

令和4年度 卒業研究

Blender を用いた函館の街並み再現
及び、車の運転シミュレータの開発

Development of a simulator to reproduce the city of Hakodate
in Blender and to simulate driving a car

函館工業高等専門学校

生産システム工学科 機械コース

閑田 紀子

指導教員 東海林 智也

目次

第 1 章 序章	3
1.1 概要 (Abstract)	3
1.2 はじめに	3
1.3 研究の流れ	3
第 2 章 使用技術	5
2.1 開発環境	5
2.2 Blender	5
2.3 Unity	5
第 3 章 研究詳細	6
3.1 函館の街並みの 3D モデル化	6
3.1.1 Blender のインストール	6
3.1.2 アドオンのインストール	6
3.1.3 街並みと建物の作成	9
3.1.4 マテリアルの設定	12
3.1.5 FBX ファイルへのエクスポート	14
3.2 車の運転シミュレータの開発	16
第 4 章 まとめ	18
4.1 成果	18
4.2 今後の課題	19
4.3 おわりに	19
謝辞	20
参考文献	20

1 序章

1.1 概要 (Abstract)

This study is a car driving simulator that aims to improve students' programming skills and logical thinking. 3D modeling software Blender [1] and game development platform Unity were used to create a cityscape of Hakodate using Blender. The goal of the project was to create a cityscape of Hakodate using Blender and create buildings, place obstacles in the cityscape, and have students solve the problems using Python programming. Currently, the car is programmed to drive on the background created with Blender.

Key words: blender, Unity, Computational Thinking, Hakodate

1.2 はじめに

本研究には2つ目的がある。1つ目の目的は、函館が持つ「趣のある街並みや実家のような安心感」などの魅力をアピールし、函館の観光業に貢献するために3D制作ソフト Blender [1]を使って函館の街並みを再現する3Dモデルを作成することである。

2つ目の目的は、函館の魅力をアピールしつつ学生のプログラミング能力と論理的な思考の向上を図るため、函館の街並みを背景とした車の運転シミュレータを Unity[2][3]を使って開発することである。2020年に小学校でプログラミング教育が必修化されたことでプログラミング教育市場規模はこれからも拡大すると考えられる。

なお観光やプログラミング教育だけでなく、例えば災害のシミュレーションで利用して避難方法を提案するなど、今回作成する函館の街並みの3Dモデルは様々な用途があると考えられる。

1.3 研究の流れ

まず、Blenderを使って函館の街並みの3Dモデルを作成する。なお函館全域をモデル化することは難しいので、今回は範囲を絞り西部地区と旧函館区公会堂とその周辺の建物に限って3Dモデル化することにした。次にUnityで函館の街並みを背景に車がプログラムに従って走行するシミュレータを作成する。このシミュレータの完成イメージを図1に示す。



図1 シミュレータ完成イメージ

2 使用技術

2.1 開発環境

開発環境を以下に示す。

OS : Windows10
グラフィックボード : Intel(R) HD Graphics 515
ライブラリ : Python

2.2 Blender

Blender ver.3.2.1 を使用する。

2.3 Unity

Unity ver.2021.3.11f1 を使用する。

3 研究詳細

3.1 函館の街並みの 3D モデル化

3.1.1 Blender のインストール

Blender はオープンソースソフトウェアであり、公式サイトからダウンロードすれば誰でも簡単に利用できる[1]。Blender は 3DCG アニメーションを作るための機能が全て揃っているソフトで実際に 3DCG アニメーションのプロが使用している。本研究では、3D モデリングの機能だけを使ったがレンダリング、VFX、アニメーションとリギング、3D ビューポートで直接描画することができるので 2D 画像を描くこともできる。

今回は以下の手順で「Blender ver. 3.2.1」をインストールした。

1. 公式ウェブサイトを開く
2. 「Download Blender」をクリックする
3. 「Windows Installer」が選ばれている事を確認する
4. 「Download Blender 3.2.1」をクリックする
5. ダウンロードが開始されることを確認する
6. エクスプローラよりダウンロードしたファイルをダブルクリックで実行する
7. Welcome to the…のページで「Next」をクリックする
8. ライセンス条項の確認画面で「Next」をクリックする
9. カスタムセットアップの設定画面で「Next」をクリックする
10. インストール開始の画面で「Install」をクリックする
11. インストールが終了し、確認画面が出てきたら「Finish」をクリックする
12. パソコンのデスクトップから起動する

以上の手順でインストールされた Blender の起動画面を図 2 に示す。

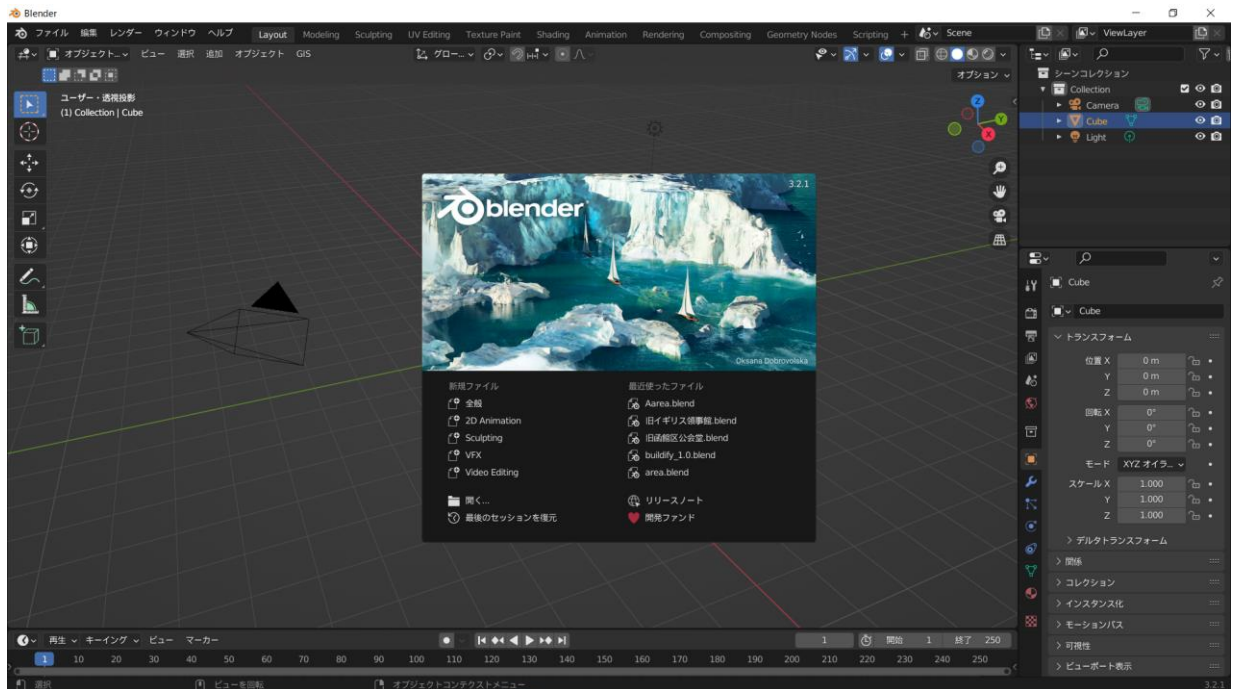


図 2 Blender 起動画面

3.1.2 アドオンのインストール

次に以下の手順でアドオンをインストールした。

1. アドオン配布サイトでインストールしたい zip ファイルをダウンロードする
2. 編集からプリファレンスを開き、7 行目にあるアドオンをクリックする
3. ウィンドウ右上にある「インストール」をクリックしてダウンロードしてきたアドオンを選択する
4. ウィンドウ右下にある「アドオンをインストール」をクリックする
5. 使用したいアドオンのチェック項目にチェックをいれ、有効化する。

以上の手順に従ってインストールしたアドオン一覧を図 3 に示す。今回はアドオンの中から「3D View: BlenderGIS」と「Mesh: Building Tools」の 2 つを使用した。

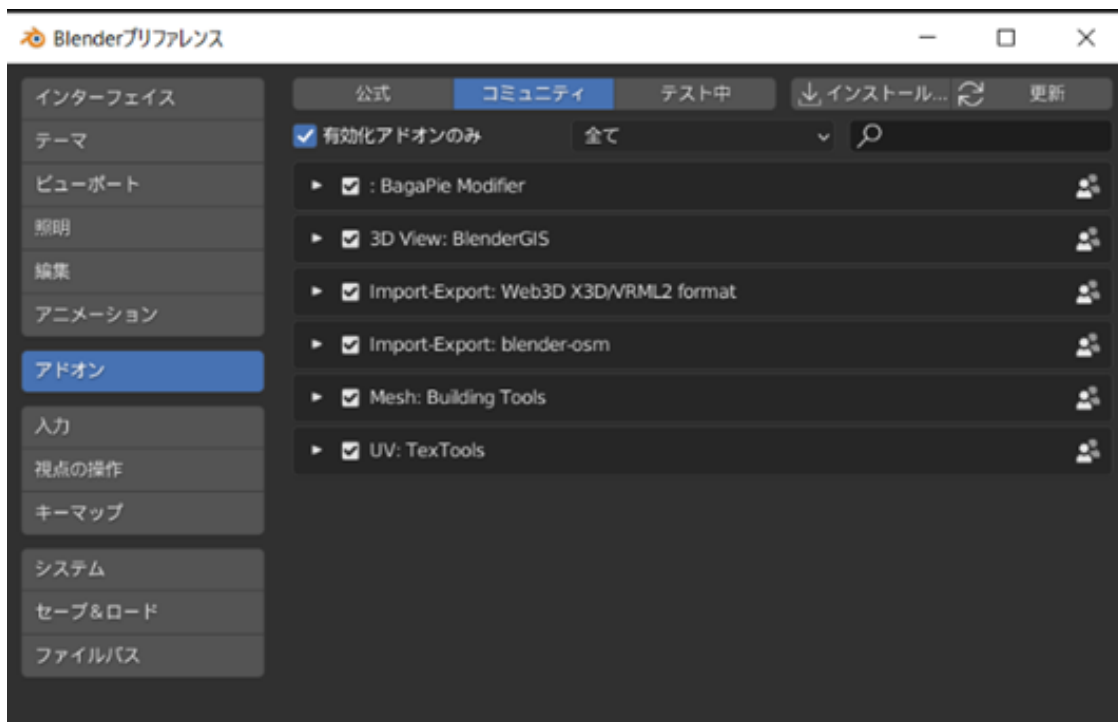


図3 アドオン一覧

3.1.3 街並みと建物の作成

次に函館の西部地区の町並みと旧函館区公会堂の 3D モデルを作成した(図 4、5)。

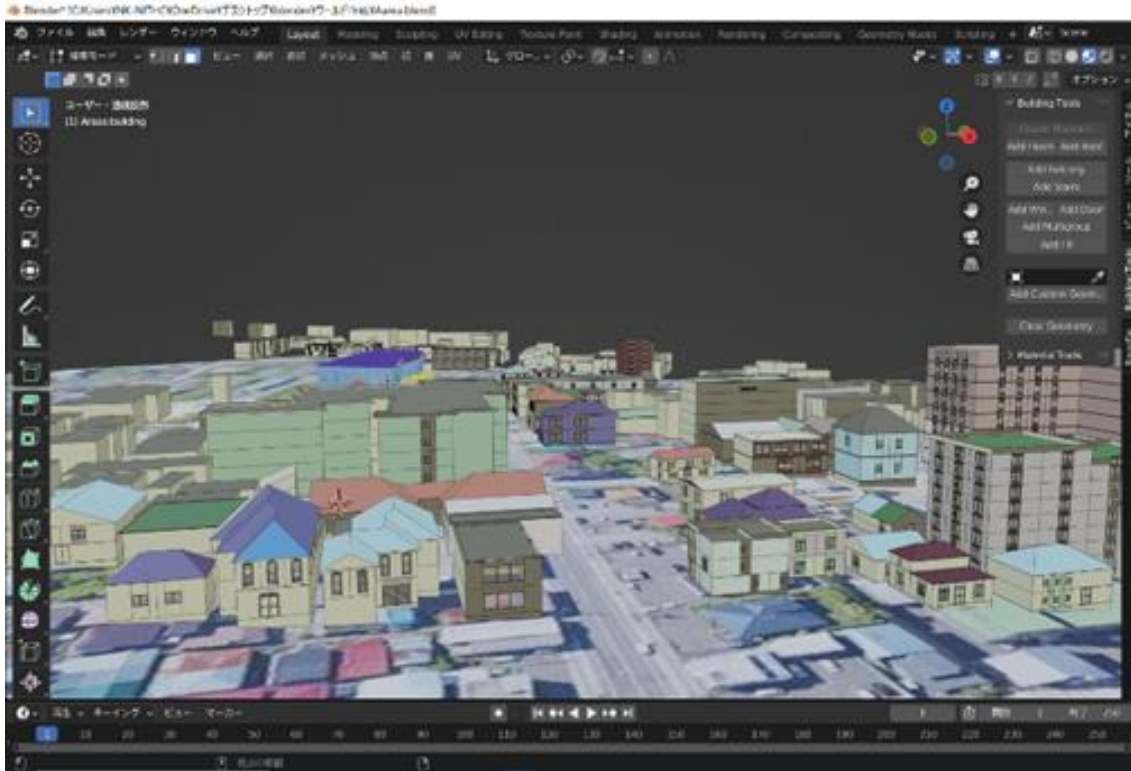


図 4 西部地区の街並み

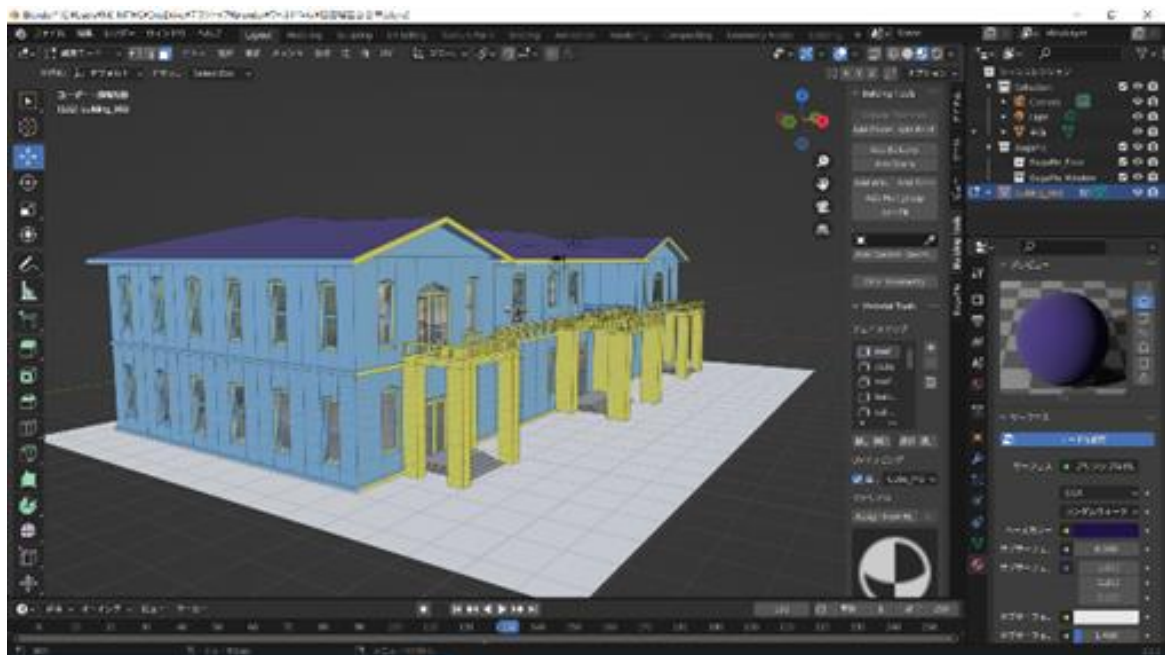


図5 旧函館区公会堂

これらは以下の手順で作成した。

1. GIS から Web geodata から Basemap をクリックする (図6)
2. ソースを Google に設定し OK をクリックする
3. 旧函館区公会堂の周辺に画面を移動し E キーを押す
4. GIS から Web geodata から Get SRTM をクリックする
5. GIS から Web geodata から Get OSM をクリックする
6. Ways タブで項目を全選択し、「Elevation from object」にチェックを入れ OK をクリックする
7. 選択した地図の範囲で建物が作成されたことを確認する (図7)。
8. N キーを押してサイドバーを展開し、「Building Tools」タブをクリックする
9. 実際の建物を参考にタブで屋根や窓、バルコニーを作成する (図8)
10. Google Earth を参考に屋根と窓を作り、マテリアル付けを行う

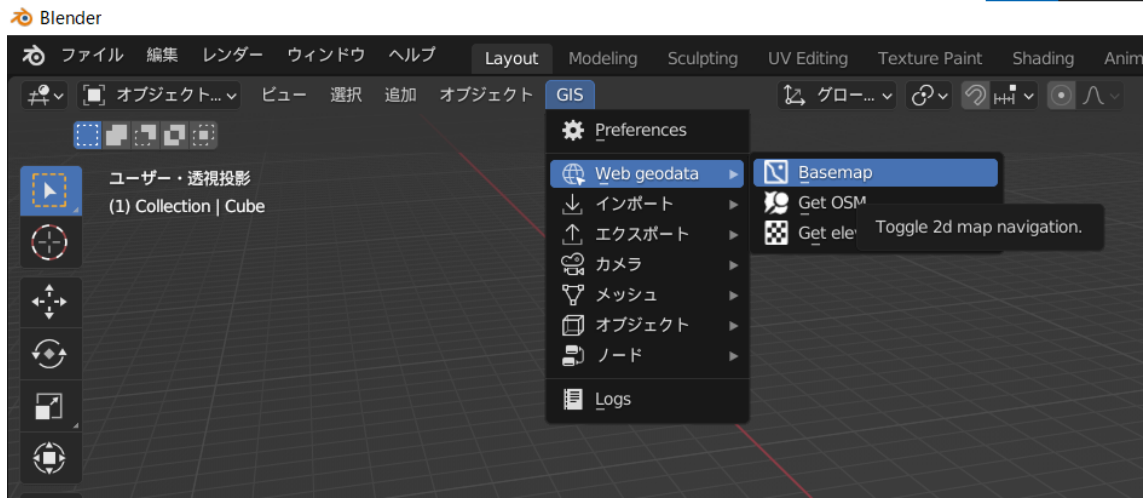


図 6 Basemap

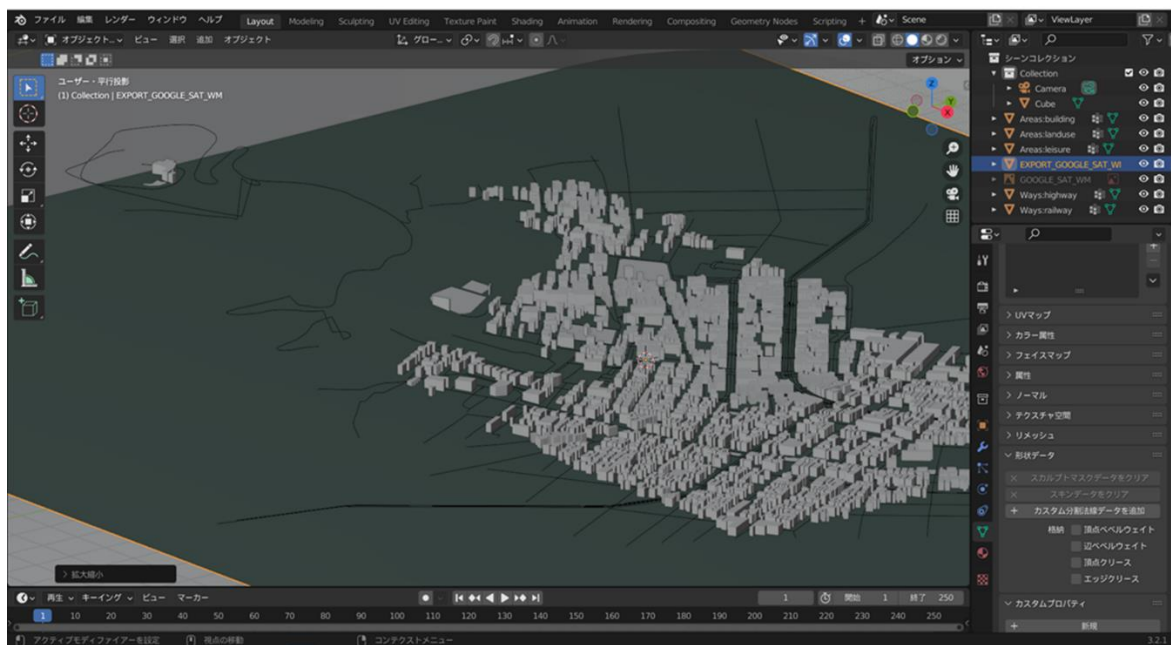


図 7 建物概形

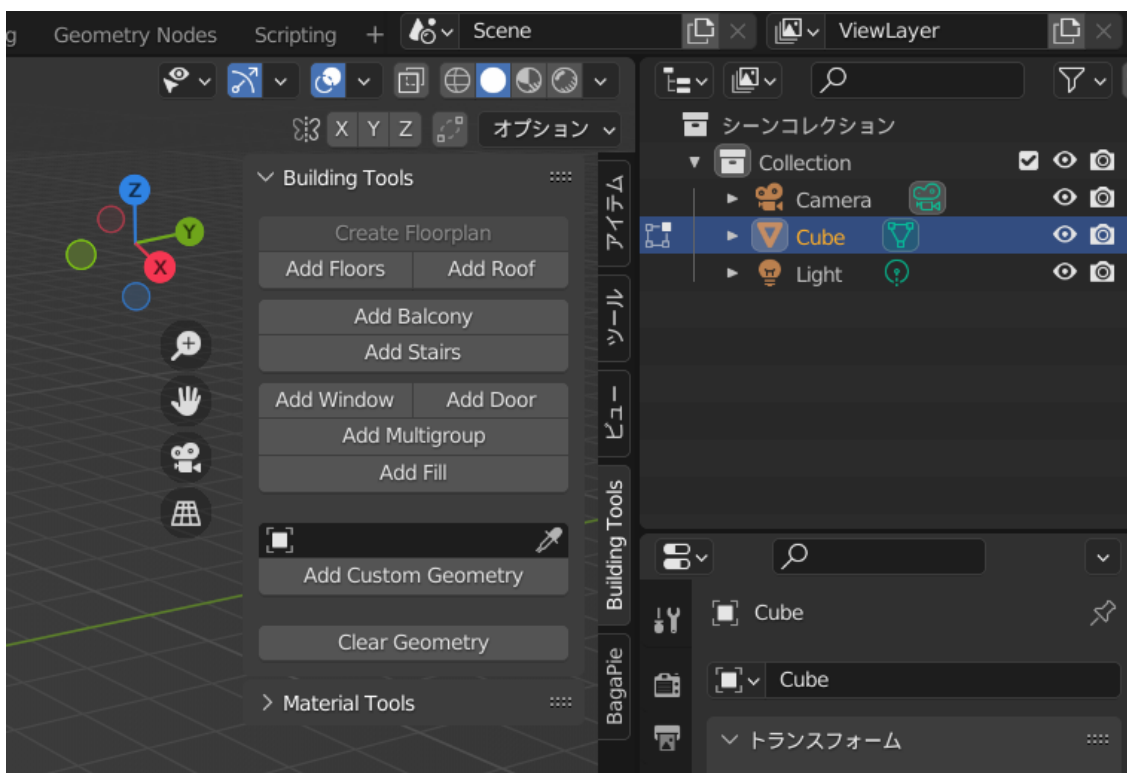


図8 「Building Tools」のタブ

3.1.4 マテリアル設定

今回設定したマテリアルの一覧を図9に示す。

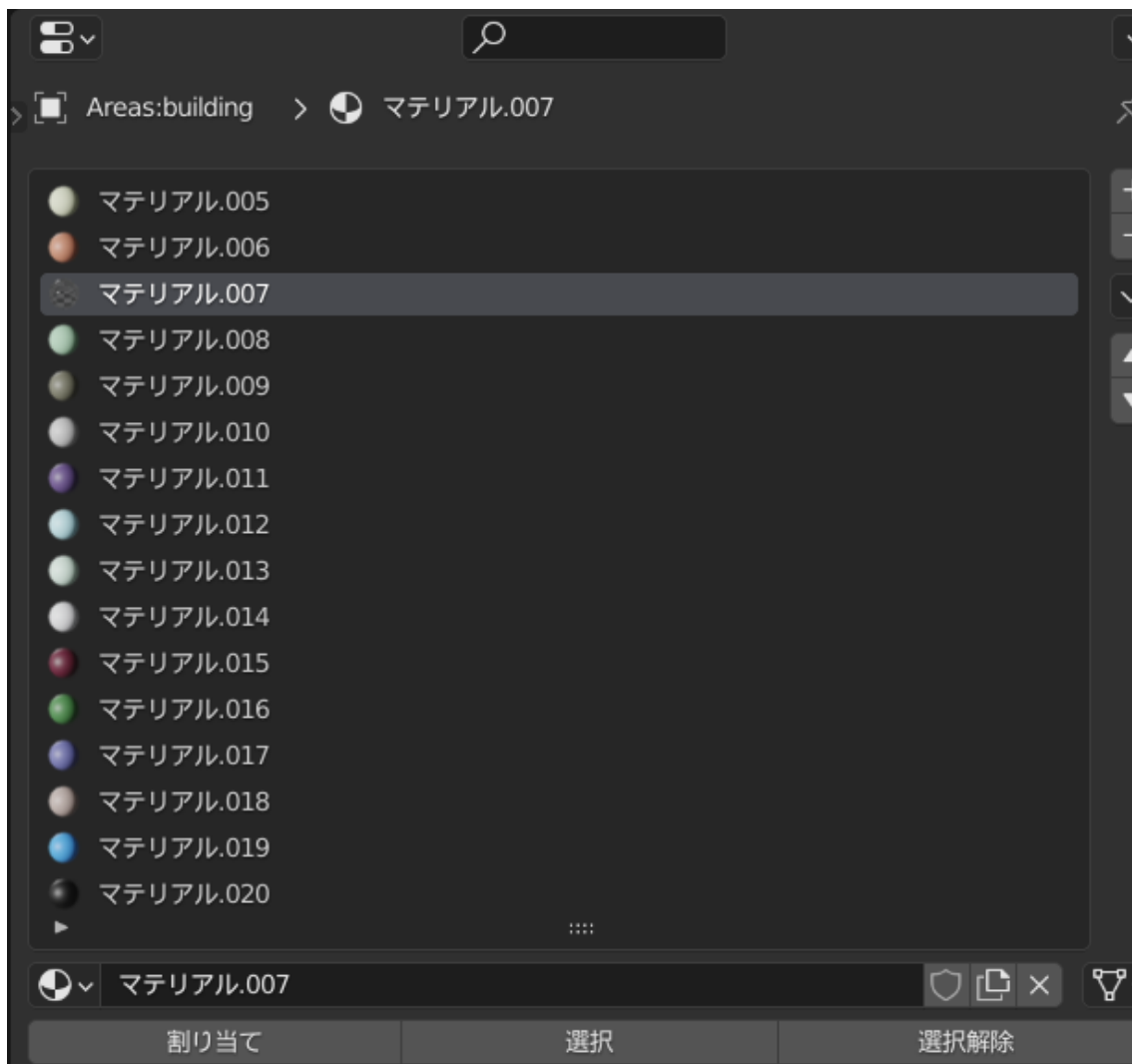


図9 マテリアル一覧

マテリアルにはいくつか種類があるが、例としてガラスマテリアルの設定手順を以下に示す。

1. マテリアル設定内のブレンドモードを「アルファブレンド」に設定する
2. 粗さを「0」に設定する
3. 伝播を「0」に設定する
4. IORを「1.57」に設定する
5. アルファを「0.6」に設定する

3.1.5 FBX ファイルへのエクスポート

表 1 のように Blender の座標系と Unity の座標系は上方向と奥方向が異なっていて右方向も反転している。よって Blender で作成した 3D モデルはそのままでは Unity で使用できない。

表 1 座標系

	右方向	上方向	奥方向
Blender	+ X	+ Z	+ Y
Unity	- X	+ Y	+ Z

そこで以下の手順で 3D モデルを FBX ファイルにエクスポートしてから Unity で利用することにした。

1. A キーでオブジェクトを全選択する
2. ファイルからエクスポートをクリックし「FBX(.fbx)」を選択する
3. パスモードをコピーに変更し、「選択したオブジェクト」にチェックする
4. オブジェクトタイプを全選択し、「スケールを適用」を「FBX 単位スケール」に変更する
5. トランスフォームを適用にチェックを入れる
6. 「FBX をエクスポート」をクリックする

以上の設定画面を図 10 に示す。

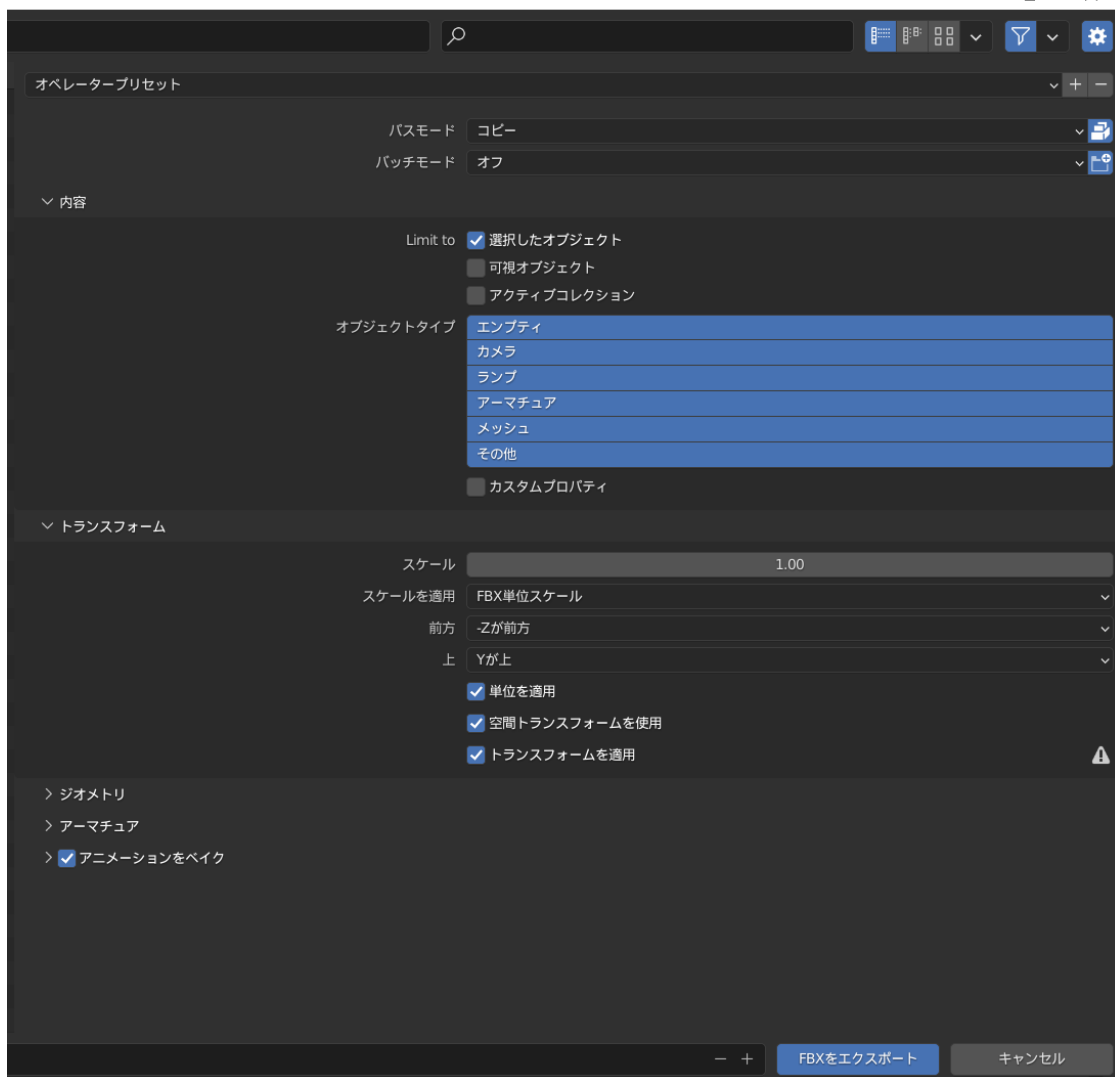


図 10 FBX 変換設定画面

3.2 車の運転シミュレータの作成

次に Unity を用いて車の運転シミュレータを作成した。作成中の Unity 操作画面を図 11 に示す。

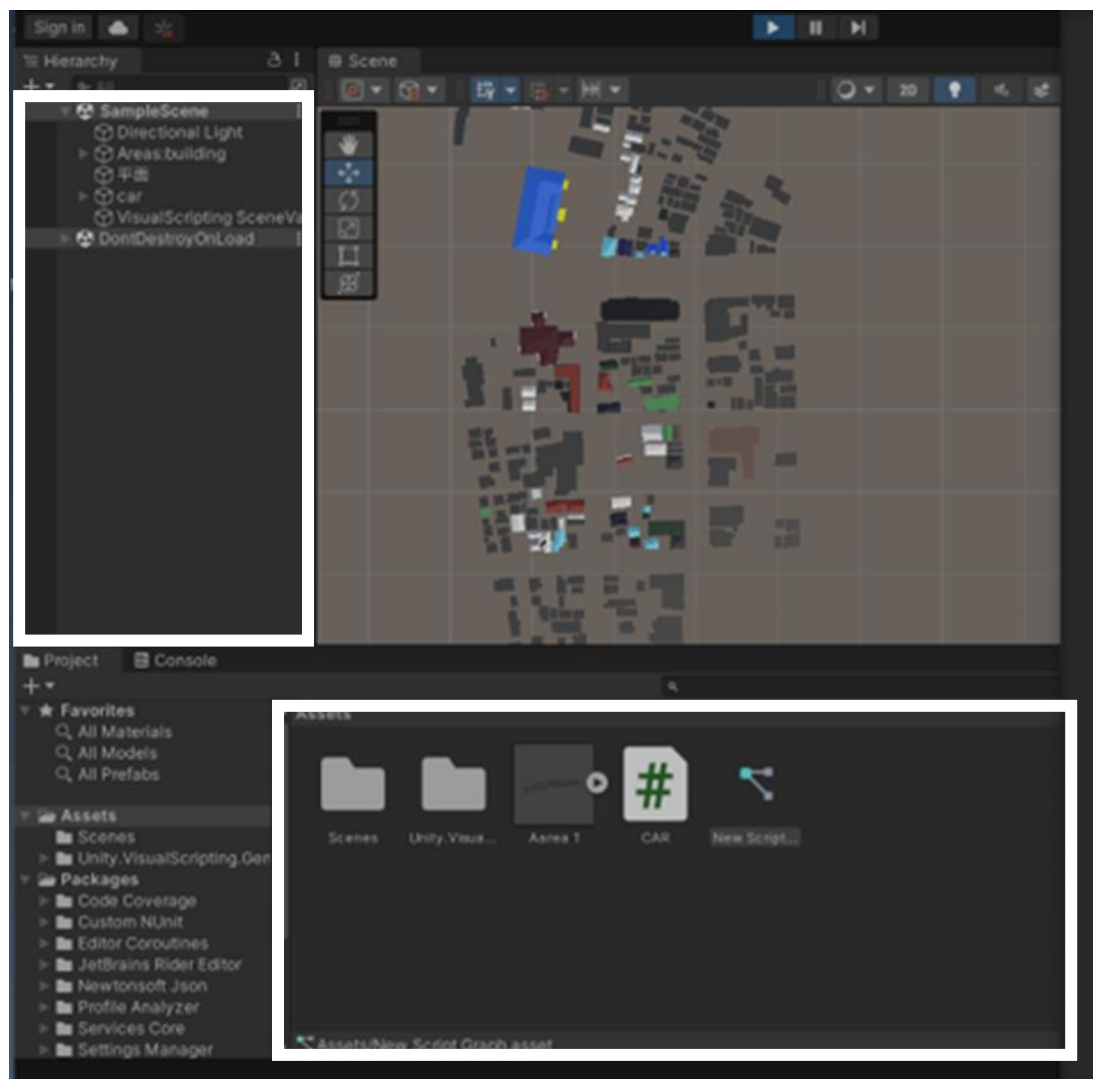


図 11 Unity 操作画面

シミュレータの作成手順を以下に示す。なお図 11 の左側の枠線をヒエラルキー、下側の枠線内を Assets ウィンドウと言う。

1. New project でプロジェクトを新規作成する
2. Assets ウィンドウに 3.1.5 で作成した FBX ファイルをドラック&ドロップする
3. ヒエラルキー上で右クリックし「Create Empty」をクリックする
4. インスペクター上にある「Add Component」から「Rigidbody」を選択し Mass を「1000」に設定する
5. 「Add Component」から「Script Machine」をクリックする
6. Assets ウィンドウ上で右クリックし、Create→Visual scripting→Script Graph を選択する
7. 6. で作成した「Script Graph」をダブルクリックする
8. 画面上で右クリックし「On Start」を追加し三角形マークを近くにドラック&ドロップする
9. Rigidbody の「Set Velocity」を追加し Z の値に 10 を代入する(図 12)
10. Script Graph を Script Machine に挿入する
11. ヒエラルキー上の「VisualScripting SceneVariables」を「Game Object」の下へ移動する

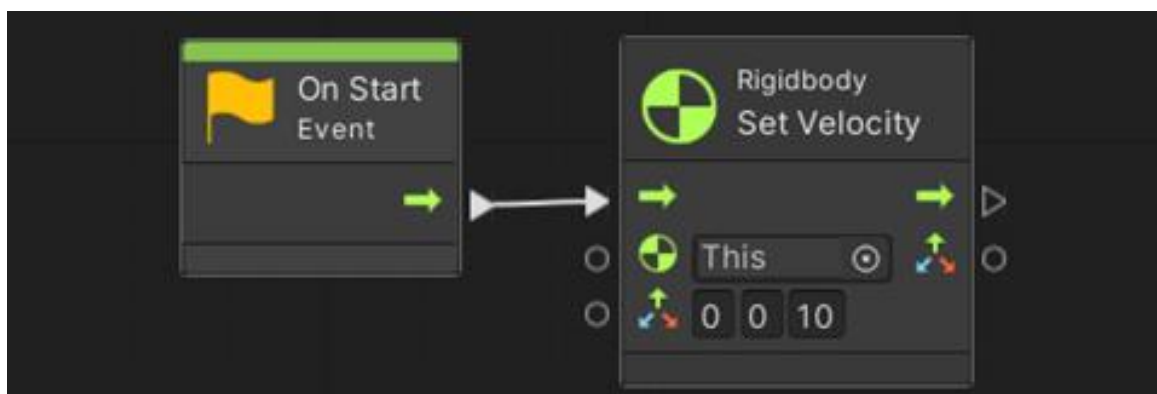


図 12 Script Graph

なお図 12 に示したスクリプトは車が単に前に進むだけのプログラムとなっている。そこでこのシミュレータを学生に配布したあと、車が建物にぶつからない様にスクリプトの続きを学生に作成してもらふことにする。その際に試行錯誤を繰り返すことで学生のプログラミング能力向上を期待できる。

4 まとめ

4.1 成果

本研究で作成した車の運転シミュレータの現状を図 13, 14 に示す。

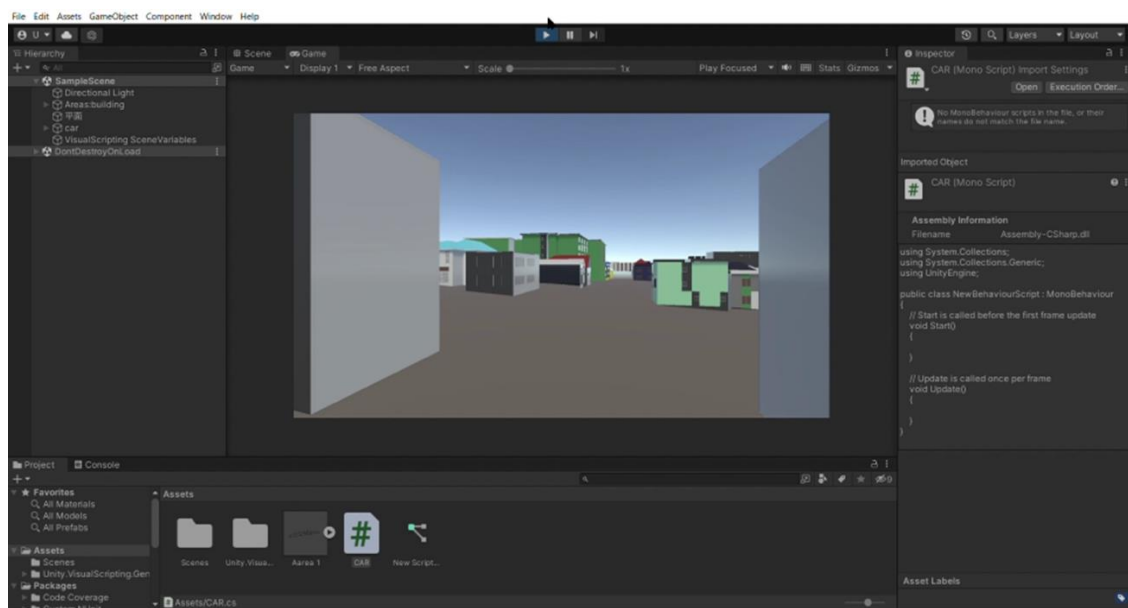


図 13 現状①

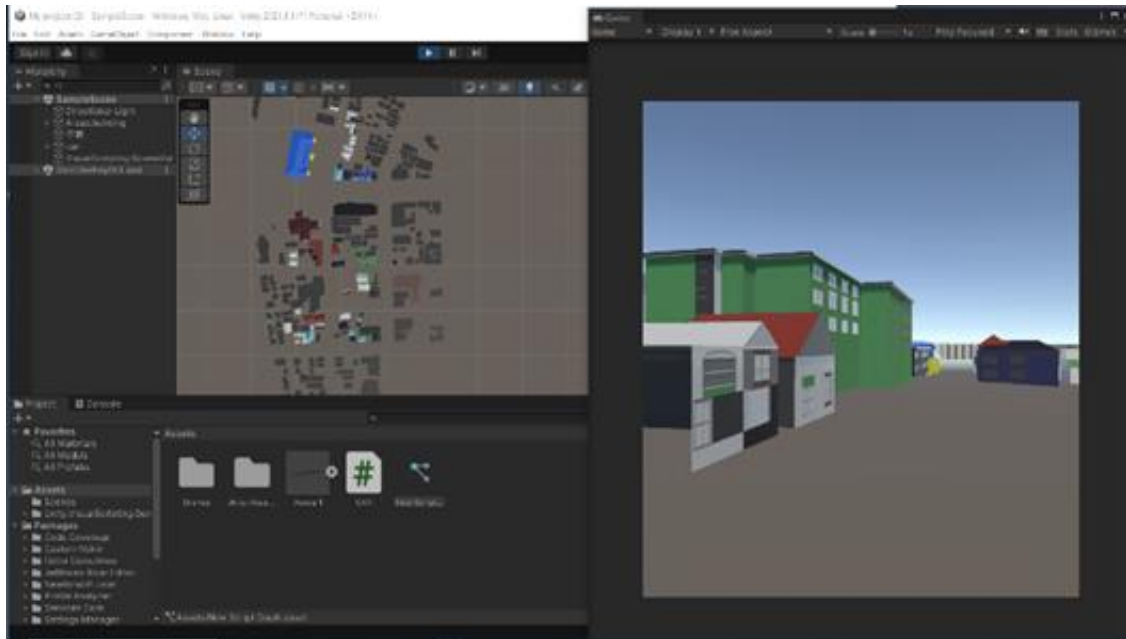


図 14 現状②

4.2 今後の課題

函館の街並みを背景に車が走行するところまで完成したが、信号機、標識や障害物の設置がこれからの課題である。また、今回はマテリアルで色付けをしたが、よりリアルな街並みを再現するテクスチャを作成したいと考えている。なお完成次第、オンラインで無料配信をする。

4.3 おわりに

現在、持続可能な社会がうたわれるなかで、アバターと呼ばれる自分の分身を介して他者とコミュニケーションをとったり、買い物したりできる「仮想空間」メタバースが注目を集めている。メタバースはバーチャル会議やアイドルのコンサート活動など発想次第で自由な空間を構築できるため可能性が無限大である。私は、そんなメタバースの発展に Blender が貢献できると考えている。本研究にとどまらずこれからも Blender で自分の可能性を見出していきたいと考える。

謝辞

本研究を進めるにあたり、ご指導くださいました担当教員の東海林智也准教授、また、同研究室のメンバーに感謝申し上げます。

参考文献

- [1] Blender ホームページ: <https://www.blender.org/>
- [2] Unity ホームページ: <https://unity.com/ja>
- [3] 東海林智也: アクティビティ: Unity 入門,
<https://tmytokai.github.io/open-ed/activity/unity-basic/>