

## Rapport de la conception du projet “Des momies et des pyramides” en Processing

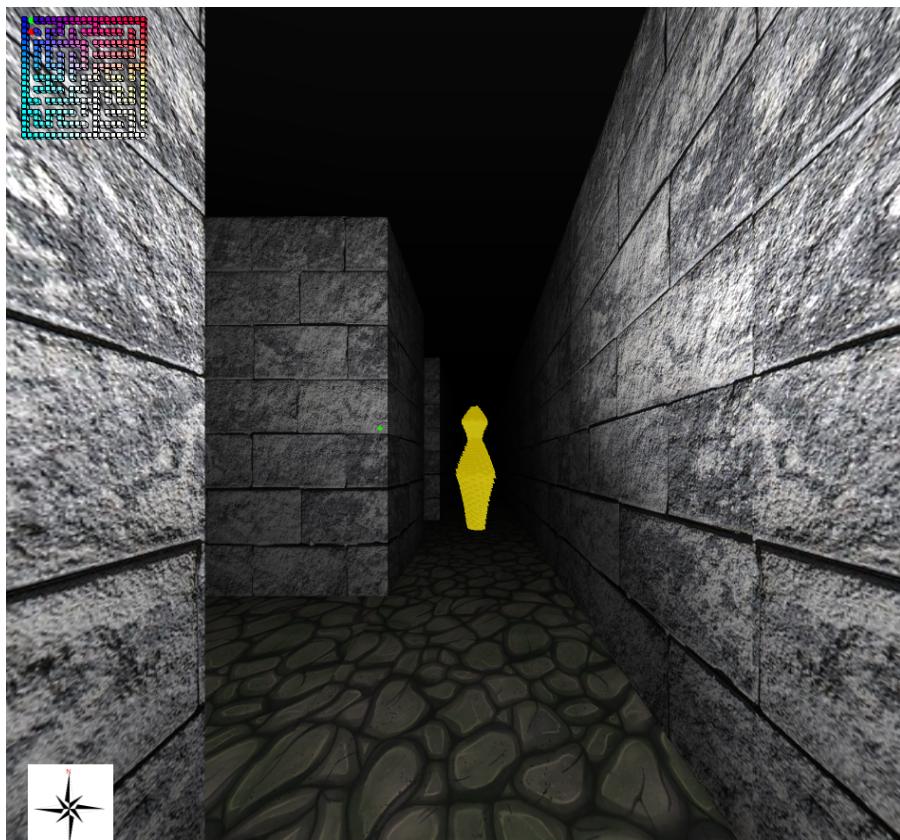
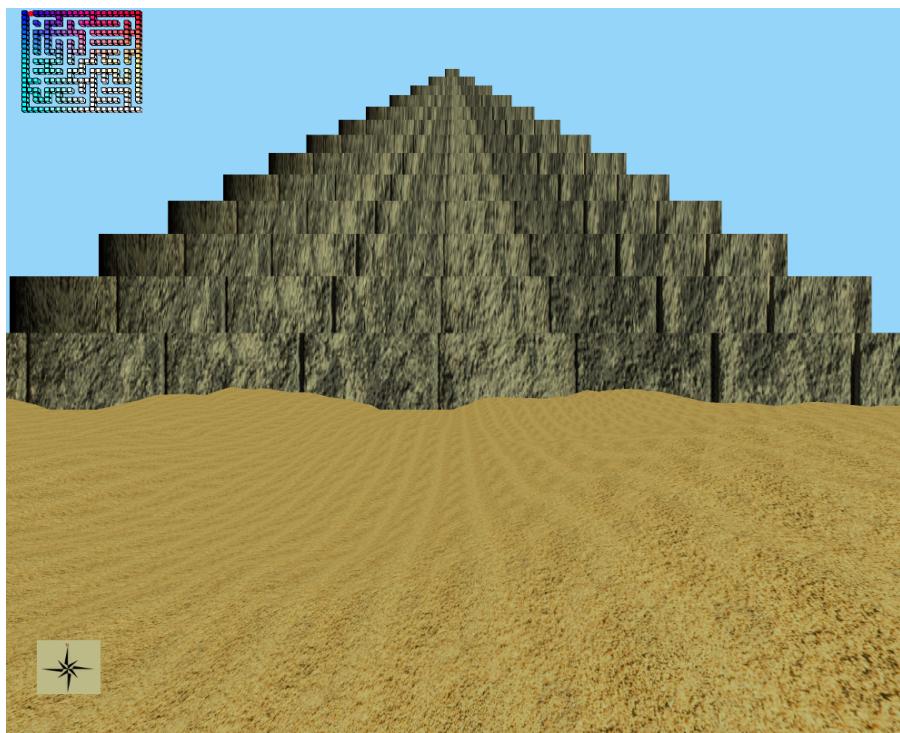
### Description du jeu :

Le projet est créé en utilisant le langage de programmation Processing. Le joueur se retrouve plongé dans une pyramide mystérieuse composée de plusieurs étages de labyrinthe complexe. Le joueur rentre dans une aventure obscure où il doit explorer les couloirs sombres de la pyramide pour trouver la sortie, tout en évitant une momie effrayante qui se déplace de manière aléatoire dans chaque étage de la pyramide.

Le joueur a la liberté de sortir de la pyramide et d'explorer l'extérieur, ajoutant ainsi une dimension de monde ouvert au jeu. Ainsi le joueur se retrouve dans un environnement de désert avec un ciel bleu et du sable à perte de vue, ce qui crée une atmosphère captivante et immersive. Pour aider le joueur dans sa quête, une boussole indique toujours la direction du nord, et une mini-map affiche la position du joueur dans le labyrinthe ou dans le désert.

La tâche du joueur est de naviguer habilement dans les niveaux du labyrinthe, en utilisant sa perspicacité et ses compétences pour éviter la momie et trouver la sortie, tout en prenant garde à ne pas se perdre dans les passages complexes de la pyramide.

### Réalisation du jeu:



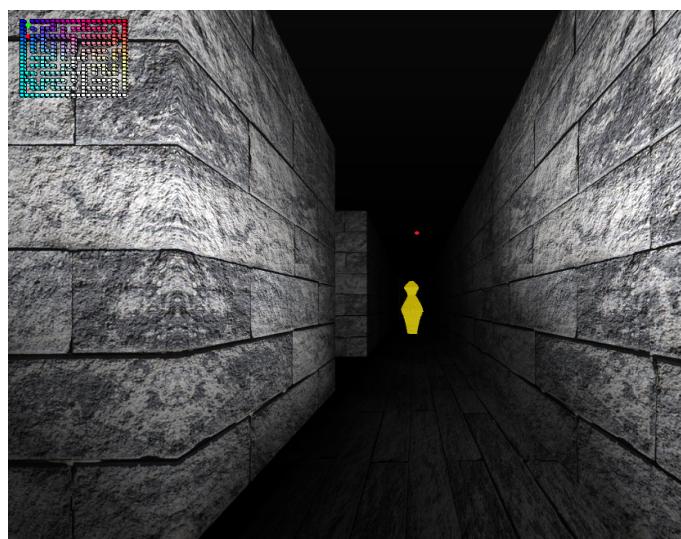
## Partie I: L'intérieur

## I. Empilement des labyrinthes:

Notre jeu est composé de plusieurs labyrinthes empilés pour créer une forme de pyramide. Les labyrinthes sont stockés dans un tableau en trois dimensions (3D) qui a été créé et initialisé dans la fonction 'setup' de Processing. À mesure que le joueur progresse dans les niveaux, les labyrinthes se rétrécissent pour reproduire le concept de la pyramide d'Égypte. Cette approche unique ajoute une dimension intrigante au gameplay, où le joueur doit naviguer à travers des passages de plus en plus étroits à mesure qu'il s'approche du sommet de la pyramide.

Exemple:

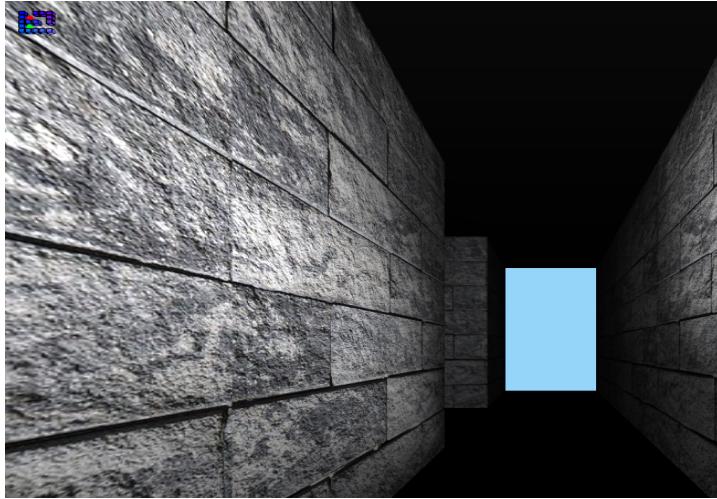
niveau 0



niveau 2



niveau 4



## II. Déplacement du joueur dans les laby

Le joueur se déplace dans le labyrinthe en utilisant les touches de direction (droite, gauche, haut, bas) pour naviguer à travers les passages vides et éviter les murs.

Une fonctionnalité intéressante est la possibilité du changement de niveau en utilisant la touche '**u**', mais cela n'est possible que s'il a **réussi** à trouver son **chemin** jusqu'à **la dernière case** du labyrinthe dans le niveau actuel.

Cette fonctionnalité permet au joueur de progresser vers les niveaux supérieurs s'il a résolu le labyrinthe précédent avec succès. Cependant, une fois qu'il est passé à un niveau supérieur, il ne peut plus revenir en arrière.

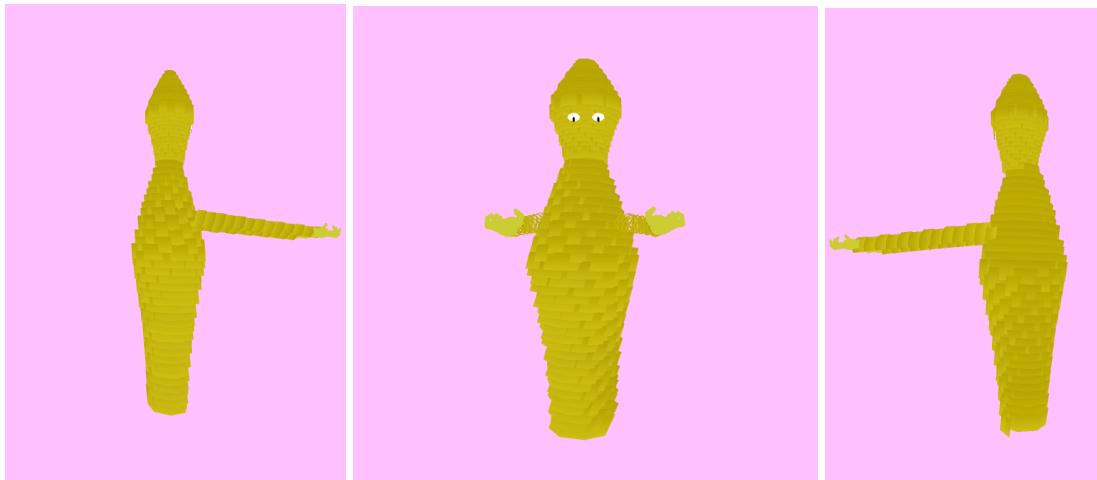
Ce système de déplacement du joueur, ainsi que la possibilité de passer d'un niveau à l'autre et les restrictions pour sortir du labyrinthe, sont implémentés dans la fonction `keyPressed()`, qui enregistre les touches de l'utilisateur et assure le bon fonctionnement du jeu en respectant ses contraintes. Cette fonction a été modifiée de sorte à faire le nécessaire.

### Bonus:

Pour ajouter un niveau de difficulté nous avons ajouté une option à la momie, qui constitue que si la momie attrape le joueur elle le remet au niveau de départ c-a-d qu'il revient à la case de départ au niveau 0 et perd tout ce qu'il avait surpassé comme niveau. cette option est ajoutée à la fonction '**draw**' de la ligne 25 à 30 qui peut être décommenter pour activer l'option bonus.

### III. Momie

#### a- création Momie



La momie est un "**groupShape**". Ce groupe contient plusieurs sous-shapes telles que les mains, le corps, la tête et les iris, qui sont ajoutés à l'aide de la fonction "**shape.addChild(shape)**".

La tête, le corps et les bras de la momie sont tous des "**quad\_Strip**", une classe de forme qui permet de dessiner des quadrilatères texturés. Nous avons utilisé la fonction "**acos(sin(x))**" pour créer la forme du corps.

Nous avons utilisé la fonction "translate" pour positionner les différentes formes au bon endroit dans la momie. De plus, nous avons utilisé la fonction "rotate" pour permettre à la momie de faire des rotations dans le labyrinthe tout en restant bien affichée.

#### b- mouvement Momie

La fonction "**MoveMomie()**" est responsable de la progression de la momie dans le labyrinthe. Nous avons initialisé les variables "**posxM**" et "**posyM**" pour déterminer la position de la momie. Si elle est dans le labyrinthe et qu'il n'y a pas de mur, sa position est incrémentée et des translations sont effectuées.

Pour garantir un mouvement **aléatoire** de la momie, nous avons utilisé la fonction "**random**". La momie tourne à gauche ou à droite avec une probabilité de 50%. Nous avons également ajouté une condition "**if (frameCount%10==0)**" pour changer la direction aléatoirement toutes les 10 frames, même s'il n'y a pas de mur.

Un problème que nous avons rencontré était que la momie avançait un peu trop rapidement pour l'affichage. Pour résoudre ce problème, nous avons modifié la fonction "draw". Lorsque nous affichons le labyrinthe, la momie est

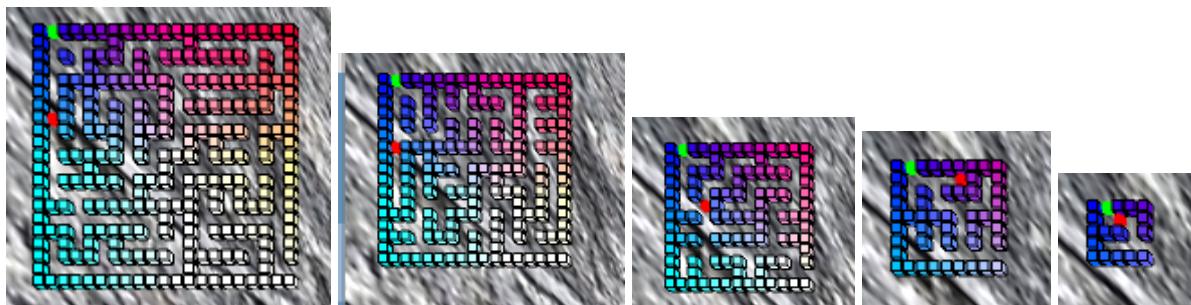
déplacée toutes les 19 frames en incrémentant sa position ou sa rotation. Sinon, la momie est affichée dans sa position précédente.

#### IV. La map

Pour créer une mini-map affichée en haut à gauche qui indique la position du joueur sous forme d'une sphère verte à la fois à l'intérieur et à l'extérieur du labyrinthe, nous avons repris la partie du code fourni par le professeur qui le faisait déjà pour le niveau 0. Nous l'avons ensuite modifié afin qu'elle prenne en compte l'étage où se trouve le joueur, et qu'elle affiche le labyrinthe correspondant sur la mini-map. Pour cela, nous avons utilisé une variable "level" pour suivre l'étage actuel du joueur, et en fonction de cette valeur, nous affichons le bon labyrinthe sur la mini-map. Nous avons également utilisé les coordonnées du joueur pour mettre en évidence sa position sur la mini-map. Nous nous assurons de mettre à jour la mini-map à chaque déplacement du joueur, de manière à refléter sa position en temps réel, que ce soit à l'intérieur ou à l'extérieur de la pyramide.

De plus, sur la même mini-map, nous avons également ajouté l'affichage du déplacement de la momie sous la forme d'une sphère rouge. Cela permet au joueur d'avoir un aperçu visuel du déplacement de la momie dans le labyrinthe. En combinant les mouvements du joueur et de la momie sur la mini-map, le joueur peut anticiper les déplacements de la momie et prendre des décisions stratégiques pour éviter d'être attrapé. Cela ajoute un élément de suspense et de stratégie au jeu, rendant l'expérience de jeu encore plus captivante pour le joueur.

#### Exemples des maps de notre jeu:

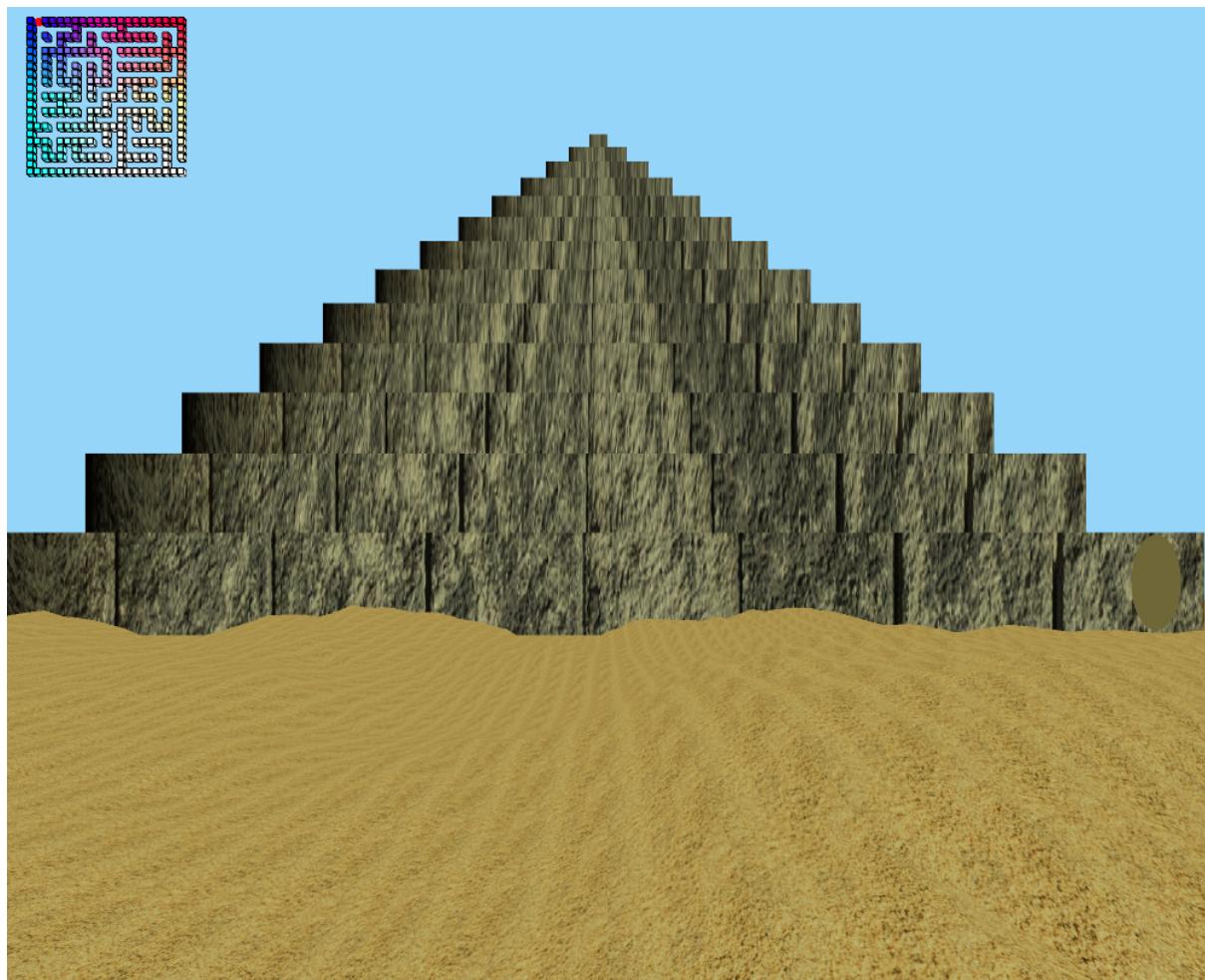


## Partie 2: L'exteriere

### I. la pyramide

La pyramide dans notre jeu est un environnement en plein air où le joueur se retrouve à l'extérieur du labyrinthe et a une vue complète de la pyramide dans son ensemble. Pour créer cette belle pyramide, nous avons utilisé la fonction "**beginShape(QUADS)**" pour dessiner les murs (rectangles) verticaux sur les quatre côtés, ainsi que des losanges au-dessus de chaque côté pour former un étage de la pyramide. Nous avons fourni les quatre sommets ("**vertex**") nécessaires pour chaque mur et losange afin de les dessiner correctement. Les murs et les losanges se rétrécissent et deviennent de plus en plus hauts jusqu'à ce que les quatre losanges se touchent et forment le sommet de la pyramide. Ce concept a été implémenté dans une fonction "**empPyr()**" qui est appelée dans la fonction "**draw**" uniquement si le joueur se trouve à l'extérieur du labyrinthe, afin d'éviter une surcharge de calculs qui pourrait ralentir l'affichage du jeu. Cela permet d'offrir une vue panoramique et réaliste de la pyramide tout en optimisant les performances du jeu.

#### exemple de vu sur la pyramide:



## II. Désert

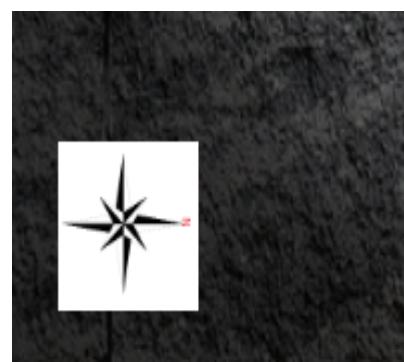


Le désert est une grille de quads créée en utilisant deux boucles "for". Les sommets des quads ont des positions en x, y, z et des coordonnées. La fonction "noise" est utilisée pour générer des variations dans la hauteur des quads en fonction de leur position sur la grille.

On essaie de donner la même valeur du noise pour que ma shape forme des mini dunes.

La texture du désert “sable” est une image du sable.

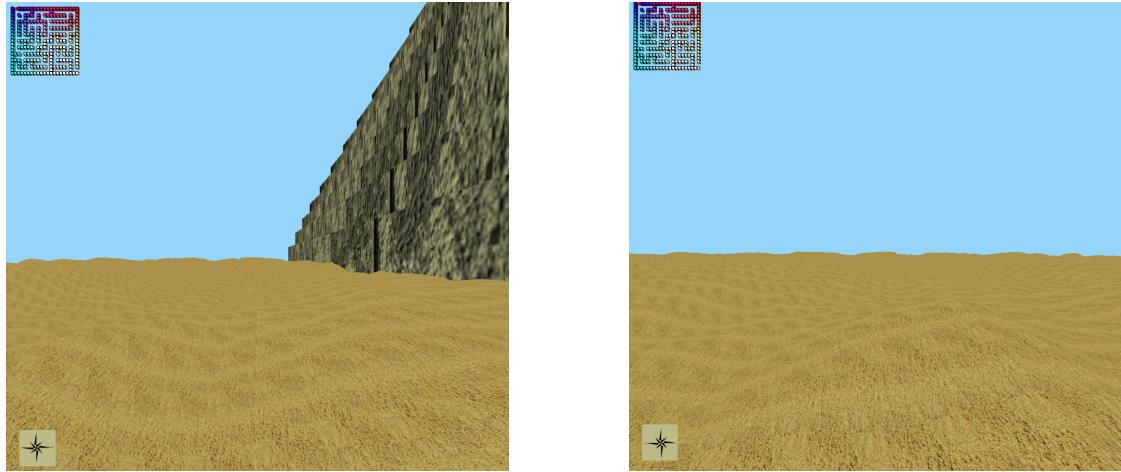
## III. Boussole



On utilise une image pour la boussole qui fait une rotation selon la position du joueur.

On enregistre la currentDirection et on initialise une variable Bdir qui incrémente ou décrémente à chaque mouvement du joueur (chaque keyPressed) , et selon cette direction, une rotation est faite.

#### IV. Déplacement du joueur a l'exterieure



Le déplacement du joueur à l'extérieur du labyrinthe dans notre jeu a été mis en place pour offrir une expérience de jeu réaliste et immersive. Lorsque le joueur se trouve à l'extérieur du labyrinthe, il peut se déplacer librement dans l'environnement en plein air et explorer les alentours de la pyramide. Pour cela, nous avons implémenté des fonctionnalités de déplacement du joueur, telles que la gestion des touches du clavier pour les déplacements vers l'avant, l'arrière, la gauche et la droite, détecter si le joueur est à l'extérieur pour afficher les éléments de l'extérieur tel que le ciel, pyramide, sable .

En outre, pour offrir une expérience visuelle réaliste, nous avons utilisé des graphismes appropriés pour représenter l'environnement extérieur, avec des textures, des couleurs et des effets visuels appropriés pour donner l'impression que le joueur se déplace dans un environnement réel en dehors du labyrinthe.

**V. Parties à améliorer:**

1- Au lieu d'utiliser une image pour la boussole, on pourrait créer un objet 3D avec une aiguille qui fait la rotation.

2- Utiliser des pShaders pour l'amélioration de l'affichage visuel en termes de couleurs, lumières..

3- le changement de niveau est implémenté par un keypressed touch 'u' mais ce serait mieux d'ajouter un téléporteur dans l'affichage , ou des escaliers.

4- Pour le déplacement de la momie on translate la momie d'une case à chaque pas mais cela peut être amélioré en faisant un déplacement sous forme de translation ralentie de pixel en ajouter une variable anim initialiser à 20 par exemple et qui decremente a chaque image qui fait que la momie s déplace comme notre joueur d'une manière plus fluide.