

PROJET DOOM-LIKE P2P

COMpte-rendu d'avancement : itération 2

Équipe : Chailan Cyprian, Gros Geoffrey, Lounici Ilyes,

Ray Marcelin

9 janvier 2026

Sommaire

- **Rappel du projet et objectifs de l'itération**
- **Fonctionnalités actuelles**
 - **Mouvement des monstres**
 - **BSP* - Création de l'arbre sémantique**
 - **BSP - Affichage d'un mur isolé**

* (binary space partitioning ou BSP) est un système utilisé pour diviser l'espace en zones convexes. (wikipedia.fr)

Objectif de l'itération 2

But principal : Démarrage du moteur de rendu BSP

Travail Impératif :

- **Découpage du moteur BSP en plusieurs étapes**
- **Réaliser des étapes du moteur BSP**
- **Réaliser les mouvements des monstres intelligent**

Objectifs Optionnels :

- **Lissage des mouvements pour paraître plus naturel**

Fonctionnalités Actuelles (pathfinding)

- Selon le livre *Artificial Intelligence for Games*, le pathfinding (ou recherche de chemin) est défini comme la capacité d'une IA à **calculer un itinéraire approprié à travers un niveau de jeu** pour un personnage
- Concrètement, cela permet au monstre de calculer une route complète à l'avance pour contourner les murs et nous traquer efficacement, rendant ainsi le jeu plus difficile et réaliste.

Fonctionnement RRT*

Le RRT* construit un arbre qui explore l'espace.

1. Exploration

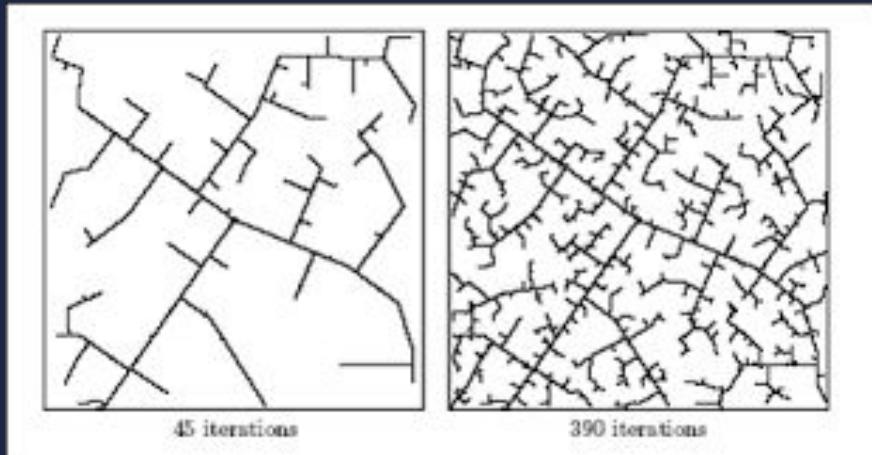
- Point aléatoire
- Noeud le plus proche
- Nouveau noeud

2. Choix du meilleur parent

- Test des voisins
- Parent avec coût minimal

3. Rewiring (recâblage)

- Réorganisation de l'arbre
- Chemins plus courts



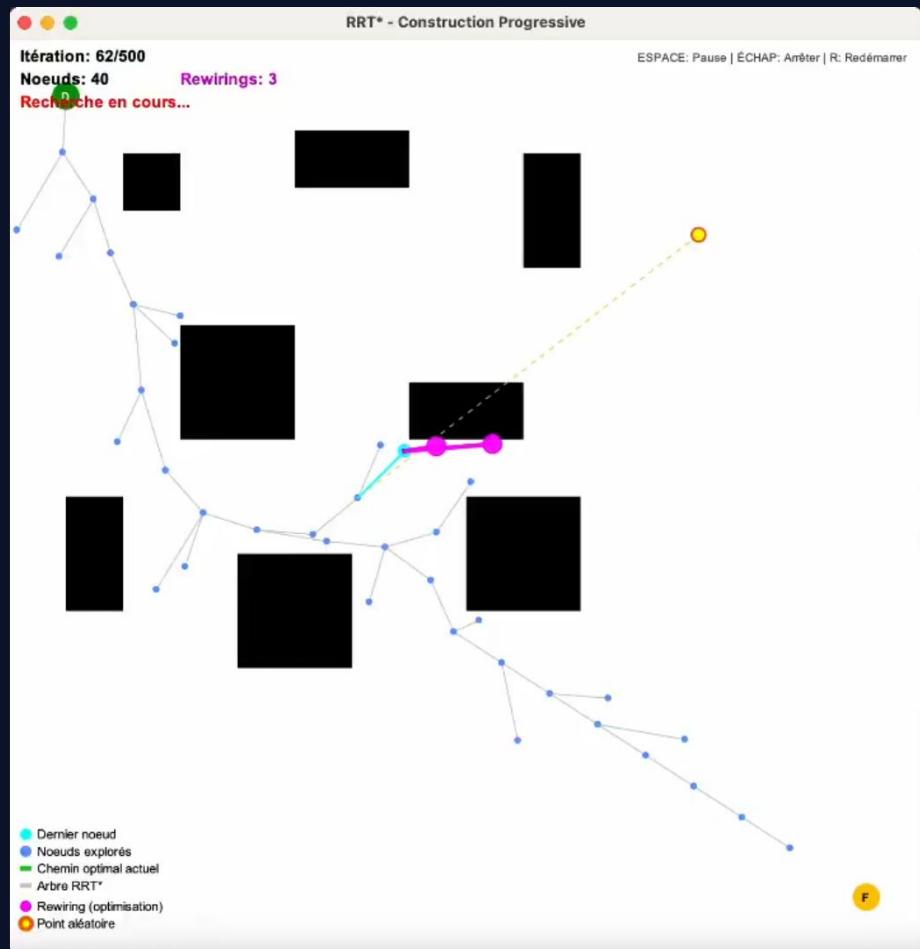
https://en.wikipedia.org/wiki/Rapidly_exploring_random_tree

<https://www.youtube.com/watch?v=OXikozpLFG0>



Fonctionnement RRT*

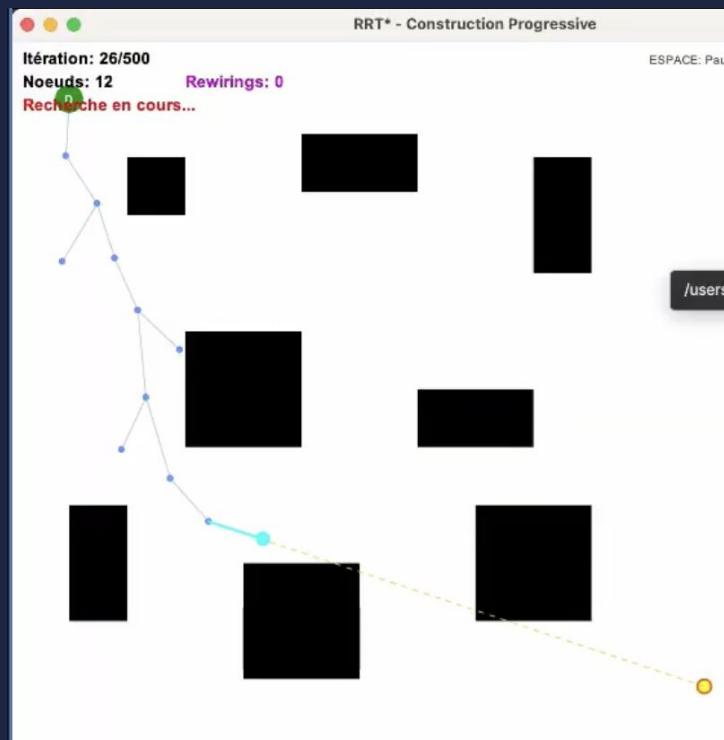
Fonctionnement RRT*



Fonctionnement RRT*(Liang-Barsky)

- Le segment relie deux nœuds RRT*
- Les obstacles sont des rectangles
- Liang-Barsky teste l'intersection segment–rectangle

Si le segment et le carré ont une intersection alors on réessaie de nouvelles coordonnées jusqu'à avoir une réponse correct.



https://en.wikipedia.org/wiki/Liang%20-%20Barsky_algorithm

Fonctionnalités Actuelles (steering behaviors)

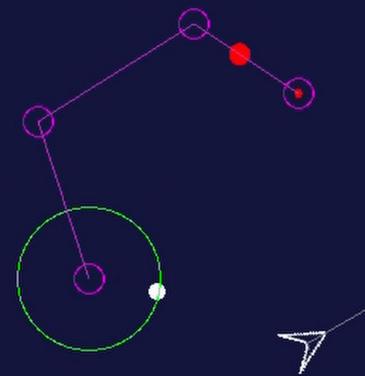
Définition (Source : Artificial Intelligence for Games)

- Les **Steering Behaviors** (ou comportements de pilotage) sont une famille d'algorithmes de mouvement qui ne modifient pas directement la position d'un personnage, mais appliquent des **forces** ou des **accélérations**
- Contrairement au mouvement "cinématique" (qui change juste la position), ils permettent des mouvements **dynamiques** et fluides, prenant en compte l'inertie et la vitesse actuelle du personnage.

Fonctionnalités Actuelles (steering behaviors)

Mais, à quoi ça sert dans le jeu ?

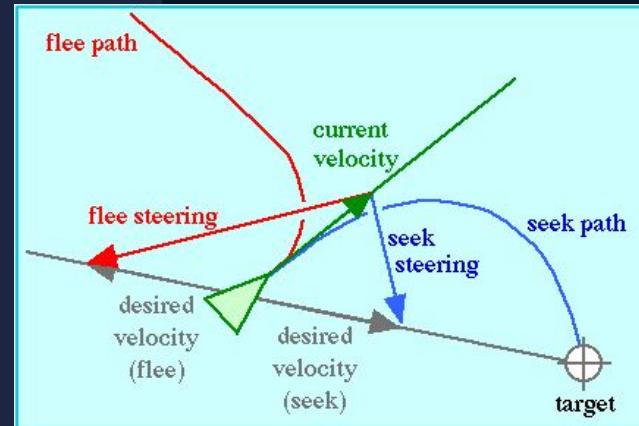
- Ils permettent d'éviter les mouvements robotiques (arrêts brusques, virages instantanés à 90°) en lissant les trajectoires. Les monstres ont une sorte de "masse" virtuelle qui rend leurs déplacements plus naturels
- Aussi, ils permettent aux monstres de suivre le chemin calculé (par l'algorithme RRT*) de manière souple, en s'ajustant dynamiquement si la cible bouge légèrement
- Enfin, ils servent de briques de base pour créer des comportements variés : poursuivre le joueur, s'arrêter doucement à une position, ou éviter des collisions



Fonctionnalités Actuelles (steering behaviors)

Fonctionnement Technique

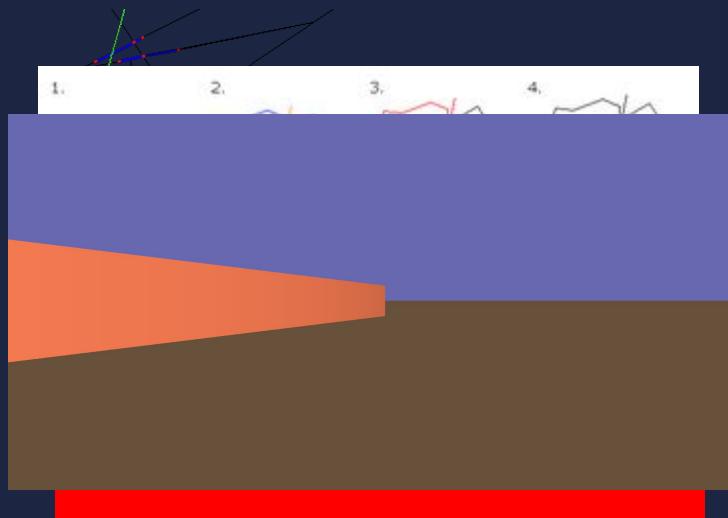
- L'algorithme fonctionne en 3 étapes (boucle de décision):
 - Intention (Desired Velocity) : Le monstre calcule sa vitesse idéale pour atteindre sa cible (ex: foncer tout droit)
 - Correction (Steering Force) : L'algorithme compare cette intention avec son mouvement actuel
(Force = Vitesse Désirée - Vitesse Actuelle)
 - Intégration : Cette force est appliquée au “moteur physique” du jeu pour mettre à jour la position du monstre à l'image suivante



Steering Behaviours For Autonomous Characters by Craig W. Reynolds
<https://www.red3d.com/cwr/steer/gdc99/>

BSP - découpage des étapes

- 1 : Découpage d'une carte (liste de Mur) en arbre**
- 2 : Affichage d'un mur isolé**
- 3 : Ajout de texture sur le mur isolé**
- 4 : Parcour de l'arbre et dessin des murs dans l'ordre**
- 5 : Ajout du dessin des Sprite (ennemi, joueur, objet, ...)**



https://en.wikipedia.org/wiki/Binary_space_partitioning
https://www.dgp.toronto.edu/public_user/JamesStewart/378notes/23bsp/

BSP - Découpage de l'arbre

Architecture - diagramme de classes

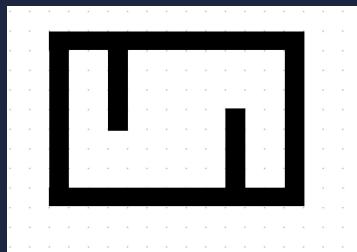
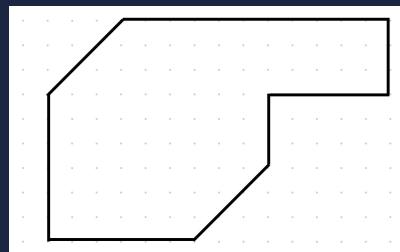
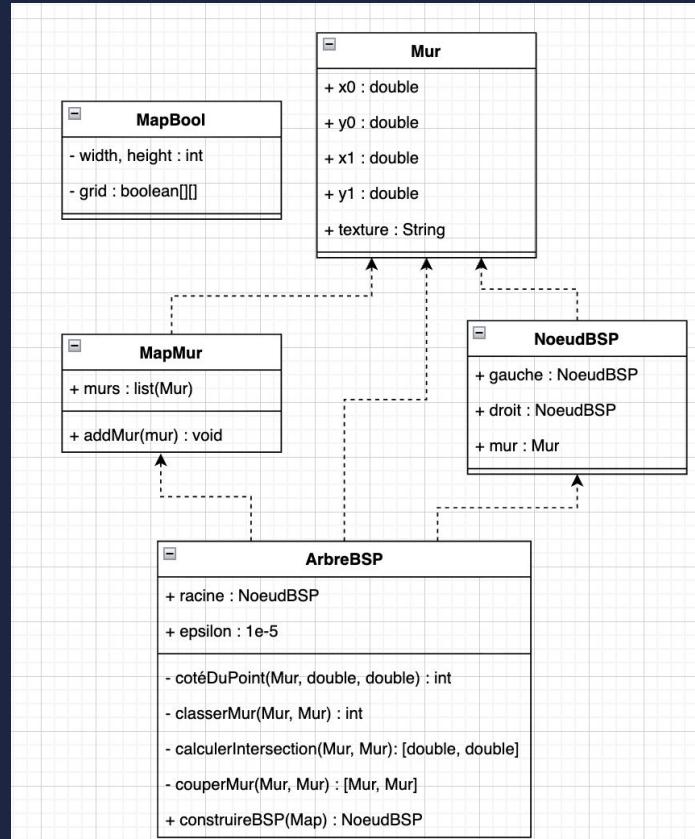


Tableau de
booléen



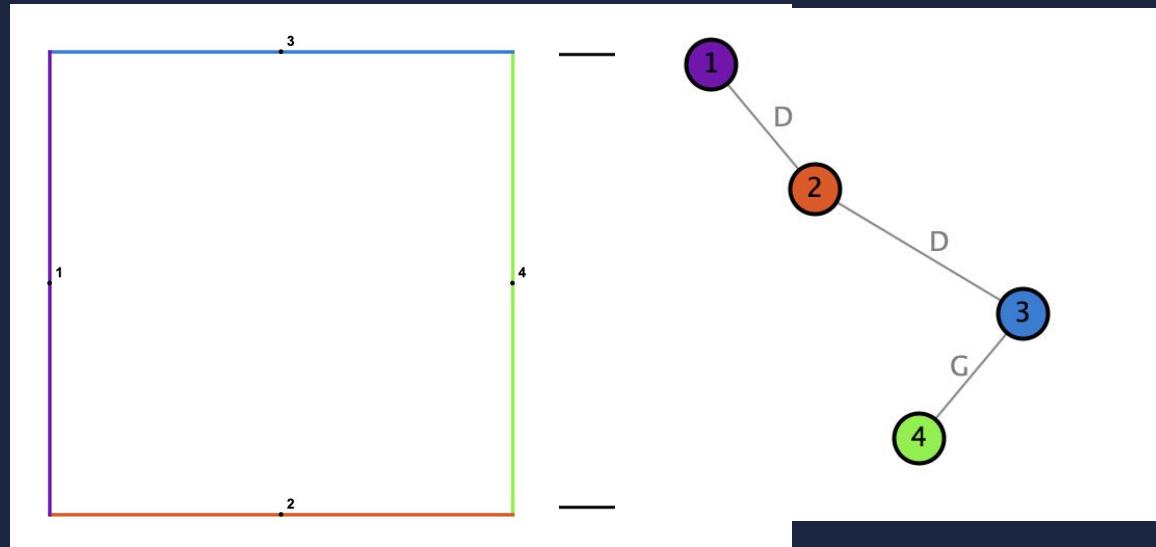
Liste de
Murs



BSP - Découpage de l'arbre

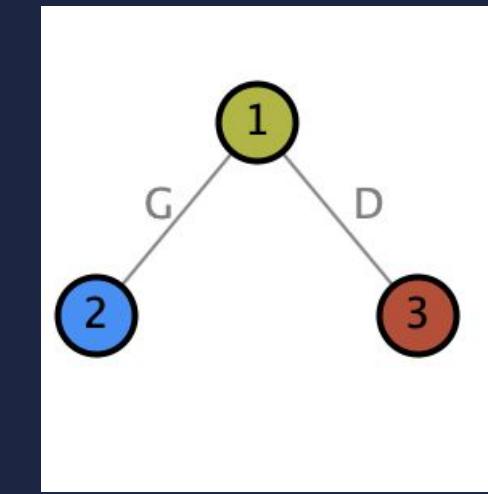
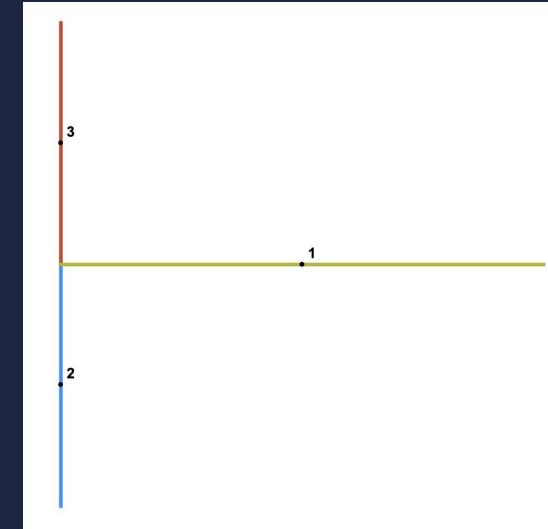
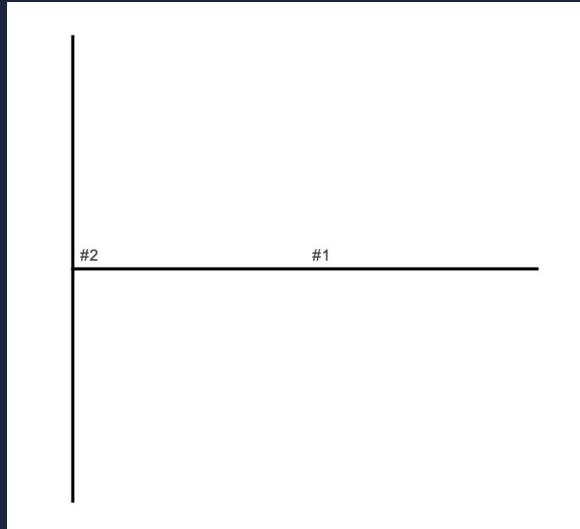
Fonctionnement :

- 1) On prend le premier mur comme racine
- 2) On parcourt tous les autres murs, et on les stocke dans une liste en fonction de s'ils sont à gauche ou à droite
- 3) On rappelle récursivement cette méthode avec les deux listes de murs
 - > le *nœudGauche* devient le sous-arbre des murs de gauche
 - > le *nœudDroite* devient le sous-arbre des murs de droite



BSP - Découpage de l'arbre

Cas spécial : Mur inclassifiable - ni à gauche, ni à droite

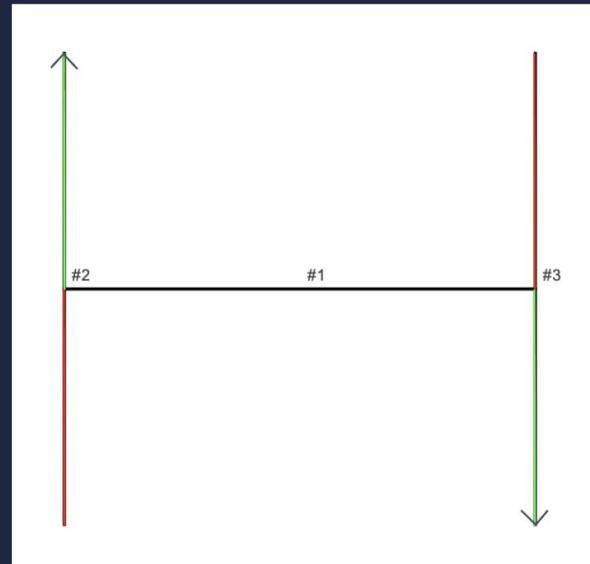


BSP - Découpage de l'arbre

Bug bloquant - découpage d'un mur

- La méthode qui découpe le mur retourne le même résultat pour ces 2 murs : [Mur rouge, Mur vert]. Et l'algorithme ajoutais arbitrairement le premier mur retourné dans la partie gauche, et le deuxième dans la partie droite.

- Il fallait vérifier le sens du mur ! (regarder si les points $x0$ et $y0$ du mur sont à gauche ou à droite du mur), et ajouter les sous-portion de mur dans la bonne liste.

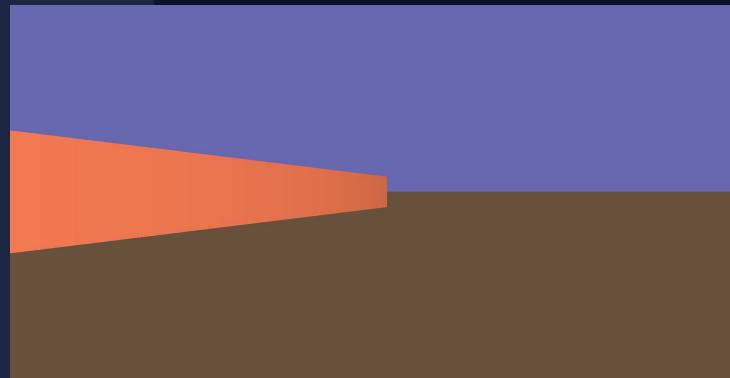


BSP - Rendu d'un mur

Fonctionnement :

Projection & Rastérisation

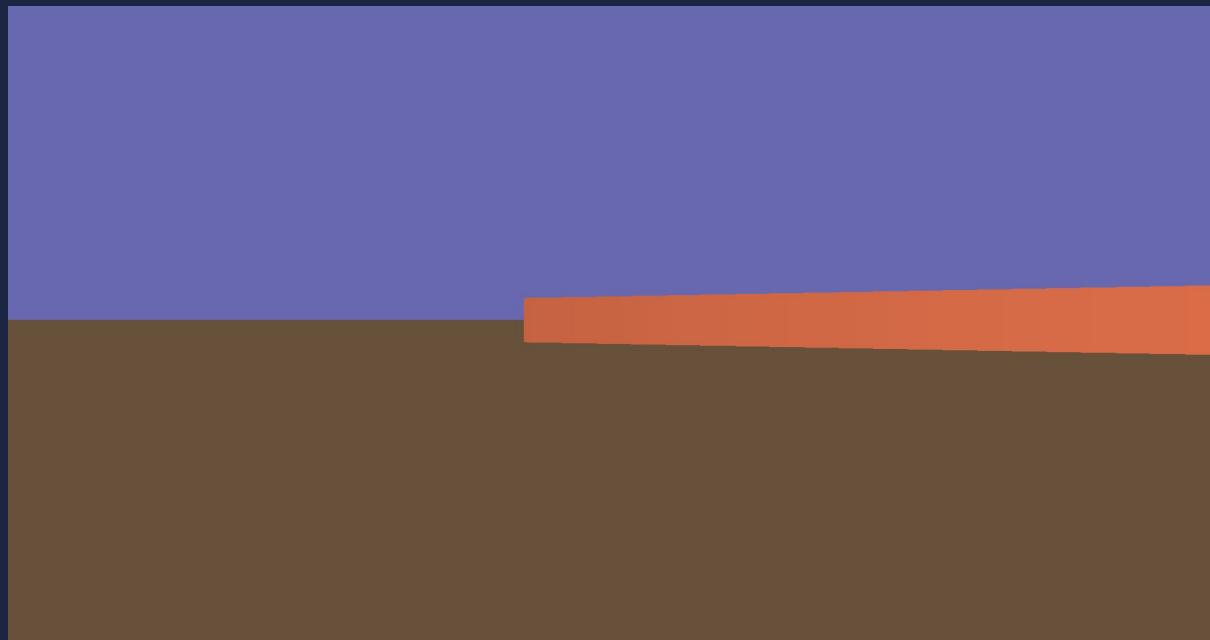
- **Projection** : Utilisation de la loi de Thalès (x / z) plus l'objet est loin, plus il se rapproche du centre de l'écran
- **Balayage** : Parcours de la zone écran du mur, colonne par colonne (horizontalement)
- **Dessin** : Pour chaque colonne, calcul de la hauteur (inversement proportionnelle à la profondeur) et traçage de la ligne verticale



Clipping : technique utilisée pour diminuer la charge de calcul lors du rendu de scènes en 3D

Rastérisation : processus qui transforme un objet graphique (comme un mur en 3D) en une grille de points colorés (pixels) affichés sur l'écran

BSP - Rendu d'un mur



Intégration et Résultats

Etat actuel

- Pathfinding et lissage des mouvements fonctionnels pour les monstres dans un environnement de 2D actuellement.
- Planification des étapes du BSP réalisé, ainsi que certaines étapes implémentées.

Perspectives

Bilan de l'Iteration 2

Les objectifs impératifs sont atteints

Prochaines étapes (Iteration 3) :

- Fin du BSP:
 - Ajout de texture
 - Parcours de l'arbre afin d'afficher tous les murs
 - Ajout d'affichage de Sprite dans le rendu final
- Arbre de décision pour les monstres

Questions ?



Merci pour votre attention