

Environment Restoration

情報電子工学科2年

名島和希

チャンドン

ティオ シーディエン

目的

- 地形作成アルゴリズム(パーリンノイズなど)を理解したい
- マインクラフトのようなゲームでパズルの属性を追加したゲームを作成したい
- 3D グラフィックの作成について学びたい

ゲームに実現したいもの

- ノイズを使用した地形生成システム
- シーンで自由に移動できるカメラシステム
- ミニマップやボタンなどの GUI システム
- 修復用のブロックをドラッグアンドドロップできるシステム
- 地形が破壊されたように見えるシステム
- ゲームのロジックシステム

ゲームに実現したいもの

- ノイズを使用した地形生成システム [✓]
- シーンで自由に移動できるカメラシステム [✓]
- ミニマップやボタンなどの GUI システム [△]
- 修復用のブロックをドラッグアンドドロップできるシステム [×]
- 地形が破壊されたように見えるシステム [×]
- ゲームのロジックシステム [×]

ゲームについて

移動



ブロック追加



その他



• 移動するキー

- W : 前に移動
- A : 左に曲がる
- S : 後ろに移動
- D : 右に曲がる
- ▲ : 上に上る
- ▼ : 下に下がる
- ◀ : 上に傾く
- ▶ : 下に傾く

• その他のキー

- J, K, L : ブロックを追加する
- N : ワイヤースケルトンモード
- ESC : ゲームを閉じる
- C : 新しい地形を作成する

地形生成の最適化

各点では

- 5 つ以上の float が必要
- $5 * 32 \text{ bits} = 160 \text{ bits}$
- 20 bytes
- GPU に転送する時に負担や時間がかかる



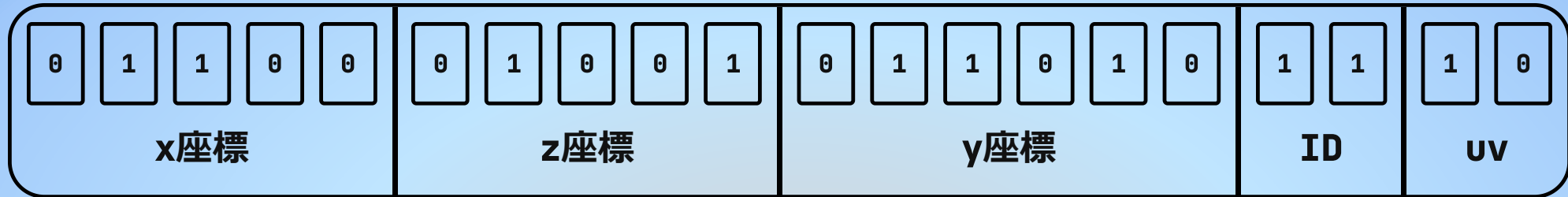
pos (13.0, 4.0, 5.0)

uv (0.0, 0.0)



431180

地形生成の最適化



=402,862

- $x = (01100)_2 = 12$
- $z = (01001)_2 = 9$
- $y = (011010)_2 = 26$
- $ID = (11)_2 = 3$
- $UV = (10)_2 = 2$

要素を分解のやり方

- $x = (num \gg 15) \& 0x1F$ (11111)
- $z = (num \gg 10) \& 0x1F$ (11111)
- $y = (num \gg 4) \& 0x3F$ (111111)
- $id = (num \gg 2) \& 0x2$ (11)
- $uv = num \& 0x2$ (11)

アイデア参照 : TanTanDev, Hopson97

https://github.com/TanTanDev/first_voxel_engine/blob/main/src/voxel_tools/mesh_builder.rs

https://github.com/Hopson97/open-builder/blob/master/shaders/chunk_vertex.glsl

地形生成の最適化

その次にチャンクを生成する時に

- 処理を複数のスレッドに分担する

両方の最適化により（実行から設定終了までの時間）

- 前 - 2.79秒
- 現在 - 1.1 ± 0.3秒
- 2.5倍ぐらい速い

ソースコード

- https://github.com/TodayNightt/environment_restoration.git

実行

```
# Change to the directory  
cd environment_restoration
```

```
# Windows  
# Cmd  
gradlew run
```

```
# Powershell  
./gradlew run
```

```
# MacOS or Linux  
chmod +x ./gradlew  
./gradlew run
```

反省点

- 予想以上に時間がかかってしまい、完成させることができなかった
- ゼロからゲームエンジンを作成するので、授業でやっていないことも学ぶ必要がありとても困難だった