



Level Design

MYST

LE RETOUR DE L'ILE



Sommaire

♦ Vision du jeu ♦	3
❖ Description ❖	3
❖ Map & Blockout ❖	4
♦ Structure du level ♦	5
♦ Layout & Navigation ♦	6
♦ Design des puzzles ♦	7
❖ Type de puzzles ❖	7
❖ Structure d'un puzzles ❖	8
❖ Difficulté ❖	8
♦ Guidage du joueur ♦	8
❖ Méthodes utilisées ❖	8
❖ Points UX critiques ❖	8
♦ Direction artistique & lisibilité ♦	9
♦ Playtests & itération ♦	9



◆ Vision du jeu ◆

❖ Description ❖

Type de jeu :

Escape game 3D en semi open world (Unity + Django)

Concept :

Le joueur explore une île mystérieuse et doit résoudre plusieurs énigmes pour progresser et atteindre un objectif final.

Objectif joueur :

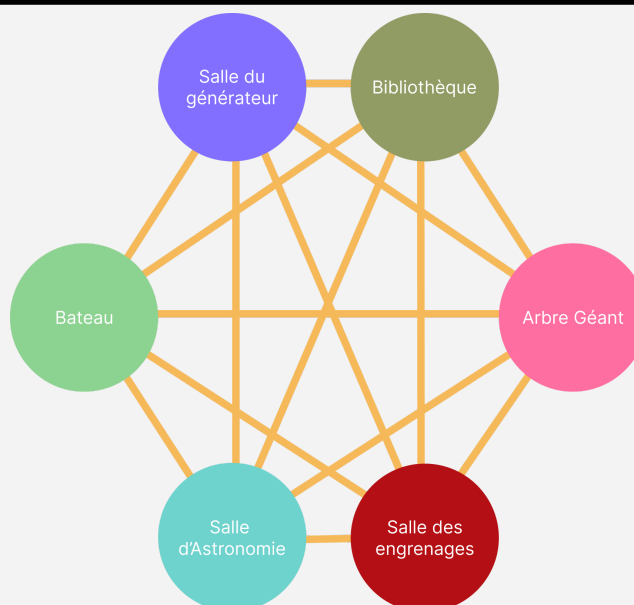
- Récupérer des objets clés (5 morceaux de papier + 1 rune)
- Résoudre les puzzles
- Déchiffrer un code final en D'ni
- Terminer le jeu en moins de 30 minutes

Intentions de design :

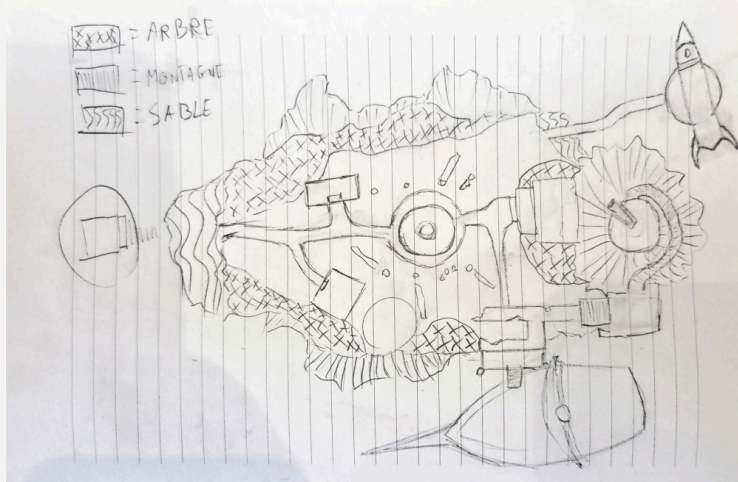
- Exploration libre mais guidée
- Immersion forte (univers cohérent)
- Compréhension progressive des mécaniques

❖ Map & Blockout ❖

Les zones de notre jeu, étant un semi-open world et n'ayant aucun ordre pour les puzzle (sauf pour la dernière), on peut explorer chaque zone de la manière que l'on veut et dans l'ordre qu'on veut sans que ça gêne ou nous bloque dans le jeu.



La map fait sur papier, donne une idée plus précise de comment le niveau ressemble et se connecte. On est sur une map qu'on peut explorer de la manière que l'on veut, il y aura plusieurs pièces à explorer pour faire les puzzles.



La version finale et peaufiner de la map présentés précédemment.

La zone limite est la mer qui fait office de barrière au joueur et refait spawn au milieu de la map



◆ Structure du level ◆

Type de level : Semi open world avec hubs

Structure globale :

- 1 île = map principale
- Semi open world
- Zones interconnectées avec puzzles

Organisation :

- Zone centrale (hub implicite)
- Plusieurs zones secondaires (puzzles)
- Accès progressif (gates)



Zone central du jeu, ou on respawn si on tombe dans l'eau

Flow :

- Exploration → découverte → puzzle → récompense → progression

◆ Layout & Navigation ◆

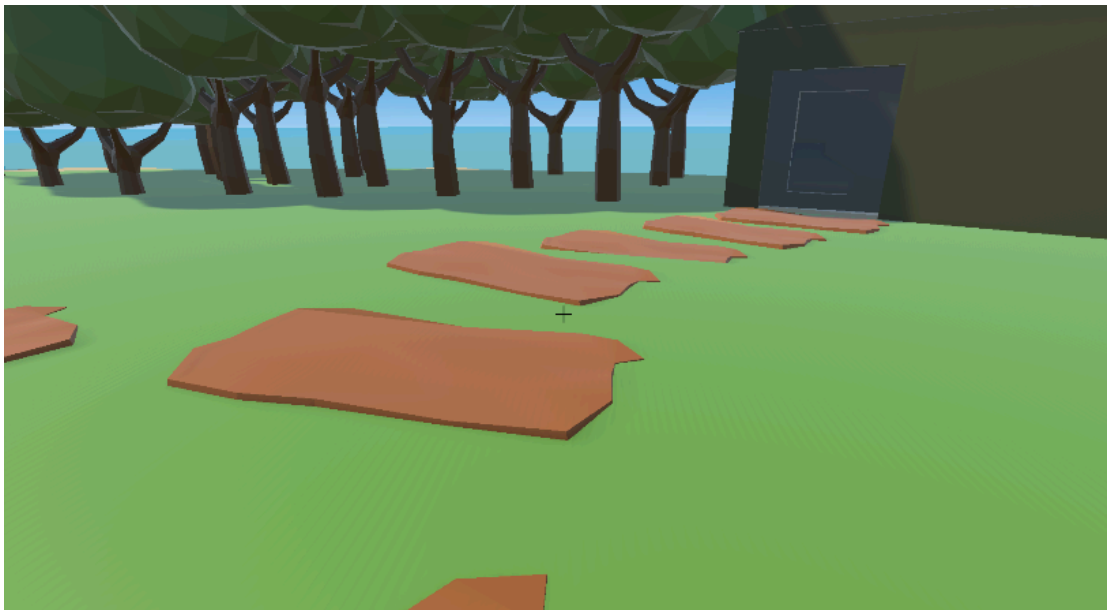
Objectif : Le joueur comprend où aller sans UI intrusive

Principes clés :

- Le joueur doit pouvoir se déplacer librement
- Pas de labyrinthe frustrant
- Retour facile entre zones (éviter frustration)

Guidage utilisé :

- Landmarks (éléments visibles de loin)
- Chemins naturels (formes du terrain)
- Points d'intérêt visibles (teasing)



Chemin naturel qui incite le joueur à y aller



Point d'intérêt visible au loin

◆ Design des puzzles ◆

Objectif : Permettre au joueur de progresser via des énigmes logiques et immersives

❖ Type de puzzles ❖

1. Puzzle d'exploration

- Trouver objets (disquettes cachées)

2. Puzzle interactif

- Interaction avec objets/environnement

3. Puzzle D'ni

- Comprendre symboles
- Traduire en chiffres

4. Puzzle final

- Réorganiser code D'ni

❖ Structure d'un puzzles ❖

1. Découverte (visuel / environnement)
2. Compréhension
3. Résolution
4. Feedback immédiat (ouverture, son, animation)

❖ Difficulté ❖

1. Accessible à débutant
2. Progression logique
3. Pas de blocage total

◆ Guidage du joueur ◆

Objectif : Le joueur doit comprendre quoi faire sans aide externe

❖ Méthodes utilisées ❖

1. Objets visibles = interactifs
2. Contraste visuel (couleur / lumière)
3. Mise en scène (cadres naturels)
4. Teasing (objet visible mais inaccessible)

❖ Points UX critiques ❖

1. Compréhension des puzzles
2. Visibilité des éléments interactifs
3. Clarté des symboles D'ni
4. Temps de compréhension

◆ Direction artistique & lisibilité ◆

Objectifs :

- Immersion
- Lisibilité

Règles :

- Éléments importants = visibles
- Éléments interactifs = distincts
- Éviter surcharge visuelle

UX critique :

Le joueur doit immédiatement voir :

- où interagir
- où aller

◆ Playtests & itération ◆

Règles :

- Tester tôt et souvent
- Observer les joueurs (sans aider)
- Corriger :
 - incompréhension
 - mauvaise lisibilité
 - frustration

Important :

Le feedback négatif est le plus précieux