

## 05 景観ジェネレータの製作

猪ノ口己太，鎌田祥

指導教員 石舘勝好

### 1. はじめに

IT の進歩により子供たちの育ち方が変わり始めている．TV やタブレット等の発展と同時に子供たちがモノを手にとって考える機会が減っていると思ったため，自分達の幼少期の頃より「創造力」が衰退していると感じた．そのような子供たちをターゲットとし，「考えて作る」ことの楽しさを伝えるため，キューブを使用した景観ジェネレータを製作した．

### 2. 研究概要

#### 2.1 仕様

建物や道路，木などを表す 15 パターンのキューブを 16 マスのスタンドに置くことで PC の画面に様々な街を表現できる．キューブには磁石が装着されていて，スタンドに埋め込んである磁気センサーが Arduino で読み取り，ネットワーク経由で Unity 側にキューブの情報を送信する仕組みである．

#### 2.2 完成図と動作環境

磁気センサーの値をネットワーク経由で送るため，Wi-Fi 機能を搭載した ESP32 を用いる．Arduino に装着された ESP32 と Raspberry Pi を通信し，Unity が一定の周期で値を受信する．

完成図を図 1 に，動作環境を表 1 に示す．



図 1 完成図

表 1 動作環境

|            |  |
|------------|--|
| OS         | Windows 10   |
| ソフトウェア     | Arduino IDE ・ Blender 2.93.4   |
|            | Unity 2020.3 ・ Tinkercad   |
| ハードウェア     | Arduino Mega 2560 R3 ( 2 台 )   |
|            | AirLift Shield – ESP32 ( 2 台 )   |
|            | Raspberry Pi 3 Model B   |
|            | ホール IC ( 磁気センサー )  |
|            | S-5716ACDL0-M3T1U ( 64 個 )   |
|            | 磁石 ( 256 個 )   |
| 3D プリンタで製作 | キューブ ( 180 個 )   |
|            | スタンド ( 4x4 , 16 マス )   |
| 外部デバイス     | テンキー ・ マウス   |
| オブジェクト     | 木 ・ 車 ・ 家 ・ 水 ・ 道路 ・ 草<br>ビル ・ 自動販売機 ・ 土 ・ 店<br>砂 ・ 公園 ・ 子供公園 ・ 花道 ・ 橋 |

### 3. ハードウェア設計・製作

基盤に磁気センサーを取り付け，その上に対応するスタンドを，Tinkercad を用いて設計し，3D プリンタで製作した．

デジタル入力が 64 個必要なので，ピンが 53 個の Arduino を 2 台使用した．

製作した収納スペースに，Arduino 2 台，Raspberry Pi 1 台を格納することで，見た目をよりコンパクトで美しい仕上げにした(図 2，図 3)．



図2 Arduino 2 台と Raspberry Pi の収納スペース



図3 スタンド(手前)と収納スペース(奥)

またキューブにはシールを貼り付け、どのキューブが何を表しているのか判別できるようにした(図4)。



図4 シールを貼り付けたキューブ(木)

#### 4. ネットワーク接続

Raspberry Pi をアクセスポイント化し、無線通信で Arduino から Unity にデータを送信する。なお、Raspberry Pi と Arduino 2 台を接続する。

##### 4.1 データ送信部(Arduino)

Arduino が Web サーバとして動作し、Unity からの定期的なリクエストに対して、キューブの位置と種類の情報を、Arduino 1 台につき 8 桁の 16 進数としてレスポンスを返す(図5)。

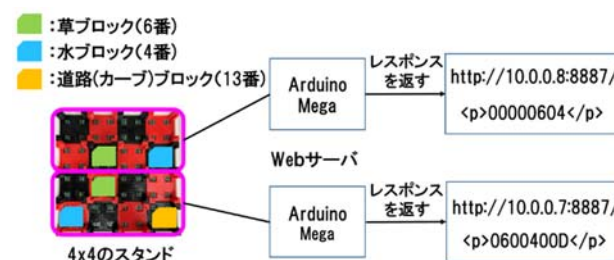


図5 データフロー図(Arduino)

##### 4.2 データ受信部(Unity)

UnityWebRequest クラスを使用して、GET で Web サーバに要求する。受け取るキューブの情報はバイナリデータなので、整数値に変換する(図6)。

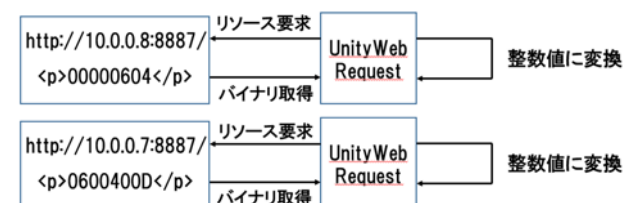


図6 データフロー図(Unity)

#### 5. ソフトウェア設計(Unity)

エディタモード(街を作る画面)と、ビューモード(街を見る画面)の2つの画面を作成した。なお、操作は全てテンキーとマウスで行う。

##### 5.1 エディタモード

Unity 上で表現できる街の大きさは理論上無限であるが、スタンドで作成できるのは 4x4 なので、カーソルを移動することで、街の作る範囲を指定する(図7, 図8)。キューブをスタンドに置くと、キューブの位置や種類によって、家や橋や水などが現れる。

エディタモードの操作方法を表2に示す。

表2 操作方法(エディタモード)

|            |              |
|------------|--------------|
| 6,5,3,9 キー | カーソルを前後左右に移動 |
| Enter キー   | 街を配置         |
| 4 キー       | ビューモードに切替    |
| +キー        | ズームイン/ズームアウト |
| クリアボタン     | 配置した街を全削除    |



図7 街を配置する様子(ズームイン時)



図8 街を配置する様子(ズームアウト時)

またズームイン中、カーソル上のオブジェクトをクリックすることで、オブジェクトの向きや、一部のオブジェクトの見た目を変えることができる(車・家・道路・ビル・店・砂)。また、図9の右下にある矢印は、現れたオブジェクトからどの方向に回転したかを表す(初期は下矢印)。なお、矢印は実際に画面上に表示される(図9)。

カスタマイズの操作方法を表3に示す。

表3 操作方法(カスタマイズ)

|            |               |
|------------|---------------|
| 6,5,3,9 キー | オブジェクトを回転     |
| 2,8 キー     | オブジェクトの見た目を変更 |



図9 オブジェクトをカスタマイズする様子

## 5.2 ビューモード

FPS(一人称)視点で、街を自由に見回ることができる(図10)。誤って街の範囲外へ出てしまわないように、透明な壁を配置している。また太陽を動かすことで一日を表現している。夜になっても外灯を点灯させることで、明るい街を景観できる(図11)。

ビューモードの操作方法を表4に示す。

表4 操作方法(ビューモード)

|            |             |
|------------|-------------|
| 6,5,3,9 キー | 前後左右を移動     |
| .キー        | ジャンプ        |
| /,*キー      | 移動速度を3段階で変更 |
| マウスのドラッグ   | 視点移動        |
| 4 キー       | エディタモードに切替  |



図10 街を見る様子(昼)



図11 街を見る様子(夜)



## 6. システムの評価

システム全体としては、スタンドに置いてあるキューブの位置や種類をその通りに画面上に反映させることができた(図12、図13)。

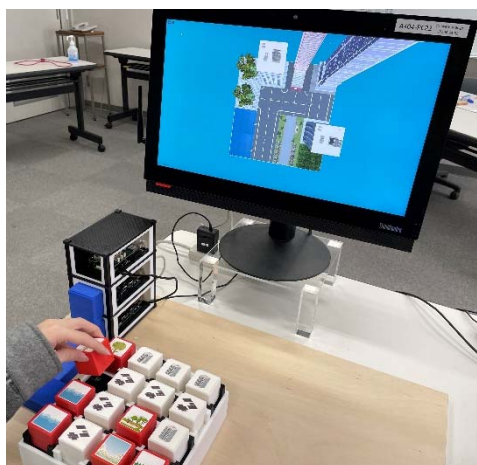


図12 システムの全体像(エディタモード)

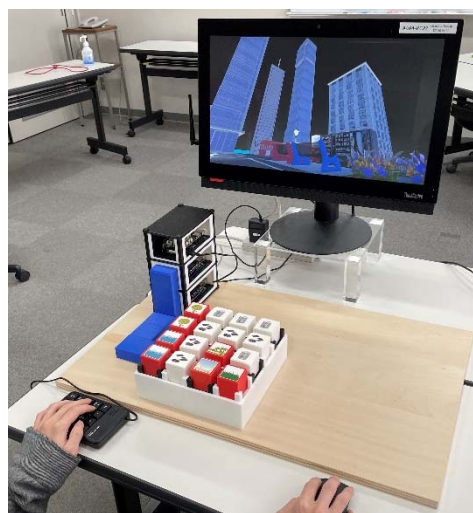


図13 システムの全体像(ビューモード)

### 6.1 ハード側

- ・接触不良が起きないように、はんだ付けに改良を加えた。
- ・ケガしないようにキューブの角を丸めた。
- ・Unity上で生成されるオブジェクトをイメージしやすいシールにすることができなかった。
- ・マウスとテンキーだと操作に難点があるので、子供でも扱いやすい専用のコントローラを製作する必要があった。

### 6.2 ソフト側

- ・1秒周期でオブジェクトを画面上に反映させることで、ほぼリアルタイムでのオブジェクト生成に近づけることができた。
- ・街を無限に配置できる機能を実装できた。
- ・オブジェクトの向きや見た目をカスタマイズできる機能を実装できた。
- ・作った街を自由に見回れる機能を実装できた。
- ・家4つの隣接で1つの豪邸になるなど、複数のオブジェクトが1つのオブジェクトになるような処理を実装できなかった。
- ・車に動きをつけるなど、より現実味のある街を作れるようにしたかった。

## 7. おわりに

私たちが製作した景観ジェネレータは、子供の「創造力」が衰退していると考え、そのような子供たちに「考えて作る」ことの楽しさを伝えるというコンセプトで取り組んできた。

自由度を高めるために、カスタマイズ機能などの追加や、安全に利用できるようにキューブの形を丸めた。

結果として、当初の予定より、安全で自由度の高い景観ジェネレータが完成した。

## 8. 参考文献

- 1) Unity スクリプトリファレンス  
<https://docs.unity3d.com/ja/2020.3/ScriptReference/>
- 2) Arduino WiFi  
<https://learn.adafruit.com/adafruit-airlift-shield-esp32-wifi-co-processor/arduino>
- 3) UnityWebRequest の使い方【Unity】  
<https://kan-kikuchi.hatenablog.com/entry/UnityWebRequest>