

PolyMaze

```
1  """PolyMaze Maze Generator"""
2  #
3  #
4  __author__ = "Lemaire Pierre & Rawat Eshane"
5  __version__ = "Random Maze generator 1.1"
6  __date__ = "2015-03-30"
7  #
8  #
9  import random
10
11 class maze_generator(object):
12     def __init__(self):
13         #Maze dimensions
14         self.width = 50
15         self.height = 50
16         #Initial stack
17         self.maze = []
18         #Directions within the maze for the constructor
19         self.xdir = [0,1,0,-1]
20         self.ydir = [-1,0,1,0]
21
22     def new_maze(self, x, y):
23         self.maze = [[0 for i in range(self.width)] for j in range(self.height)]
24         return self.carve_maze(x,y)
25
26 """PolyMaze Maze Rendering"""
27 #
28 #
29 __author__ = "Lemaire Pierre & Rawat Eshane"
30 __version__ = "Maze rendering 1.1"
31 __date__ = "2015-03-30"
32 #
33 #
34 import Mazegen
35 from PIL import Image, ImageOps, ImageChops
36
37 class maze_rendering(Mazegen.maze_generator):
38     def __init__(self):
39         Mazegen.maze_generator.__init__(self)
40         #Basic img dimension to be used.
41         self.imgW = 512
42         self.imgH = 512
43
44     def render_maze2D(self):
45         #simple method using pillow lib and carve_maze to "draw" the maze
46         image = Image.new("RGB", (self.imgW, self.imgH))
47         pix = image.load()
48         maze = self.new_maze(0,0)
49         #cell building, white if path, black if wall
50         for cy in range(self.imgH):
```



Pierre Lemaire & Eshane Rawat

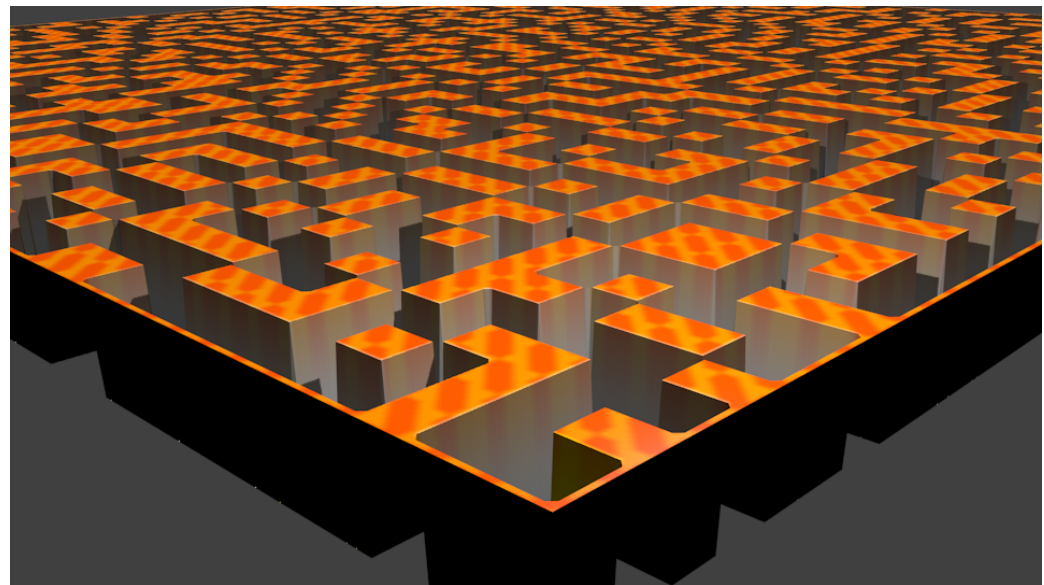
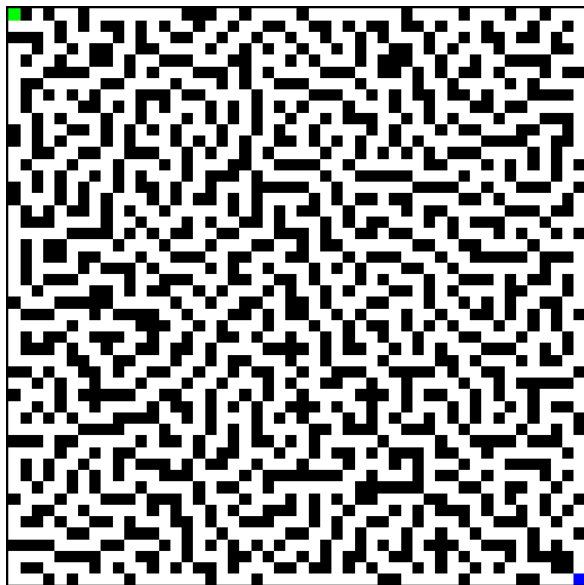
What's in the maze?



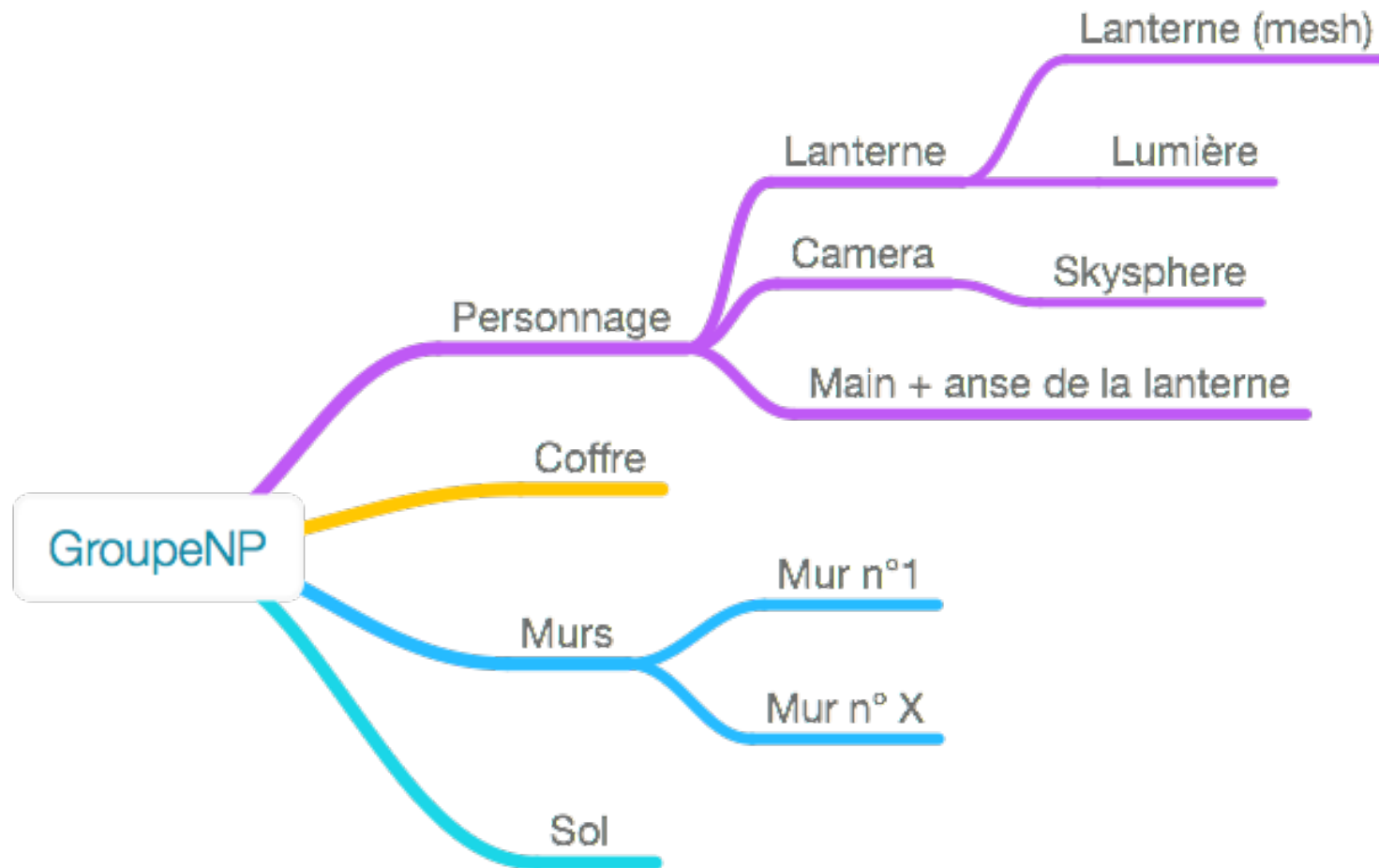
regroupe 2 projets utilisant Python comme langage:

- un jeu utilisant le moteur panda3D;
- un générateur de labyrinthe 2D et 3D (.obj).

Chacun de ces projets aborde des aspects différents, vu au cours des TD. Nous nous concentrerons cependant sur le jeu, la simplicité du générateur ne nécessitant pas d'approfondissement. Voici néanmoins un bref aperçu des résultats possibles grâce à ce programme:



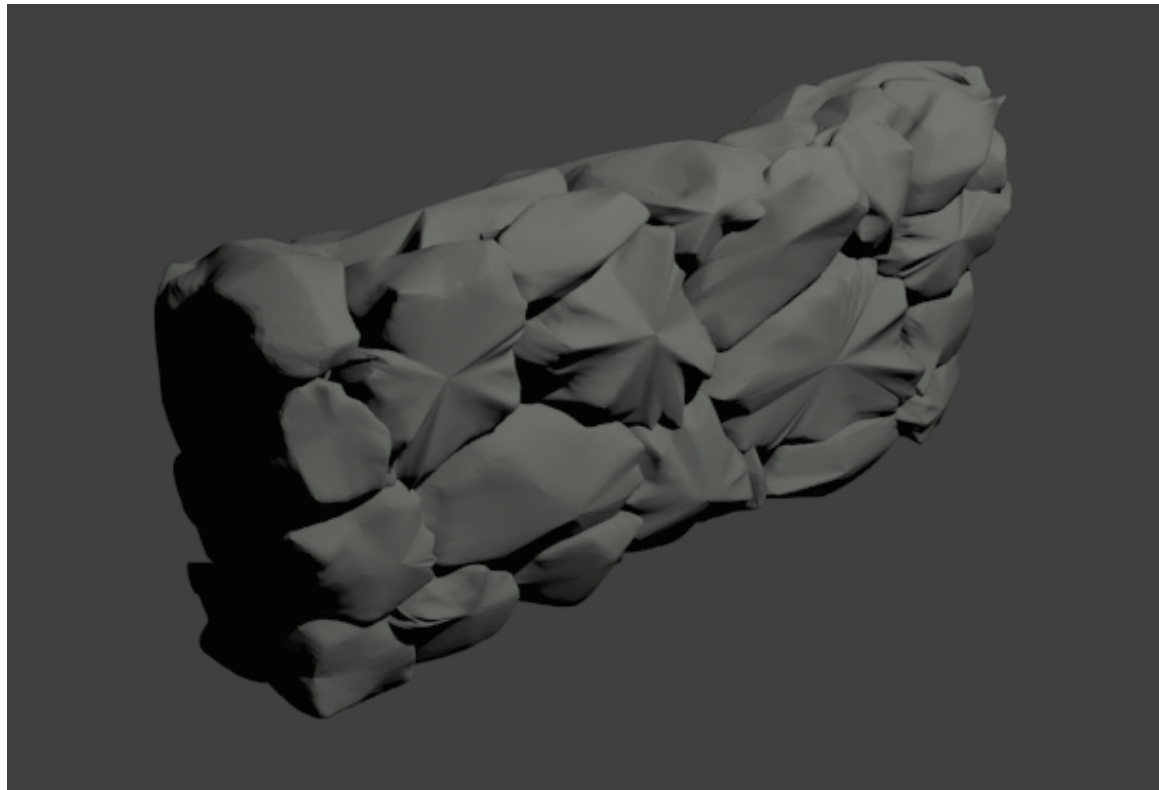
Graphe de scène



Art & Mesh

L'ensemble de nos mesh ont été créés via Blender, par nos soins. Nous avons choisi une esthétique « low poly » tant pour des raisons artistiques que pour des raisons de distribution. En effet, si nous avions conservé nos mesh dans leur forme d'origine, il n'aurait pas été possible de faire tourner le jeu sous la majorité des machines.

Exemple de mesh d'un mur comportant plus de 87000 faces.



Callbacks

- Gestion des déplacements (commandMove)
- Mouvement de la souris (mouse_get)
- Condition de défaite (win_lose)
- Animation de la lumière (light_animation)
- Collisions (Collision handler → pusher)
- Pause (other_commands)

Intelligence des composantes

- Générateur de labyrinthe → les labyrinthes seront toujours résolubles
- Jeu PolyMaze → Commandes intuitives & gestion de la caméra réaliste:
Il est impossible de tourner la caméra à des angles irréels



- Univers graphique
 - Esthétique low poly
 - Design unique, propre à PolyMaze
- Projet Open → Code source disponible via Github



Développement futur

Gameplay & (re)jouabilité

- Fusion des deux projets afin de créer une infinité de niveaux de PolyMaze
- Ajout de pièges & augmentation de la taille du labyrinthe afin d'offrir différents niveaux de difficulté
- Ajout monstres et d'équipements dépendant du niveau afin d'offrir une grande amplitude de gameplay possible
- Ajout de condition de victoire

Univers

- Ajout d'une bande son originale
- Amélioration des mesh existants & ajouts de nouveaux mesh de décor
- Ajout de nouveaux thèmes de labyrinthe (ex: thème hivernal, halloween,...)

Ressources

- TD Imagerie Numérique
- Panda3D forum
- Documentation:
 - Blender
 - Panda3D