

# Présentation du projet Portal 0.0

BADSTÜBER Elian BIDAULT Matthieu FOCHEUX Vital  
Licence 3 Informatique

Mars 2024



Tuteur : Julien BERNARD

# Table des matières

## 1 Introduction

## 2 Algorithmes et techniques

- Construction de mur
- Les collisions
- Différentes stratégies pour un rendu 3D

## 3 Implémentation

- Les rendus
- Les portails

## 4 Conclusion

# Introduction

## Système de jeu

- Portal 0.0 → principes techniques de plusieurs jeux vidéos connus
- Résolution d'énigmes à l'aide de portails
- Téléportation lorsqu'on passe à travers
- Principe de Portal (2007)



Figure: Portal (2007)

# Introduction

## Technique graphique

- Portal 0.0 → principes techniques de plusieurs jeux vidéos connus
- Méthode raycasting
- Rendu 2.5D popularisé dans les années 90
- Principe de Wolfenstein3D (1992)



Figure: Doom (1993)

# Introduction

## Technologies utilisées



# Plan

## 1 Introduction

## 2 Algorithmes et techniques

- Construction de mur
- Les collisions
- Différentes stratégies pour un rendu 3D

## 3 Implémentation

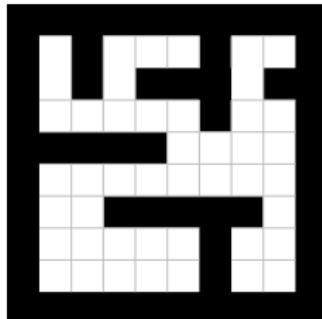
- Les rendus
- Les portails

## 4 Conclusion

# Construction de mur

## Commencement

- Création de la carte à partir d'un PNG
- Récupération des coordonnées des cellules à l'aide d'un parcours en profondeur

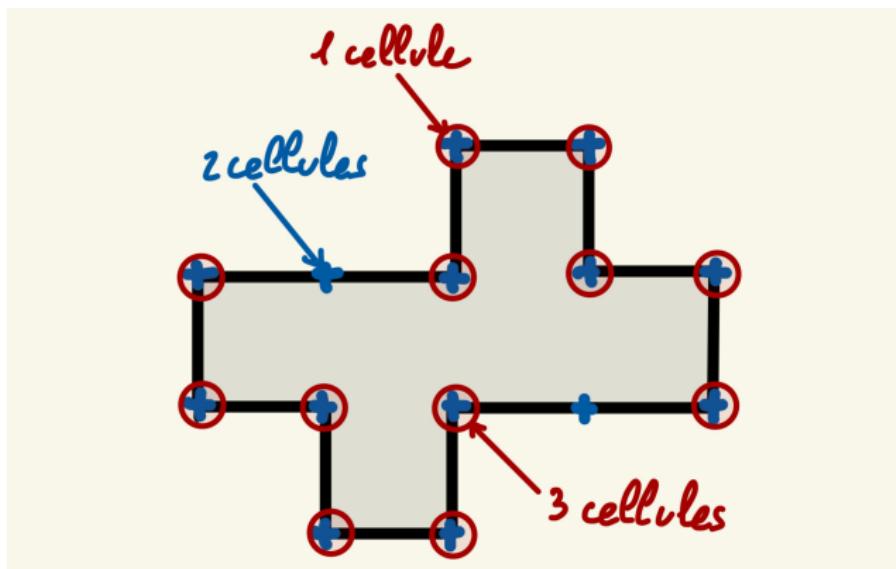


- Les coordonnées des cellules correspondent au coin supérieur gauche de chaque cellule
- Une cellule correspond à un bloc unitaire de la carte

# Construction de mur

## Sommets utiles

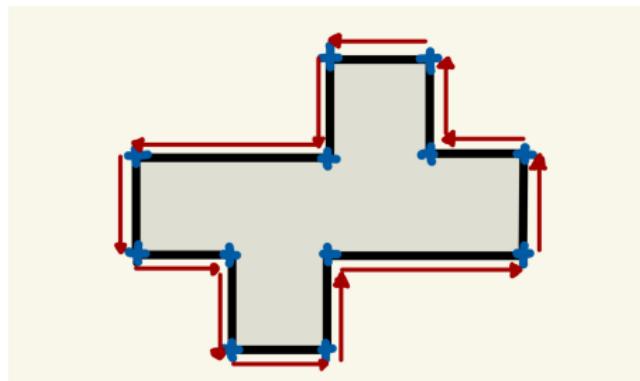
- Boucle sur les coordonnées des cellules
- Parité du nombre de cellules adjacentes



# Construction de mur

## Trie des sommets

```
d = {'d', 'b', 'g', 'h'}  
dir = d[i%4], dir2 = d[(i+1)%4]  
while(v.contient(sommetA(dir)) ou v.contient(sommetA(dir2))):  
    if canGo(dir) and !canGo(dir2):  
        v <- sommetA(dir)  
    else:  
        v <- sommetA(dir2)  
        echange(dir, dir2)
```



# Plan

## 1 Introduction

## 2 Algorithmes et techniques

- Construction de mur
- **Les collisions**
- Différentes stratégies pour un rendu 3D

## 3 Implémentation

- Les rendus
- Les portails

## 4 Conclusion

# Les collisions

## Système de collision cercle/rectangle

- Recherche du point le plus proche
  - ▶ méthode de clampage
- Validation de proximité
  - ▶ remplacement éventuel du personnage

# Plan

## 1 Introduction

## 2 Algorithmes et techniques

- Construction de mur
- Les collisions
- Différentes stratégies pour un rendu 3D

## 3 Implémentation

- Les rendus
- Les portails

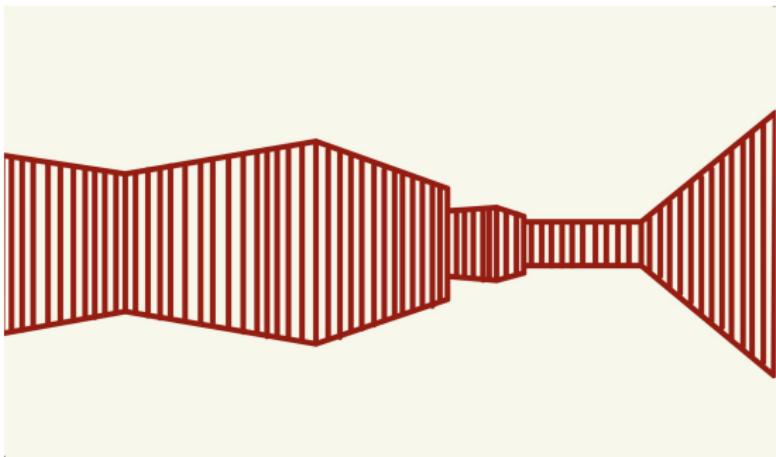
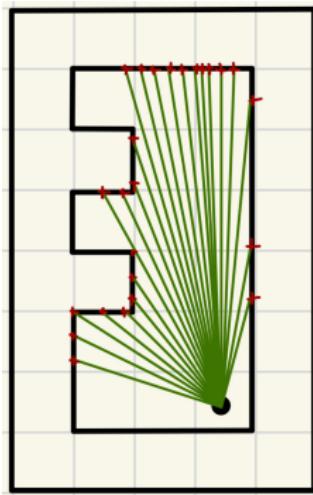
## 4 Conclusion

# Différentes stratégies pour un rendu 3D

## Approche historique

### Raycasting

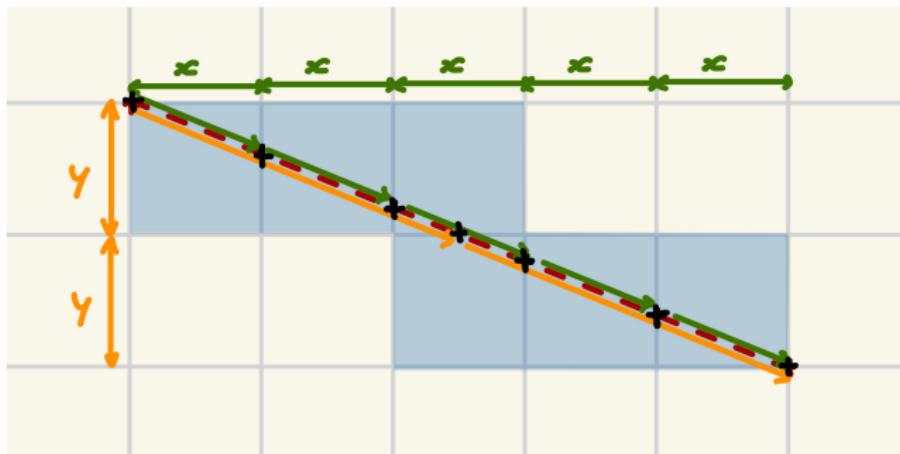
- Projection de rayons depuis la position du joueur.
- Pour chaque colonnes de pixels de la fenêtre.



# Différentes stratégies pour un rendu 3D

## DDA

- Conçu pour la rasterisation de lignes
- Repose sur l'itération linéaire
- Employé pour déterminer où les rayons projetés intersectent avec les objets de l'environnement

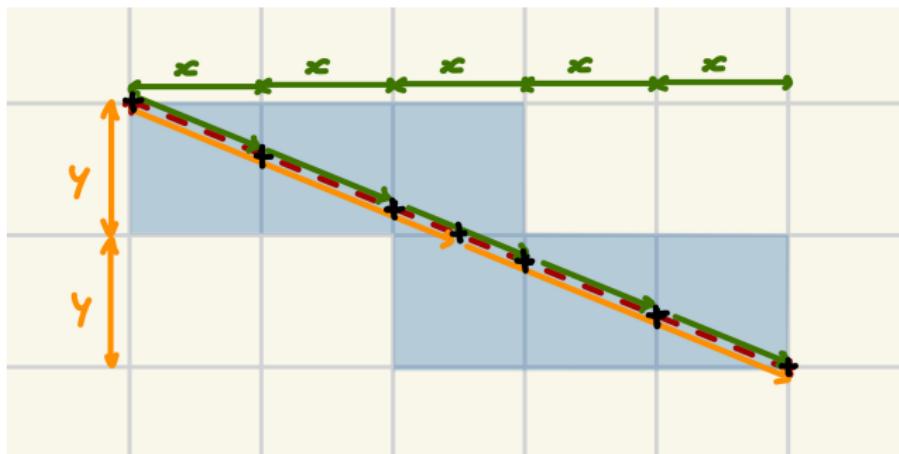


# Différentes stratégies pour un rendu 3D

## DDA

### Fonctionnement de DDA.

- Déplacement vertical et horizontal d'une unité.
- Comparaison entre les distances parcourus sur le segment.
- Choix de la distance la plus petite.

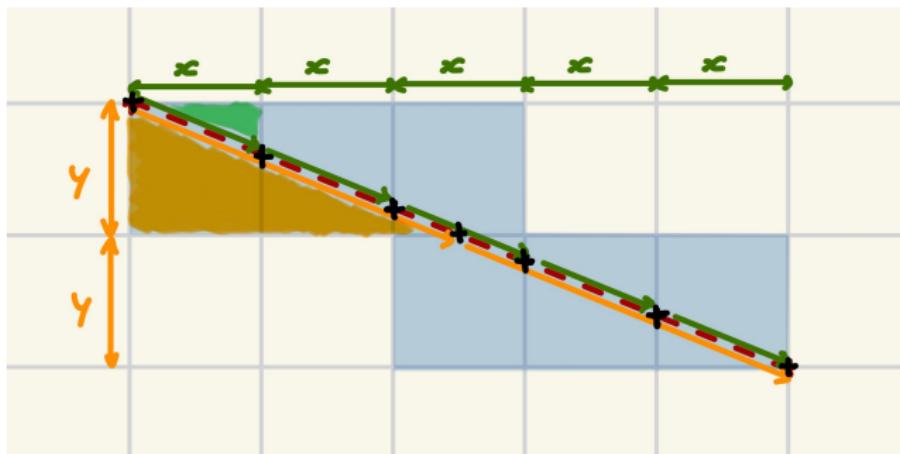


# Différentes stratégies pour un rendu 3D

## DDA

### Fonctionnement de DDA.

- Déplacement vertical et horizontal d'une unité.
- Comparaison entre les distances parcourus sur le segment.
- Choix de la distance la plus petite.

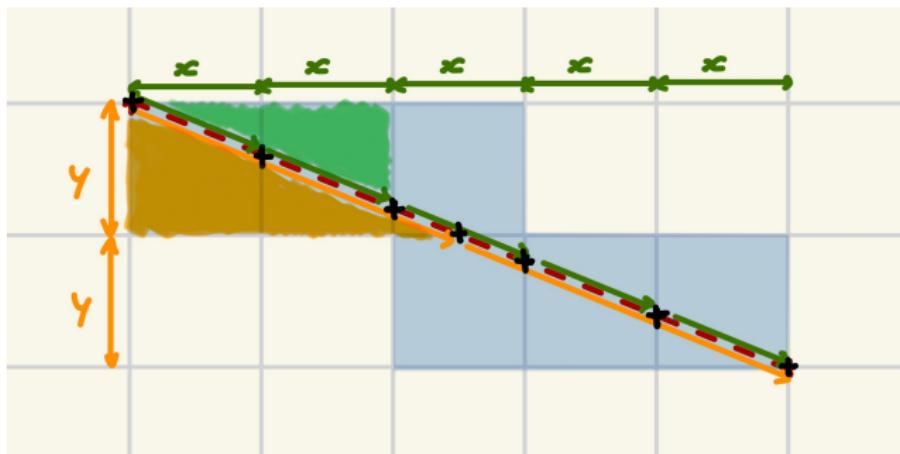


# Différentes stratégies pour un rendu 3D

## DDA

### Fonctionnement de DDA.

- Déplacement vertical et horizontal d'une unité.
- Comparaison entre les distances parcourus sur le segment.
- Choix de la distance la plus petite.

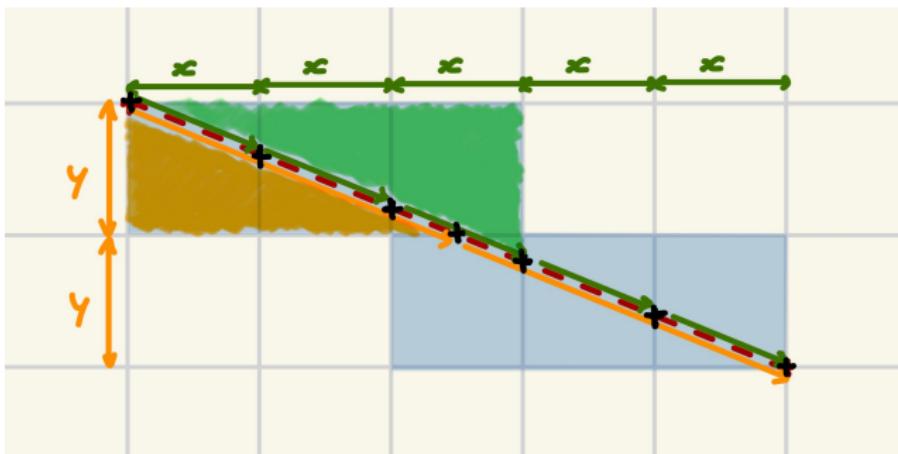


# Différentes stratégies pour un rendu 3D

## DDA

### Fonctionnement de DDA.

- Déplacement vertical et horizontal d'une unité.
- Comparaison entre les distances parcourus sur le segment.
- Choix de la distance la plus petite.

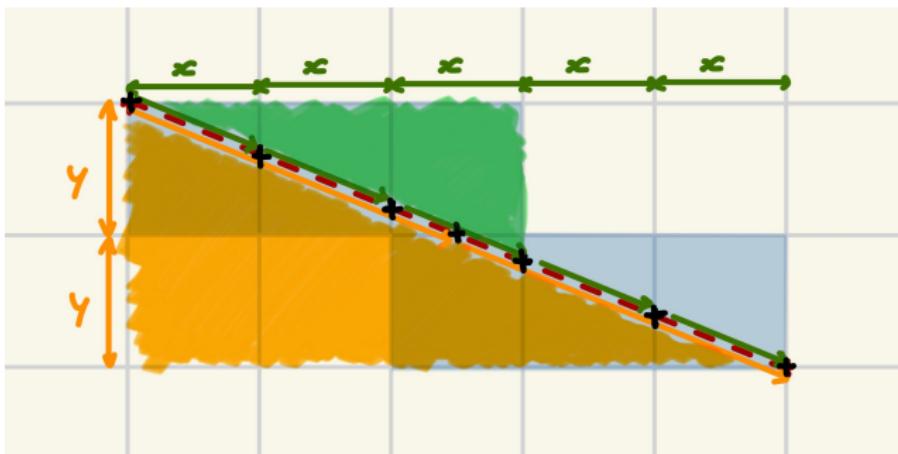


# Différentes stratégies pour un rendu 3D

## DDA

### Fonctionnement de DDA.

- Déplacement vertical et horizontal d'une unité.
- Comparaison entre les distances parcourus sur le segment.
- Choix de la distance la plus petite.

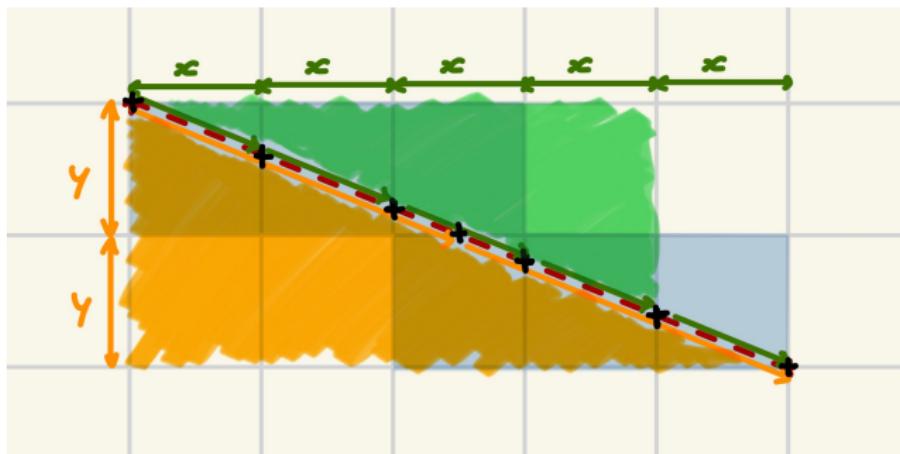


# Différentes stratégies pour un rendu 3D

## DDA

### Fonctionnement de DDA.

- Déplacement vertical et horizontal d'une unité.
- Comparaison entre les distances parcourus sur le segment.
- Choix de la distance la plus petite.

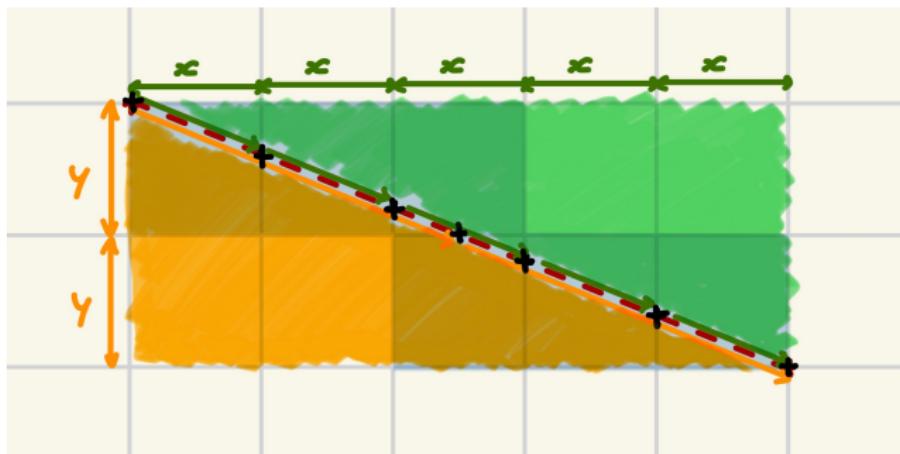


# Différentes stratégies pour un rendu 3D

## DDA

### Fonctionnement de DDA.

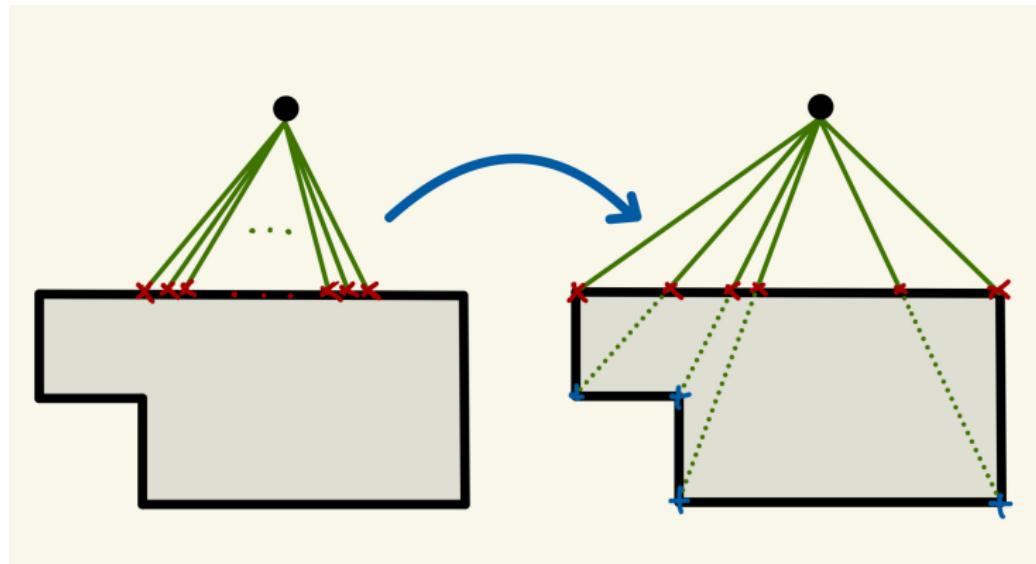
- Déplacement vertical et horizontal d'une unité.
- Comparaison entre les distances parcourus sur le segment.
- Choix de la distance la plus petite.



# Différentes stratégies pour un rendu 3D

## Approche moderne (Line Of Sight)

- Envoyer un rayon pour chaque sommet de chaque mur



# Plan

## 1 Introduction

## 2 Algorithmes et techniques

- Construction de mur
- Les collisions
- Différentes stratégies pour un rendu 3D

## 3 Implémentation

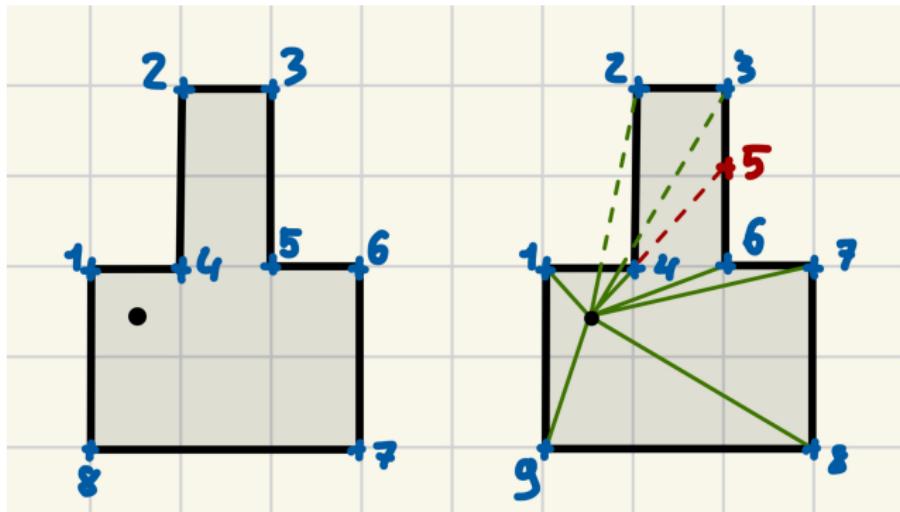
- Les rendus
- Les portails

## 4 Conclusion

# Les rendus

## Deux étapes

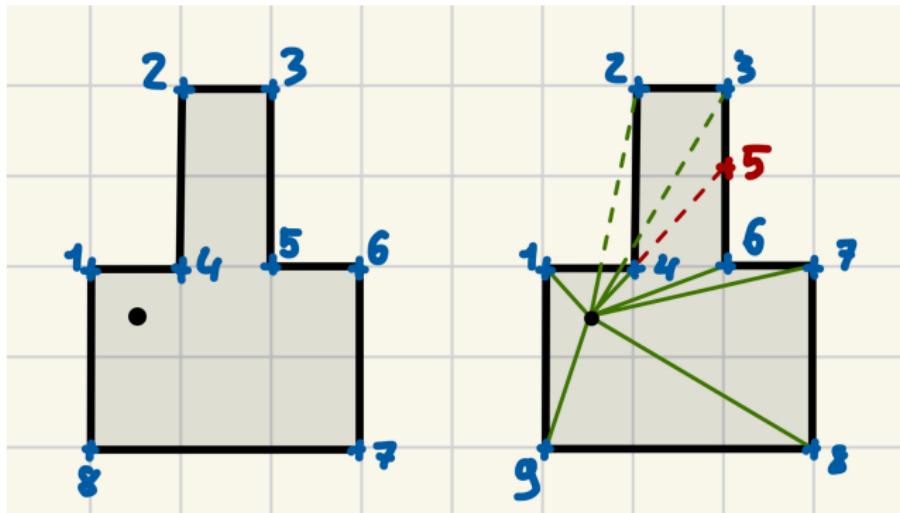
- Récupération des sommets triés
- Envoie des rayons dans l'ordre avec DDA



# Les rendus

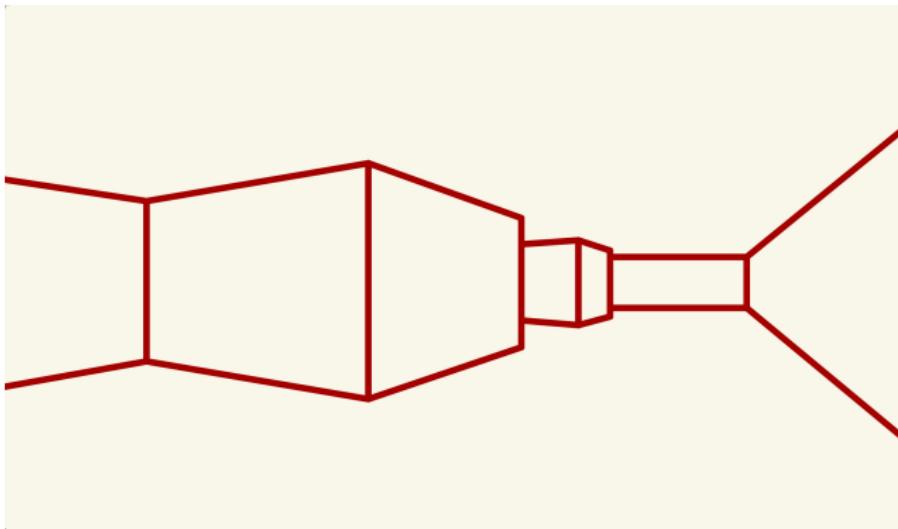
## Trois cas possibles

- Le sommet est atteint et on arrête le rayon.
- Le sommet n'est pas atteint donc on arrête le rayon.
- Le sommet est atteint mais on continue le rayon.



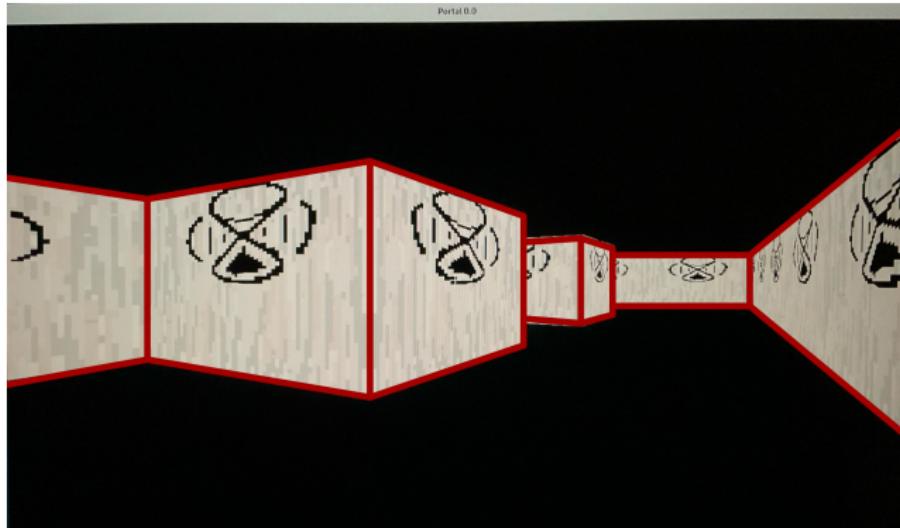
# Les rendus

- Résultat sans textures.



# Les rendus

- Résultat avec textures.



# Plan

## 1 Introduction

## 2 Algorithmes et techniques

- Construction de mur
- Les collisions
- Différentes stratégies pour un rendu 3D

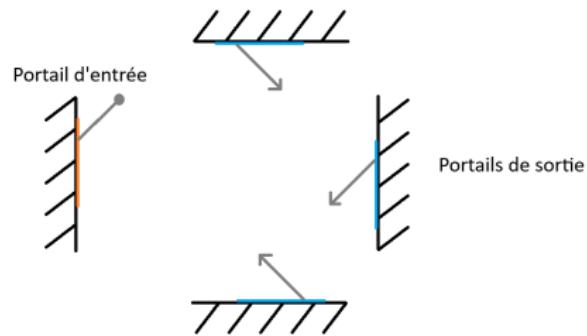
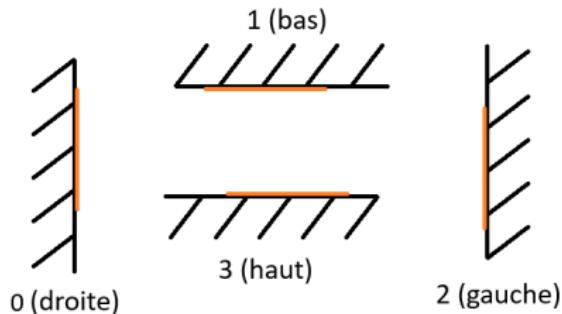
## 3 Implémentation

- Les rendus
- **Les portails**

## 4 Conclusion

# Les portails

## La téléportation



# Les portails

## Calculs de position et de direction

- Calcul de la nouvelle direction

$$\alpha = \beta + \pi + (Ps.dir - Pe.dir) \times \pi/2$$

- Nouvelles positions possible

$$(Ps_x + (J_x - Pe_x); Ps_y - (J_y - Pe_y))$$

$$(Ps_x + (J_y - Pe_y); Ps_y + (J_x - Pe_x))$$

$$(Ps_x - (J_x - Pe_x); Ps_y + (J_y - Pe_y))$$

$$(Ps_x - (J_y - Pe_y); Ps_y - (J_x - Pe_x))$$

- Déterminisation du cas

$$Ps.dir + Pe.dir \equiv cas_n(4)$$

# Conclusion

Merci de votre attention.

# Questions

Avez-vous des questions ?