



Visual Computing - Final Presentation

Lucius Vinicius, 96123
Martinho Tavares, 98262

Idea

- A game prototype where you should navigate a 3D cubic labyrinth;
- The scene will be mostly dark, and the player will have the ability to light it using a flashlight and a lamp that can be placed in the map;
- The inside layout is occluded by walls and windows. The player has to position the camera in order to get a better view;
- The map will be populated with objects and mobs;
- Random events to disorient the player;
- Implemented in Python with Panda3D.



Transformations

Translation



The player's and mobs' movement in the cube;

Rotation



The player is looking to the direction it's heading (rotation around the origin) and an animal rotating around the cube (arbitrary point rotation);

Scale



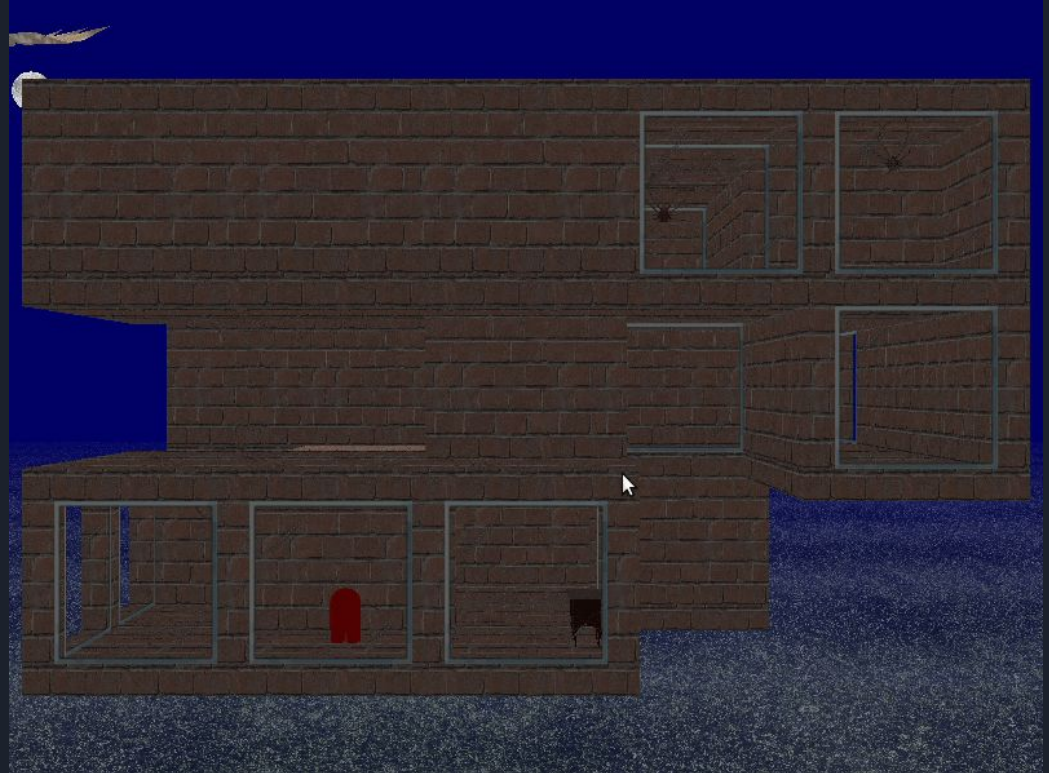
Scenario mobs with random scale;

View and Projection

Camera

Outside view into the scene and manually rotate and zoom in/out around the map.

Zooming in and out is limited to not collide into geometry, and it affects flashlight intensity.



Random events

Some random events were created to make gameplay more chaotic, and experiment different visual effects.

Orthographic



Camera switches from a perspective lens to an orthographic lens with a similar view

Lightning strike



Brightly illuminate the whole scene, with a lightning texture following the camera behind the scene

Flashlight flicker



Flickering flashlight intensity

Lighting

Spot Light Source



The spotlight object;

Point Light Source



The resource which the
player can place on the
map;

Lighting

Directional Light Source



The light emitted by the moon*.
Affects top-most roof of the labyrinth;

Ambient Light Source



Default light for objects;

** in the figure the light has double the normal intensity for demonstration purposes*

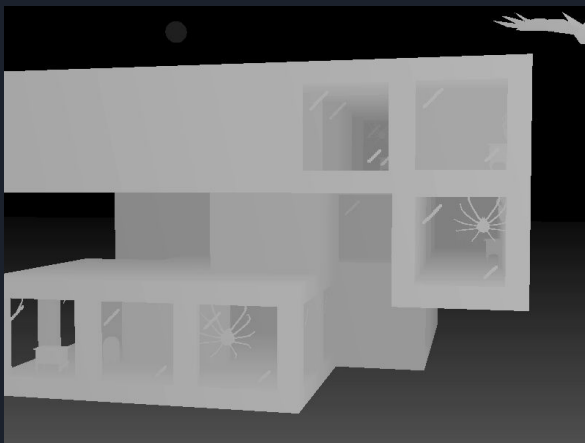
Fragment shader

Deferred shading: render the scene into an image (texture), and apply post-processing to that image in a fragment shader.

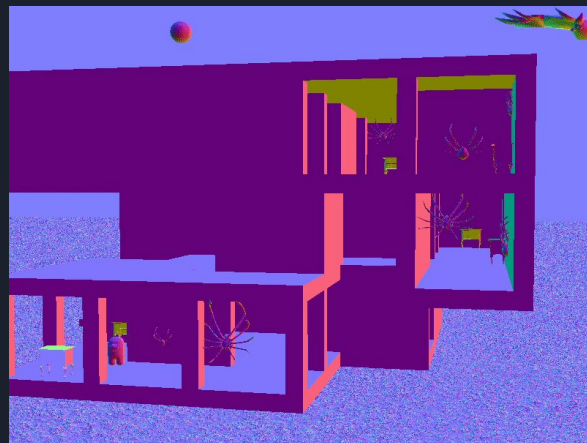
Base



Depth buffer



Normal buffer



Fragment shader

Flashlight at the mouse position. Take into account normals and depth.

Base

X

Flashlight mask

=

Final color



Fragment shader

Keep lit geometry on the scene even when unlit by the flashlight: threshold on HSV value (max RGB component).

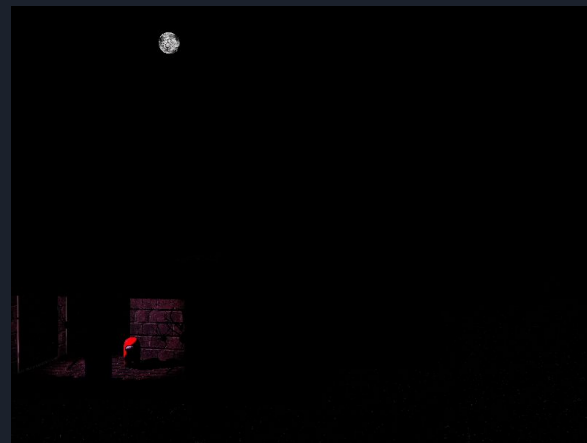
HSV Value



Smoothstep

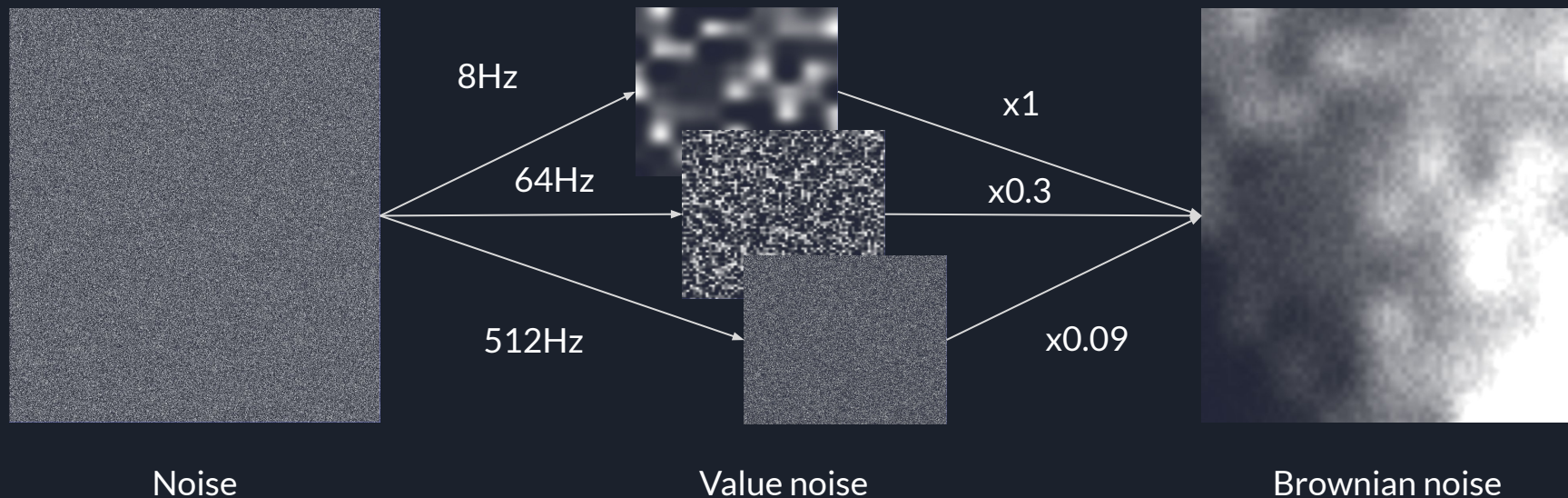


Mix



Fragment shader

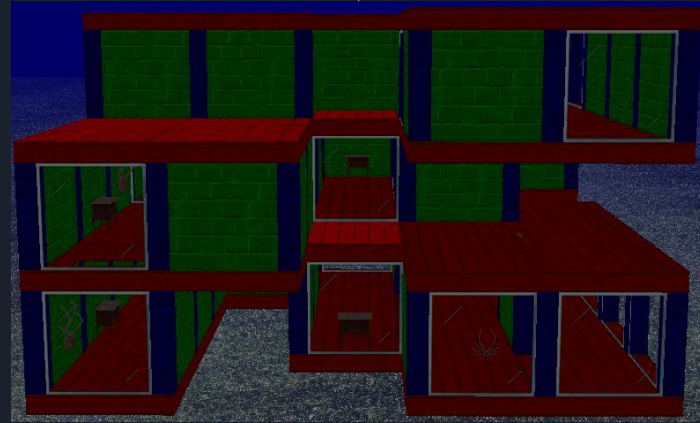
Flashlight flicker effect: sample Brownian noise over time and use as light intensity.



Geometric Modelling



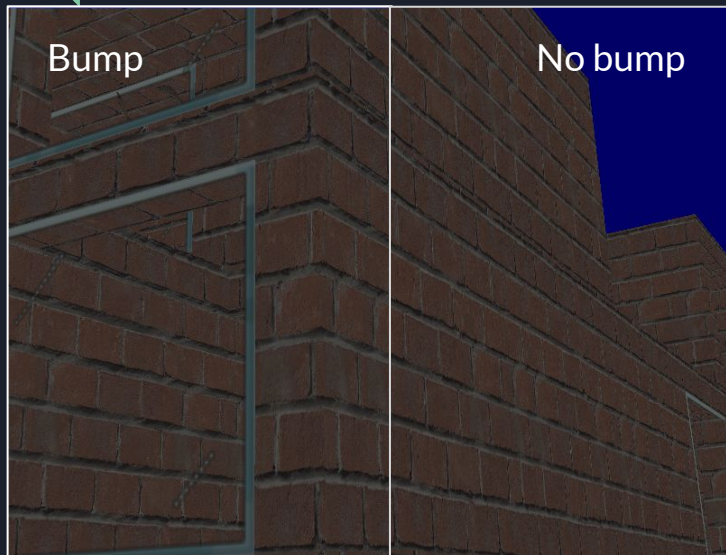
Player model, objects'
models and mobs'
models;



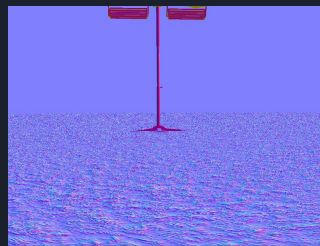
Map mesh dynamically created in
rectangular blocks (UVs, normals,
binormals, tangents and color);

```
1  +_+_+_+
2  !...S...!
3  +_+_+_+_+
4  !.....!
5  +_+_+_+_+_+
6  !.....!
7  +_+_+_+_+_+
8  !.....!
9  +_+_+_+_+_+
10 +_+_+_+_+_+
11 +_+_+_+_+_+
12
13
14      +_+
15      |.|
16      +_+ +_+
17      |.| |.|
18      +_+_+_+_+
19      !.....|
20 +_+_+_+_+_+_+
21 |.....! !..!
22 +_+_+_+_+_+_+
23 !X.....!...!
24 +_+_+_+_+_+_+
25
26 +_+_+_+_+_+_+
27 !.....X]
28 +_+_+_+_+_+_+
29 |.....|
30 +_+_+_+_+_+_+
31 +_+_+_+_+_+_+
32      |.|
33      +_+_+_+
34      |...|
35      +_+_+_+_+
36      |.....F|
37      +_+_+_+_+
```

Textures



The walls: Bump mapping, tiling wrap;



Grass texture with normal mapping;

Challenges and Problems

Problems which persisted:

- Displacement mapping on the terrain;
- Player light performance;

Challenging problems fixed:

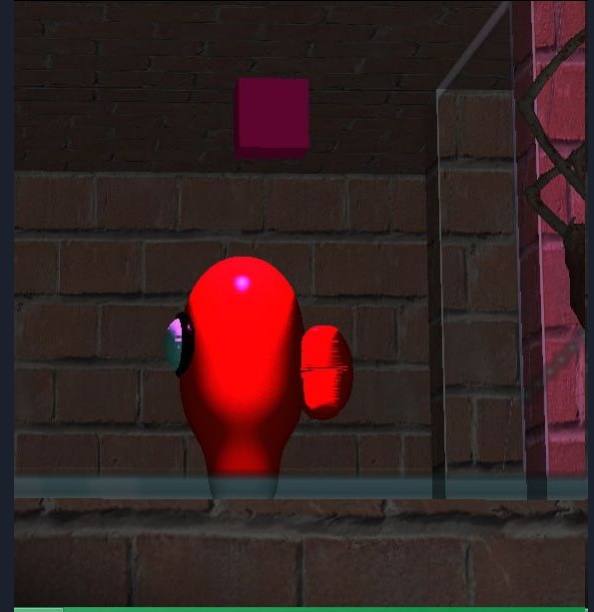
- Light “saturated” easily (limited brightness at the scene);
- Player model and its materials (model-space coordinates off-center, ambient material);
- Camera and player relative math;
- Flashlight shader (knowing how to apply the flashlight effect, with which techniques);



Attempt at better player light performance

Alterations on the Initial Goal

- Initially, the player would be able to put multiple lights. But due to the performance issues brought by shadow mapping, we limited it to 1;
- The flashlight was supposed to be a Panda3D Spotlight object, but complications with tracking the mouse on the 3D scene led us to consider creating the effect in GLSL the more appropriate approach;
- Flat Shading to the player model instead of the mobs because it was harder to see the effect on them;



Player's flat shading



Future Work

- Fix the grass's displacement map and player light performance problems;
- Insert a walking monster on the labyrinth that slowly follows the player with a pathfinding algorithm, disrupting their lights;
- More variety between mobs and objects;



Questions?



Self-evaluation: 18
Work distribution: 50/50