volume



Dans unity Screen.resolutions permet d'obtenir la liste des résolutions de votre écran.

Le volume est lié à un Mixer Audio.

LE MENU D'INVENTAIRE

Deux fenêtres d'inventaire :

- ▷ Inventaire de cartes
- ▷ Inventaire d'objet

Patron de conception → SINGLETON

```
// Implementation of singleton instance
public static Inventory instance;

private void Awake()
{
    // Vérifie si c'est bien un singleton
    if(instance != null){
        Debug.LogWarning("Il y a plus d'une instance d'Inventory dans la scène !");
        return;
    }

    instance = this;
}
```



L'INVENTAIRE DE CARTES

- ▷ Permet de choisir des cartes
- ▷ Permet de voir la description et le nom de la carte
- ▷ 5 cartes par pages
- ▷ Effet d'agrandissement quand on survole une carte

ALGORITHME IMPORTANT DE L'INVENTAIRE

Fonction pour choisir les cartes dans l'inventaires

Variable :

numéro de la page : correspond à la page sur laquelle le joueur est dans l'inventaire. (5 cartes par pages)

id : id du checkbox (valeur passé en paramètre de la fonction)

checkbox : tableau représentant les 5 checkbox permettant de choisir les cartes dans l'inventaire

AlreadyChecked : liste représentant les checkbox qui était déjà coché auparavant

nombre de cartes choisies : nombre de cartes déjà choisies.

ALGO :

Si checkbox (id) est coché :

Si nombre de cartes choisies < 5 :

Si la carte possédant l'id i + numéro de la page * 5 n'est pas déjà choisie :
on l'ajoute au cartes choisie
on ajoute son id dans **AlreadyChecked**
nombre de cartes choisies++

Sinon :

on décoche la checkbox (id)
on affiche le message "popup"

Sinon

Si nombre de cartes choisies > 0 :

Si la carte possédant l'id i + numéro de la page * 5 est déjà choisie :
on la retire des cartes choisie
on retire son id dans **AlreadyChecked**
nombre de cartes choisies--



Fonction pour afficher les cartes dans l'inventaires

Variable :

numéro de la page : correspond à la page sur laquelle le joueur est dans l'inventaire. (5 cartes par pages)

bouton : tableau représentant les 5 boutons disponible sur l'inventaire.

checkbox : tableau représentant les 5 checkbox permettant de choisir les cartes dans l'inventaire

AlreadyChecked : liste représentant les checkbox qui était déjà coché auparavant

ALGO :

Pour i allant de 0 à 5 :

checkbox (i) et bouton (i) sont activé

Si i + numéro de la page * 5 < taille de la liste de carte du joueur :

On affiche l'image de la carte rangé à la position i + numéro de la page * 5 sur le bouton(i)

Si AlreadyChecked(i + numéro de la page * 5) retourne vrai :

on coche la checkbox(i)

Sinon on la décoche

Sinon

On désactive le bouton (i) et la checkbox (i)

L'INVENTAIRE D'OBJET

Pas encore implémenté mais facilement implémentable
Contiendra les objets qui ne sont pas des cartes.



LE MENU DE FIN DE PARTIE

- ▷ Rejouer
- ▷ Accéder au menu principal
- ▷ Quitter le jeu

Précédé de plusieurs animations



AFFICHAGE DE QUÊTES

Le texte est mise à jour à jour à chaque changement de Quêtes

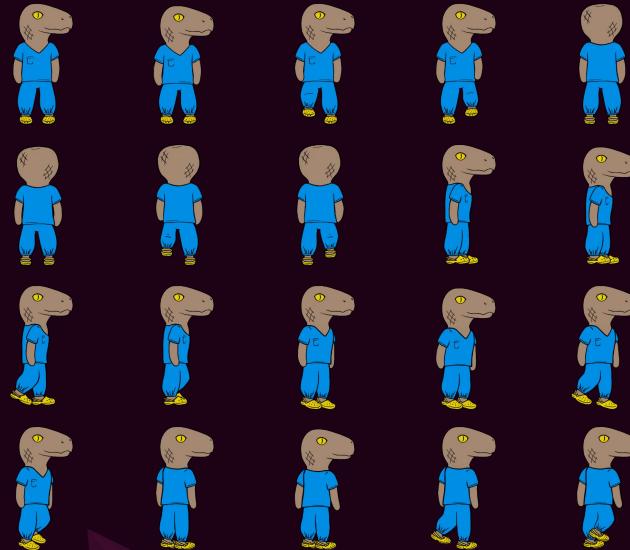


PHASE EXPLORATION

Sprites, environnement, IA et interactions

DÉPLACEMENT DU PERSONNAGE

- Animé sur 8 directions via l'Animator
- Réagit aux entrées clavier

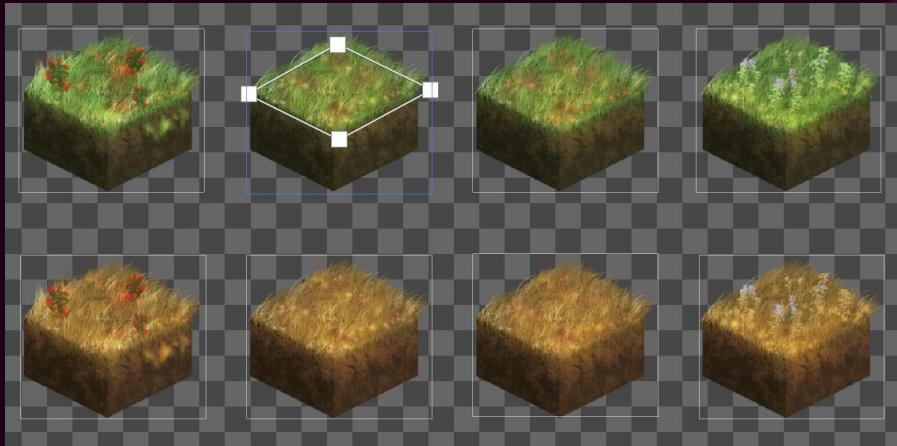


Feuille de sprites du personnage principal

PALETTE DE SPRITES



La palette de sprites utilisée pour la ville



Ajustement manuel des bordures d'un sprite pour le calcul des collisions

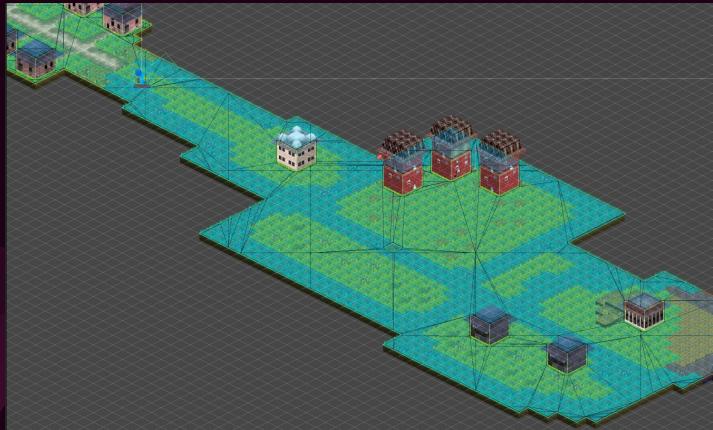
ENVIRONNEMENT DE JEU

Un niveau de jeu est composé de 4 couches :

- Le sol
- Le rez-de-chaussée
- Le premier étage des bâtiments
- Les toits des bâtiments



COMPORTEMENT DES ENNEMIS



Basé sur l'usage d'un Maillage de navigation



Chaque ennemi est représenté par un Agent et se déplace en cherchant le plus court chemin dans le maillage

SYSTÈME D'INTERACTIONS

```
public class Interaction : MonoBehaviour {
    [SerializeField] private List<InteractionEffect> effects;

    private Canvas canvas;
    private Collider2D playerCollider;
    private bool isColliding;

    private void Start() { ... }

    private void Update() { ... }

    private void OnTriggerEnter2D(Collider2D collision) { ... }

    private void OnTriggerExit2D(Collider2D collision) { ... }

    private IEnumerator Interact() { ... }

    [System.Serializable] private class InteractionEffect {
        private enum Type { None, Dialogue, Teleport, Item }

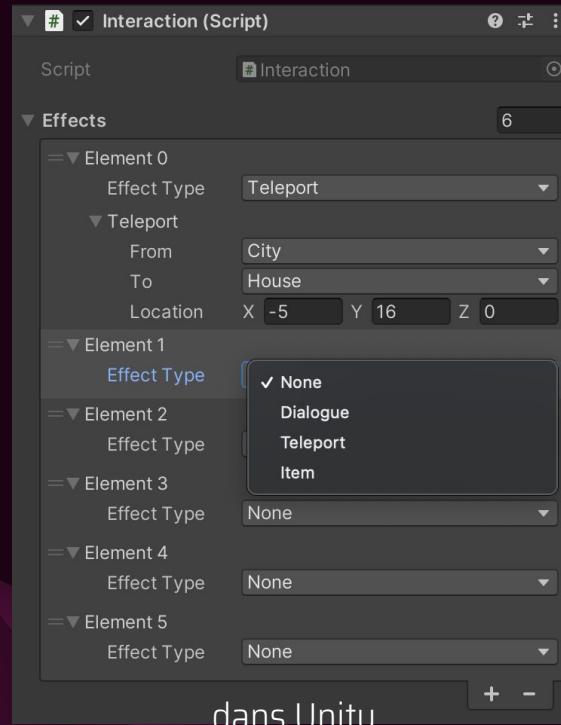
        [SerializeField] private Type effectType = Type.None;
        [ConditionalField(nameof(effectType), false, Type.Dialogue)] public Dialogue dialogue;
        [ConditionalField(nameof(effectType), false, Type.Teleport)] public Teleport teleport;
        [ConditionalField(nameof(effectType), false, Type.Item)] public Item item;

        public IEnumerator Activate() { ... }
    }

    [System.Serializable] private class Teleport { ... }

    [System.Serializable] private class Item { }
}
```

implémentation



dans Unity

SYSTÈME D'INTERACTIONS



Interaction possible

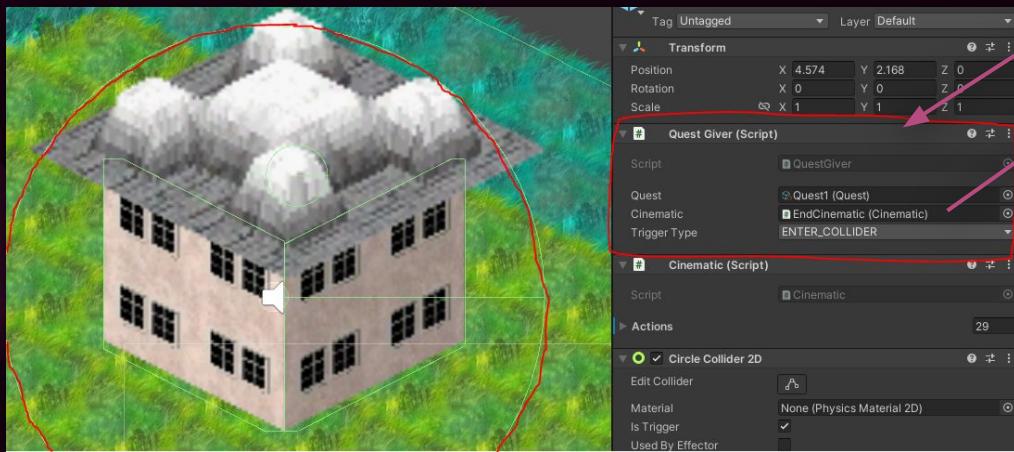
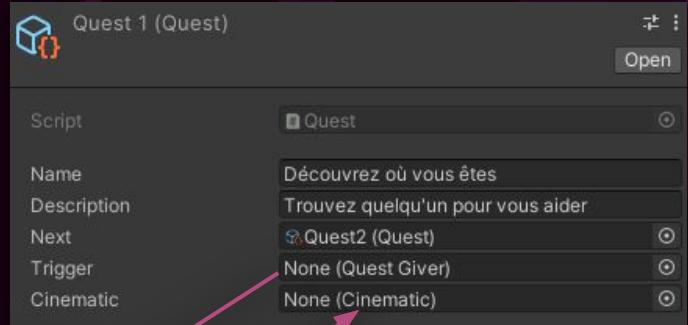
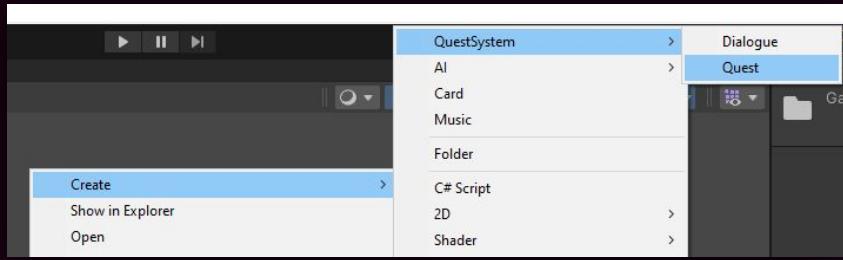


Dialogue déclenché

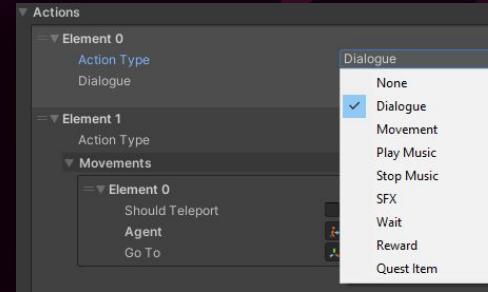
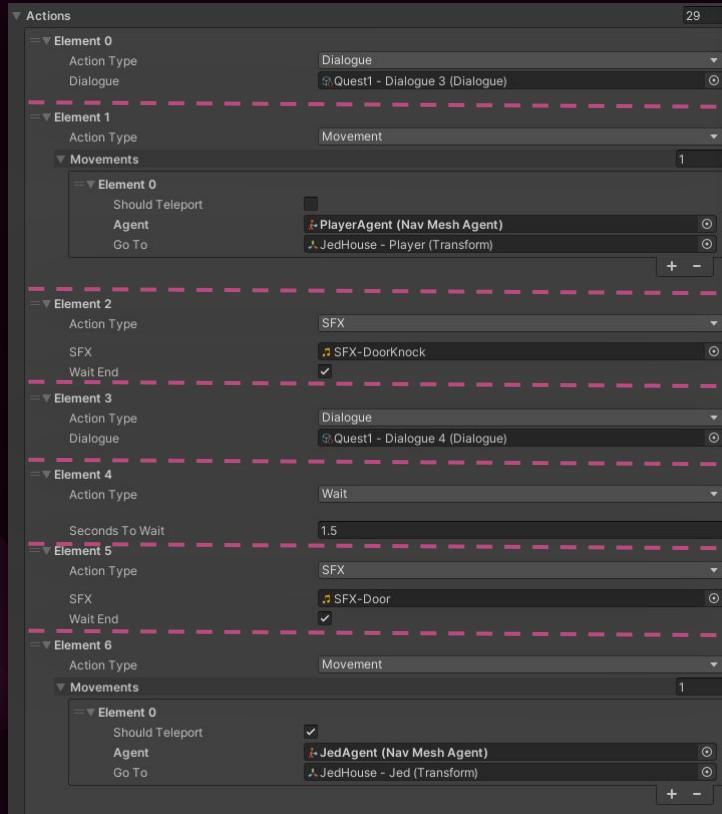
QUÊTES ET CINÉMATIQUES

Sérialisation, Manager Générique, Actions

SÉRIALISATION DES QUÊTES



SÉRIALISATION DES CINÉMATIQUES



→ Utilisation du
ConditionnalField
du Tool *MyBox*

→ Résultat :



ACTIONS D'UNE CINÉMATIQUE

IALOGUE



=▼ Intéressant

Text	Intéressant
Display	Left
Sprite	<input checked="" type="radio"/> Steve

=▼ Vous venez d'acquérir vos premières cartes. Vous pouvez ouvrir votre inventaire e

Text	Vous venez d'acquérir vos premières cartes. Vous
Display	None
Sprite	None (Sprite)

=▼ Je retourne chez moi chercher et organiser mes notes, je vous y attendrai pour le c

Text	Je retourne chez moi chercher et organiser mes nc
Display	Right
Sprite	<input checked="" type="radio"/> Jed

ITEM DE QUÊTE



ACTIONS D'UNE CINÉMATIQUE

MOUVEMENTS



Element 15

Action Type: Movement

Movements

- Element 0
 - Should Teleport
 - Agent
 - Go To
- Element 1
 - Should Teleport
 - Agent
 - Go To

PlayerAgent (Nav Mesh Agent)

City Point - 1 - Player (Transform)

JedAgent (Nav Mesh Agent)

City Point - 1 - Jed (Transform)



Element 4

Action Type: Wait

Seconds To Wait: 1.5

ACTIONS D'UNE CINÉMATIQUE ... NON-VISUELLES

PLAY MUSIC

Element 13

Action Type	Play Music
Music	Music1 (Music)

STOP MUSIC

Element 27

Action Type	Stop Music
Fade Out	<input checked="" type="checkbox"/>

SFX

Element 2

Action Type	SFX
SFX	SFX-DoorKnock
Wait End	<input checked="" type="checkbox"/>

REWARD

Element 24

Action Type	Reward
Reward	
Cards	
Element 0	Rooted (Card Asset)
Element 1	SinisterPath (Card Asset)
Element 2	Loneliness (Card Asset)
Element 3	Resting (Card Asset)
Element 4	Loneliness (Card Asset)
Element 5	SinisterPath (Card Asset)
Element 6	Resting (Card Asset)

PHASE DE COMBAT

Sérialisation, Combat différé, Bézier, Feedback...

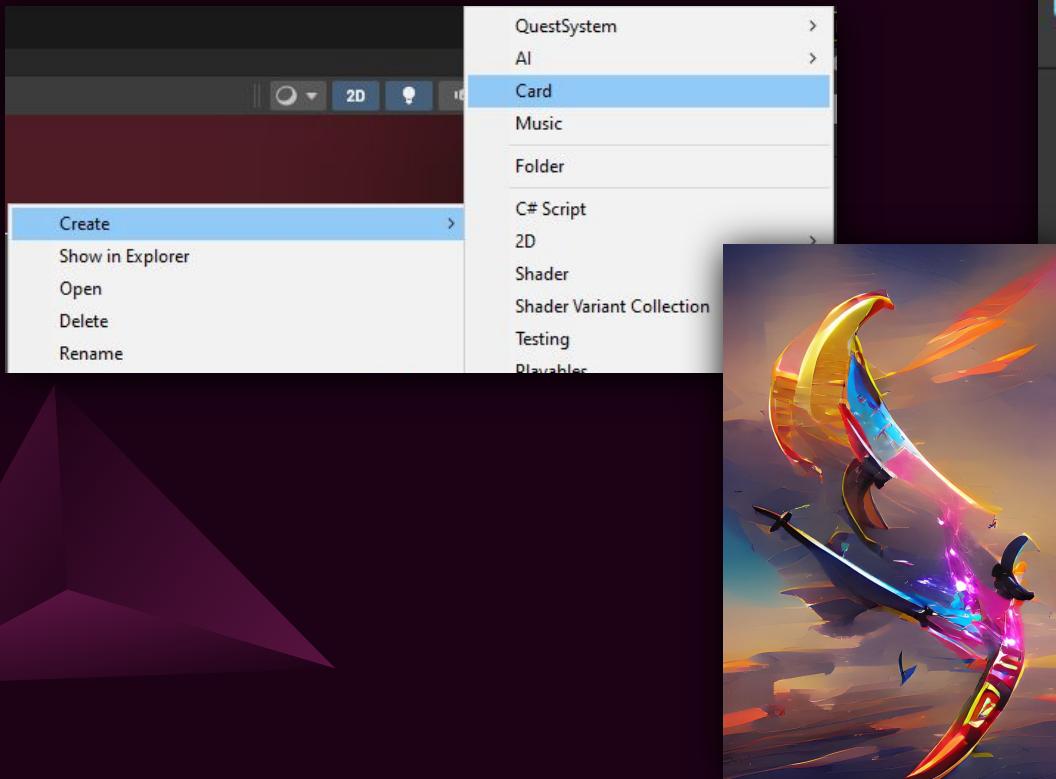


DÉCLENCHEMENT DU COMBAT



- ▷ Collision
- ▷ Application d'une texture de transition
- ▷ Apparition des éléments graphique par des Coroutines

SÉRIALISATION DES CARTES



SÉRIALISATION DES CARTES

```
// asset filename
switch (name)
{
    // Debuff
    case "FlyingCarpet":
        fighter.status.Clear();
        break;

    // Cure 20% of your health
    case "Resting":
        fighter.Heal((int)(fighter.MaxHP * 0.2f));
        break;

    // Deal 5% damage, then 10%, and 15%
    case "Loneliness":
        opponent.Hurt((int)(opponent.MaxHP * 0.05f));
        AddStatus(opponent, new Status(this, opponent, Status.Type.Hurt, (int)(opponent.MaxHP * 0.10f)), Turn + 1);
        AddStatus(opponent, new Status(this, opponent, Status.Type.Hurt, (int)(opponent.MaxHP * 0.15f)), Turn + 2);
        break;
}
```

- ▷ 4 actions possibles : Soigner, Blessier, Passer un tour, Enlever status
- ▷ 2 entités : Combattant, Opposant
- ▷ Possibilité d'appliquer des *Status* : pour appliquer un effet dans *n* tours

SYSTÈME DE COMBAT

INTERACTIONS DU JOUEUR

ONHOVER / ONEXIT



ONDRAg / ONDROP



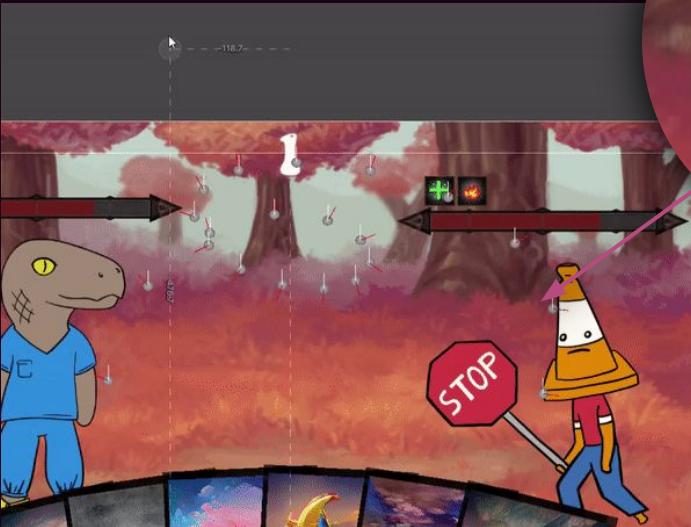
SYSTÈME DE COMBAT

COMBAT DIFFÉRÉ

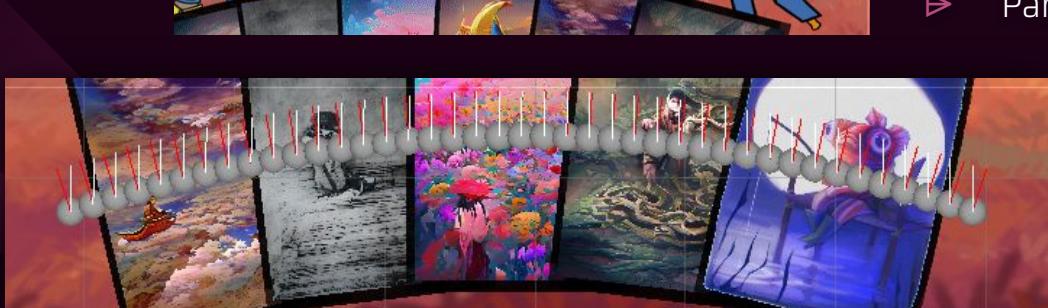


- ▷ 4 actions possibles : Soigner, Blessier, Passer un tour, Enlever status
- ▷ 2 entités : Combattant, Opposant
- ▷ Possibilité d'appliquer des *Status* : pour appliquer un effet dans n tours

COURBES DE BÉZIER



- ▷ Algorithme de Casteljau
- ▷ Réutilisables dans le jeu
- ▷ Position et orientation des points avec la normale
- ▷ Parcours du début à la fin (et inversement)



FEEDBACK

Dégâts



Soin



Pétrification



CONCLUSION

Critiques, enseignements et perspectives

AUTO-CRITIQUE

- Ampleur réelle du travail sous-estimée, fonctionnalités absentes
- Grande disparité de compétences
- Mauvaise gestion des imprévus liés aux personnes externes

ENSEIGNEMENTS ACQUIS

- Ne pas sous-estimer la phase d'apprentissage d'une nouvelle technologie
- Nécessité de mieux anticiper les différentes phases de production en définissant une organisation claire et flexible
- Prise de conscience sur la complexité de développer un jeu vidéo

PERSPECTIVES FUTURES

- Ajouter d'autres personnages et environnements
- Intégrer le reste du scénario (déjà écrit) en différents actes
- Publier la version complète sur itch.io

REMERCIEMENTS

Noura FARAJ | encadrement du projet

Pénélope WINTRINGER | artiste 2D

Ninon | artiste 2D

Iannis HENNECART | sound designer

Gabriel SALACH | scénariste

...et à tous ceux qui nous ont fait des retours à Into the Game

MERCI POUR VOTRE ATTENTION !

Des questions?

