GAMEPLAY DE NAVIGATION

**Mécanique de déplacement**

1. **But**

Augmenter sa vitesse

1. **Mécanique**

Le joueur peut se déplacer et dévaler des pentes pour augmenter sa vitesse

1. **Inputs**

**Joystick gauche** (deux axes) : faire pivoter l’avatar su lui-même

**Gachette droite** : accélérer ou faire avancer le personnage.

1. **Challenges**

**Anticipation** : la latence des mouvements de l’avatar, lorsqu’il tourne, entraîne une projection de position “objectif” que l’avatar doit faire pour ajuster ses contrôles ;

**Reflexe** : recalculer sa trajectoire en fonction de l’apparition d’une nouvelle pattern ;

**Précision** : certaines pentes ont un accès optimal qui requiert une précision dans la trajectoire ;

**Mesure** : avec la vitesse, le contrôle de l’avatar se dégrade, recquérant plus de précision dans l’input ;

**Tactique** : plusieurs choix s’offrent au joueur, en fonction des patterns qui apparaissent

**Deduction** : en fonction de la pente, le joueur peut estime si sa descente va lui offrir une augmentation de vitesse suffisante ou non pour repousser la limite de vitesse max ;

1. **Paramètres de gameplay**

**Paliers de vitesse** : le joueur débloque un palier de vitesse lorsqu’il dépasse au moins une fois sa valeur.en atteignant une certaine vitesse, le joueur rentre dans une période “d’extase” qui le maintient à cette vitesse pendant un temps limité.

*Exemple de corrélation entre les paliers et la vitesse nécessaire pour les débloquer*

|  |  |
| --- | --- |
| Palier | Vitesse nécessaire pour le débloquer |
| 0 | Débloqué par défaut (10) |
| 1 | 20 |
| 2 | 30 |
| 3 | 50 |
| 4 | 80 |
| 5 | 150 |
| 6 | 300 |

**Dégradation** : la perte de vitesse, inaltérable (exeptée en phase “d’extase”) s’effectue de moins en moins rapidement à mesure que la vitesse s’écarte du dernier palier atteint. La perte de vitesse cesse de se dégrader une fois le palier 0 atteint.

*Exemple de corrélation entre les paliers et la perte de vitesse*

|  |  |
| --- | --- |
| Palier (n = palier max. atteint) | Perte de vitesse  (1 = valeur de dégradation max.) |
| n | 1 |
| n - 1 | 0.75 |
| n - 2 | 0.5 |
| n - 3 | 0.25 |
| ... | .. |
| 0 | 0 |

1. **RLD**

En fonction de sa vitesse, des patterns de LD apparaissent sur le chemin du joueur, actualisant ainsi ses objectifs.

**Mécanique d’orientation de la caméra**

1. **But**

Appréhender son environnement.

1. **Mécanique**

Le joueur peut orienter sa caméra pour observer son environnement indépendamment de la trajectoire de son avatar.

1. **Inputs**

**Joystick droit** (deux axes) : faire pivoter la caméra

1. **Challenges**

Mesure :

**Orientation** : le joueur doit avoir une bonne représentation de son avatar dans l’espace (proprioception) pour savoir où diriger sa caméra pour atteindre l’angle souhaité.

1. **Paramètres de gameplay**

(voir 3C :

<https://docs.google.com/document/d/1zVv0fNurR9kuGz21eQ_G6oYcKuUDleidtnaBuwQq2vk/edit#> )

1. **RLD**

L’apparition de patterns ne s’effectue pas exclusivement devant l’avatar. Il doit donc surveiller son environnement, autour de lui, mais également au dessus.