

令和4年度卒業研究報告書

A-Frame を用いた学校見学サイトの作成

情報技術科 佐々木 悠真

佐々木 蓮斗

菅原 匠真

指導教員 安倍 春菜

目次

第 1 章	はじめに	3
1.1	経緯と現在の VR 学校見学	3
1.2	目的	3
第 2 章	研究概要	4
2.1	開発環境	4
2.2	使用技術	4
2.2.1	HTML	4
2.2.2	CSS	4
2.2.3	SCSS	5
2.2.4	JavaScript	5
2.2.5	A-Frame	5
2.3	撮影について	6
第 3 章	サイトの設計について	7
3.1	ページ一覧	7
3.2	サイトのデザイン	8
3.3	ファーストビュー	9
3.4	モーダルウィンドウ	10
3.5	紹介のページ	11
3.6	ヘッダー	11
第 4 章	コーディングについて	12
4.1	トップページ	12
4.1	メインページ	15

4.4 読み込みの高速化.....	24
4.4.1 JavaScript と CSS をインラインで記述	26
4.4.2 JavaScript と CSS を簡略化しコード量を削減.....	26
4.4.3 JavaScript と CSS ファイルの圧縮	28
4.5 レスポンシブ対応.....	29
4.6 ジャイロ機能制限.....	32
4.7 CSV ファイルでの管理.....	33
4.8 A-Frame の日本語対応.....	34
第 5 章 完成品.....	52
第 6 章 おわりに	55
6.1 評価.....	55
6.1.1 良かった点.....	55
6.1.2 悪かった点.....	55
6.1.3 今後について	56

第1章 はじめに

1.1 経緯と現在の VR 学校見学

現在,当校の公式サイトでは VR 学校見学のサイトが公開されている.しかし,サイト内で使用されている画像の雰囲気が暗いという問題点があった.さらに,授業の中で一年生を対象に行ったアンケート調査の結果,入学前に授業内容をもっと詳しく知りたかったという意見が多かった.

そこで私たちは,今公開されているサイトをリニューアルしようと考えた.

1.2 目的

新しく写真を撮影することでサイト全体の雰囲気を明るくし,授業内容を知ることができる要素を追加することで,ターゲットに設定していた本校に興味を持っていて入学予定の中高生に当校のことを知ってもらい,使用感と見栄えの良いサイトを作ることを目標として作成した.

第2章 研究概要

2.1 開発環境

表1 開発環境

開発環境	VScode,
使用言語・使用ライブラリ	HTML,CSS,SCSS,JavaScript,A-Frame
使用ソフト	Blender
対応ブラウザ	Chrome,Safari
使用機材	RICOH THETA

2.2 使用技術

2.2.1 HTML

HTMLとはHyper Text Markup Language（ハイパーテキスト・マークアップ・ランゲージ）の略で,WEB ページ作成のために開発されたマークアップ言語である.HTMLはHyperText と Markup の2つの特徴を持つ.HyperTextとはハイパーリンクを埋め込むことができるテキストで,これを利用するとWEB ページをほかの画像や動画に紐づけることができる.Markupはタグを使って見出しや強弱などの印づけをしてコンピュータに伝えることができる.

2.2.2 CSS

CSSとはCascading Style Sheet（カスケーディング・スタイル・シート）の略でHTMLと組み合わせることで,サイト内の色やサイズ,レイアウトなどのデザインを定義するための言語.

2.2.3 SCSS

SCSS とは Sass の構文の一つであり,先ほど説明した CSS を拡張して扱いやすくしたスタイルシートである.Sass とは Syntactically Awesome Stylesheets の略で,構造的にすばらしいスタイルシートという意味.CSS はコーディングを進めるにつれてより肥大化し,複雑になってしまうが SCSS はそれらを抑え,効率的に,読みやすくメンテナンスしやすいコードを書くことができる.具体的には変数を使用できたり,ネスト (入れ子構造) を使うことができる。

2.2.4 JavaScript

JavaScript とは WEB ブラウザ上で動作するプログラミング言語のことである.HTML と CSS で構築されたサイトに動きを加えたり,ユーザーの操作に応じて内容が変わる動的なコンテンツを実装することができる。

2.2.5 A-Frame

A-Frame とは WEBVR を作成することができるオープンソースのフレームワークのことである.プログラミング言語は JavaScript で,HTML のタグを記入するだけで簡単に VR 空間にオブジェクトを表示することができる.また,VIVE/Oculus Rift/Oculus Go など多くの VR ヘッドセットに対応しており,AR 開発にも使える。

2.3 撮影について

各教室の撮影には情報技術科にある 360 度カメラ「RICOH THETA」を使用した。



図1 RICOH THETA

矢巾校の撮影は 10 月中旬から開始し、水沢校は 12 月 7 日に実際に行って撮影した。

撮影の際に気を付けたことは、サイトを見た人にその教室でどんな作業をしているのか伝わるように写すことと、カメラの高さや位置を全部の写真に掛けた。撮影した部屋は全部で約 40 部屋行った。



図2 撮影した写真の例

第3章 サイトの設計について

3.1 ページ一覧

本サイトは最初に矢巾校の3D オブジェクトの画面に飛ぶ.その後水沢校か教室紹介のページに移るか選ぶことができる.ピンの置いてある校舎をクリックすることで対応した教室に移動する.そこで○×ゲームをするか,別の教室に移動することができる.

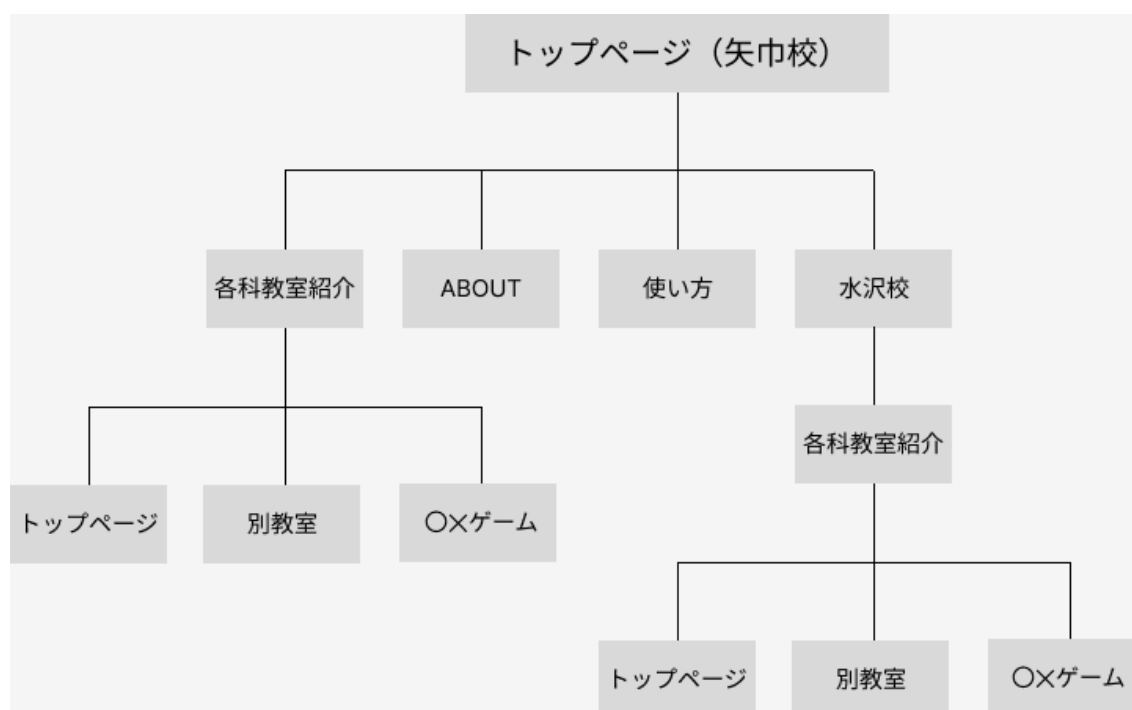


図3 ページ一覧

本サイトのユーザー側のメリットとしては,3D オブジェクトで校舎を作ることによって全体の構造をより理解してもらえることと,VR 画像で教室を紹介したり○×ゲームを実装することで,実際の雰囲気や授業内容を知ってもらえることが挙げられる.

3.2 サイトのデザイン

サイト内で使用している色に関しては公式 Web サイトのデザインガイドラインに沿ったものを使用している。



図4 デザインガイドライン

各校舎の3D オブジェクトの上に置いているピンの色も、各科のイメージカラーを使用することで区別しやすく、視覚的にも楽しめるものになっている。



図5 各科ピン

3.3 ファーストビュー

ファーストビューには,だれでも操作方法が分かるようにモーダルウィンドウを強制的に表示するようにしている.モーダルウィンドウ内には文字だけではなく,実際に動かしている動画を貼り付けることでPCの操作にあまり慣れてない人でも使いやすいように意識している.ここで使っている色もデザインガイドラインに沿った色を使用していて,サイトの雰囲気を変えずにシンプルなデザインを心掛けて作成した.



図6 ファーストビュー

3.4 モーダルウィンドウ

ヘッダーの ABOUT や使い方のページにもモーダルウィンドウを用いている.紹介ページの使い方のモーダルウィンドウはページを分けて実装している.ページの切り替えには JavaScript を用いて addClass と removeClass を使い,HTML のタグにクラスを足したり削除することで表示内容と動きを変えている.



図7 モーダルウィンドウ

3.5 紹介のページ

制作初期は各科・各教室の紹介ページからいつでも別の科の教室に移動することができるようになっていたが、校舎の3D オブジェクトをせっかく作ったのに表示する機会が一度しかないのはもったいないという意見をいただき、今いる科の教室のみ移動できるように変更した。そうすることで3D オブジェクトを見る機会が増え、利用者により詳しく学校全体の構造をより理解してもらえと考えられる。

3.6 ヘッダー

ヘッダー部分もデザインを変え、機能を少し追加した。初期は背景色が青で文字が白だったが、色を反転して背景色を白にすることで、より明るくシンプルで清潔感のあるデザインに変更した。

また、大きな画面で教室の写真を見もらうため、ナレーションの切り替えボタンもヘッダー内に埋め込んだ。



図8 ヘッダー

第4章 コーディングについて

4.1 トップページ

VR は基本カメラの軸が固定で,見渡す形になるが,このページではオブジェクトに軸をおきカメラを固定して,オブジェクトを回転して使用したかったため,このページでは `aframe-orbit-controls.min.js` というコンポーネントを使用した.

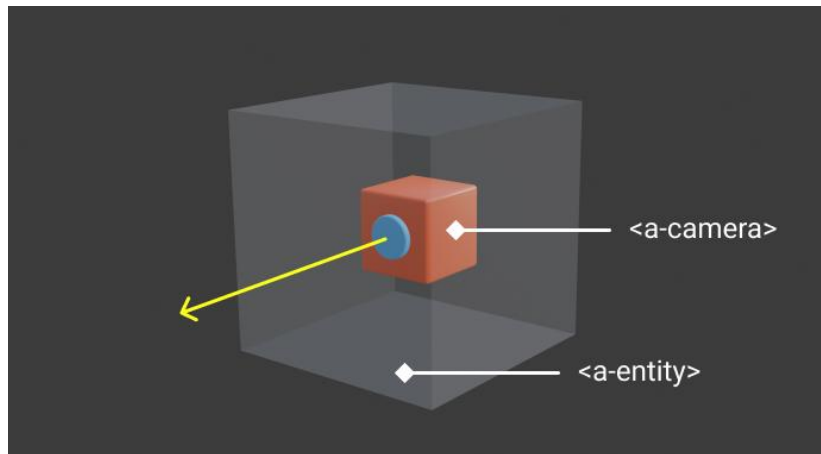


図9 初期状態の A-Frame のカメラの設計 1

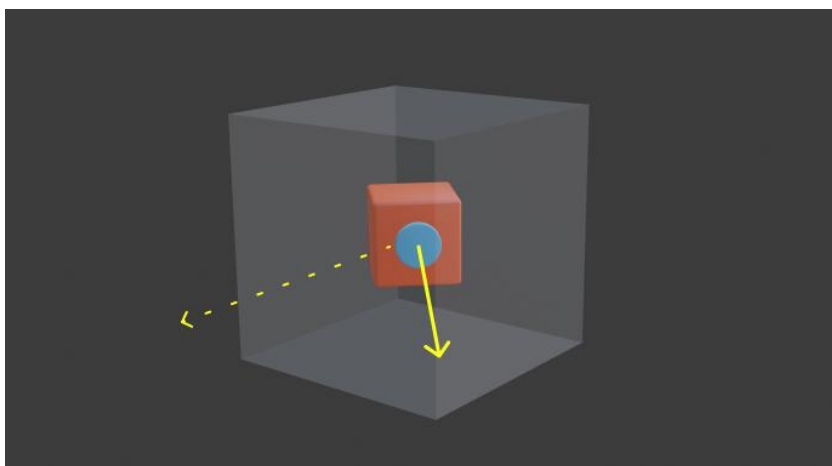


図10 初期状態の A-Frame のカメラの設計

4.1.1 カメラ設定

A-Frame.js では、マウス操作によって自分自身が移動するため、3D モデルが画面の中心からずれてしまう。そのため `aframe-orbit-controls.min.js` を使用した。

`aframe-orbit-controls.min.js` は `<a-entity>` を用いることでカメラを指定することができる。

```
<a-entity id="camera_pc" camera look-controls="enabled: false;"  
  magicWindowTrackingEnabled="false"  
  orbit-controls="target: -4 1.6 -0.5; //視点の先として固定する座標  
    maxPolarAngle:360; minDistance: 0.5; //最大角度  
    maxDistance: 360; //変えることができる最短の距離を指定  
    initialPosition: 30 10 -15" //カメラの最初の位置>  
</a-entity>
```

図 11 オブジェクトの

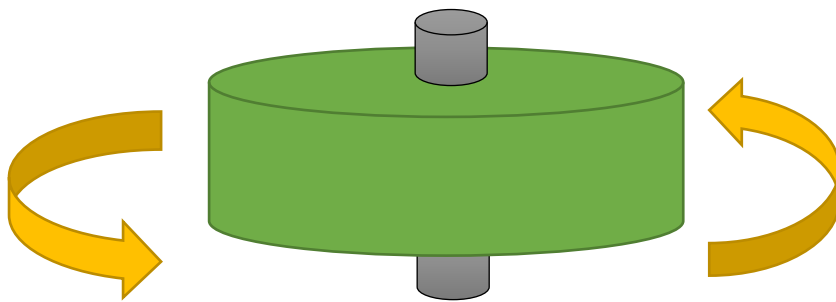


図 13 回転させたいオブジェクト

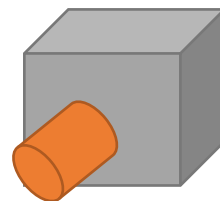


図 12 固定したカメラ

4.1.2 オブジェクト設定

以下は,情報技術科の教室オブジェクトの設定である.

<a-gltf-model>タグを用いることで,blender で作成したオブジェクトを表示することができる.

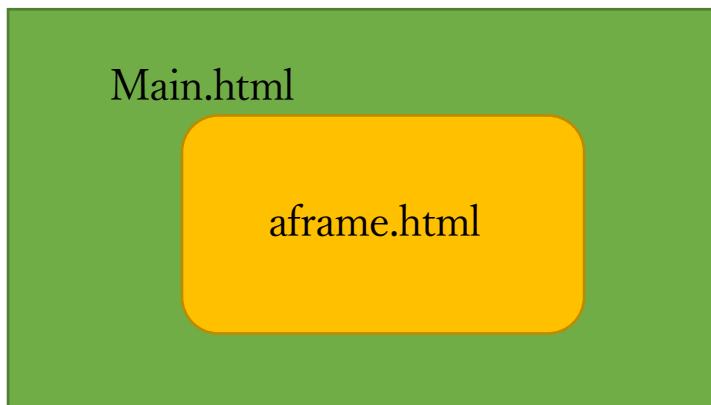
```
<a-gltf-model id="zyohou_model"
  visible="true" //オブジェクト表示 ・ 非表示 設定
  src="top_aseets/zyohou.glb"
  position="-13 0 0" //位置の指定(X Y Z 軸)
  scale="1 1 1" //大きさの指定(X Y Z 軸)
  rotation="0 0 0" //回転の指定(X Y Z 軸)
  animation__mouseenter="property: position; //ホバーのアニメーションの指定
    from: -13 0 0; //最初の位置を指定(position)
    to: -13 0.4 0; //終わりの位置を指定(position)
    dur: 600; //アニメーションの時間指定
    loop: false; //繰り返しの指定
    startEvents: mouseenter"
  animation__mouseleave="property: position;
    from: -13 0.4 0; //最初の位置を指定(position)
    to: -13 0 0; //終わりの位置を指定(position)
    dur: 600; //アニメーションの時間指定
    loop: false; //繰り返しの指定
    startEvents: mouseleave">
</a-gltf-model>
```

4.1 メインページ

背景の VR 画像は,A-Frame を用いて実装しており,周りの教室名やリスト,ボタン等は HTML を用いて,作成した.

HTML で作成した,メインページに iframe タグを用いて,A-Frame を用いて作成した HTML ファイルを挿入し,一つのページとしている.

```
<iframe id="main_yahaba_aframe"
  src="" //js で A-Frame のファイルを挿入.
  frameborder="0" //frame のボーダーラインの削除
  onload="iframe_load()" //iframe がと見込まれたときの処理
></iframe>
```



※実際のイメージ

図 14 iframe タグのイメージ図

実際のメインページ



図 15 メインページ画面

A-Frame での背景設定は下記のとおりである。

```
<a-sky id="back-ground"
  src="" //src には javascript で各教室の判定を行い,3D 画像を挿入
  position="0 0 0" //位置を指定(X Y Z)
  rotation="0 -120 5" //回転を指定(X Y Z)
  animation="property: rotation; //背景を回転させるアニメーション
  from: 0 0 0; //開始の位置
  to: 0 360 0; //終わりの位置
  loop: true; //繰り返しの指定
  dur: 60000; //時間の指定
  easing: linear">
</a-sky>
```

サイトの左右にある,矢印ボタンが教室一覧リストをクリックした際に,syokai_page という関数に id を二つ付与して,id1 と id2 からサイトを再読み込みし,選んだ教室にページを更新している.



図 16 メインページ左矢印

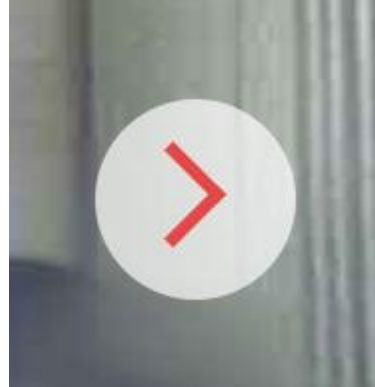


図 17 メインページ右矢印

```
<li><a onclick="syokai_page('5','1')">計算機工学実験室</a></li>
<li><a onclick="syokai_page('5','2')">工作室</a></li>
<li><a onclick="syokai_page('5','3')">通信実験室</a></li>
<li><a onclick="syokai_page('5','4')">画像実験室</a></li>
<li><a onclick="syokai_page('5','5')">端末室</a></li>
```

実際の syokai_page 関数

```
function syokai_page(id1, id2) {
  window.location.href = "../vr_main_html/yahaba_main.html?a=" +
    'text' + "=" +
    id1 +
    "=" +
    id2;
}
```

教室の説明文については,右下のボタンから表示非表示が可能となっている.

ページを読み込み最初の段階では,表示されており,×ボタンを押すと非表示にすることができる

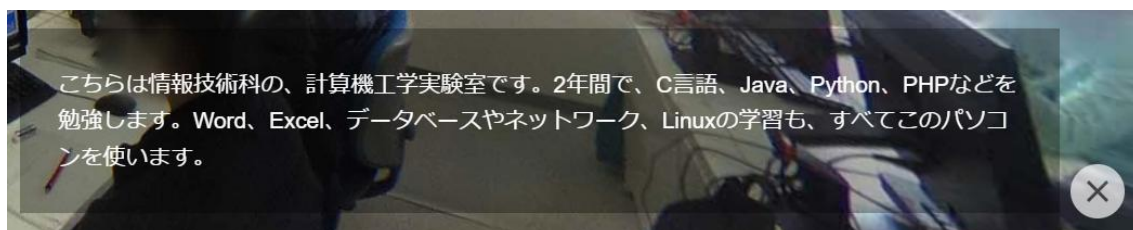


図 18 教室の説明文表示中の画面

非表示の状態が下記の写真である.再度表示したい場合は,本のマークになっているボタンを押すことで再度教室の説明文を,表示することができる.



図 19 教室の説明文非表示の画面

教室一覧のリストについては,科の名前が書かれているボックスを,マウスオーバーするとその科の教室一覧が,表示されるようになっている.

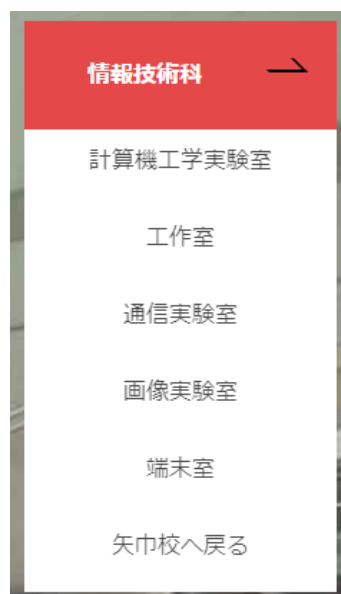
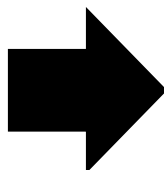
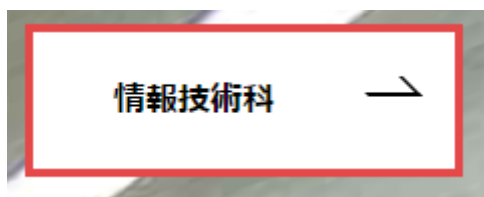
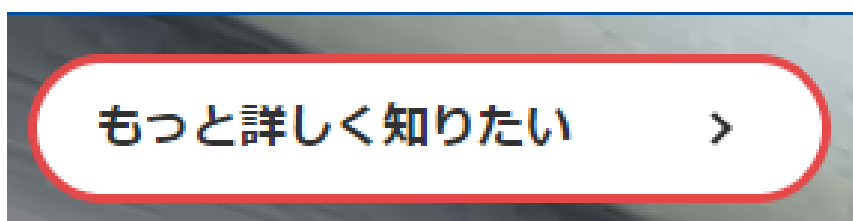


図20 教室一覧リストのマウスオーバー

教室名をクリックすると,その教室に遷移することができる

図21 教室一覧リストのマウスオーバー



もっと詳しく知りたいボタンから,〇×ゲームページに遷移することができる.

4.3 ○×ゲームページについて

○×ゲームページは、メインページと同じHTMLファイルを使用しており、iframeの中を○×ゲームのA-Frameファイルに変更して実装している。

最初の画面が下記の写真である。「スタート」と書かれたボタンを押すことで○×ゲームをスタートすることができる。



図 22 ○×ゲームスタート画面

問題は1から5問表示され、左右の○・×のオブジェクトをクリックし、ゲームを進めていく。

問題数については、教室によって少ない場合もある。

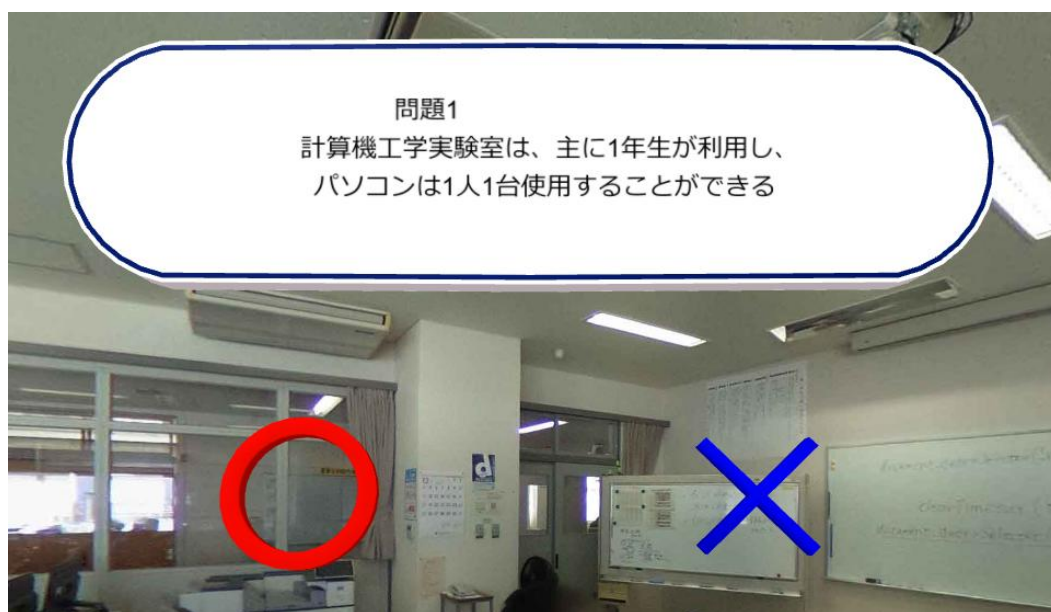


図23 ○×ゲームプレイ画面1

○か×を選択すると,正解か不正解かが表示され,選択したのがどちらかわかるように,オブジェクトが残る仕様としている.

「次へ」 ボタンを押すと次の問題へ移動することができる.

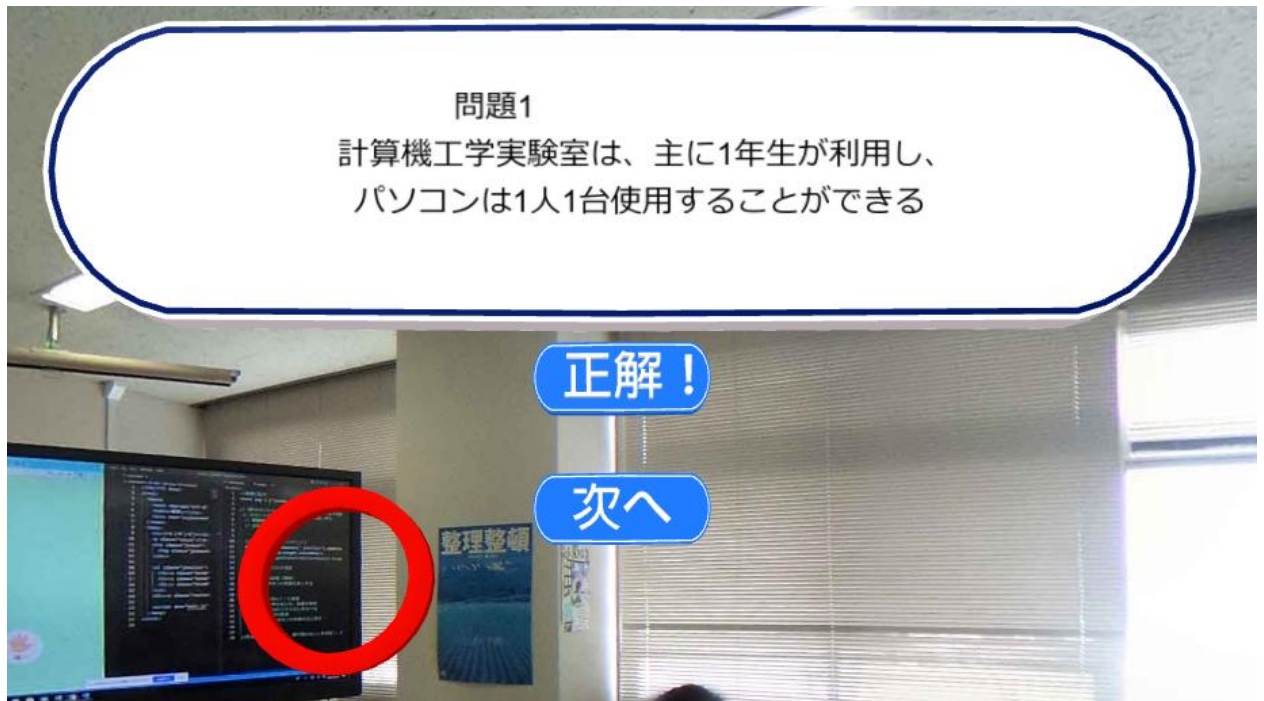


図24 ○×ゲームプレイ画面2

最後の問題が終了すると,正解した数がスコアとして表示されるようになっている.

下にある「もう一度」ボタンをクリックすると,再度〇×ゲームをプレイすることができる.

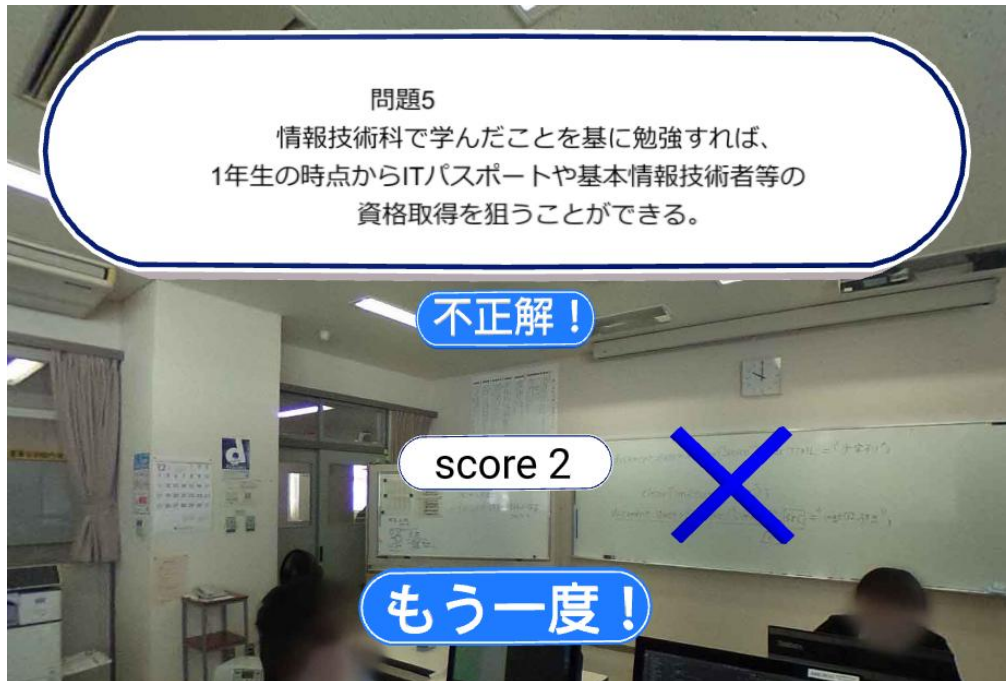


図 25 〇×ゲームプレイ画面3

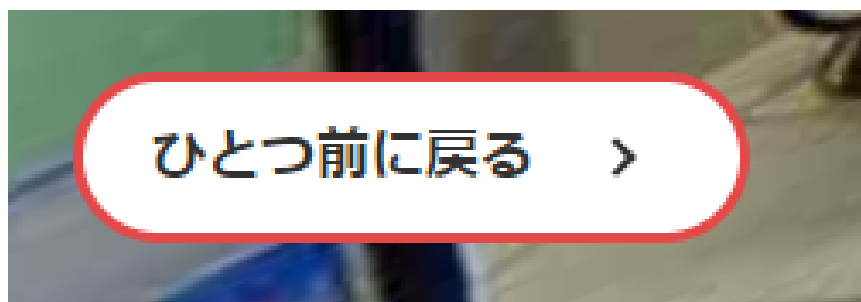


図 26 〇×ゲームから教室紹介画面へ戻るボタン

ひとつ前に戻るボタンから,いつでも〇×ゲームをやめ,教室紹介ページに戻ることができる.

4.3.1

○×問題文の作成について

問題文のファイルは csv ファイルで作成している,問題文は 1 から 5 までで,好きな数作成してもらう.問題文は改行処理の関係で,一文を 25 文字以下にし,3 文まで入れることができる.また文を区切るときは¥n(バックスラッシュエヌ)で記述する.

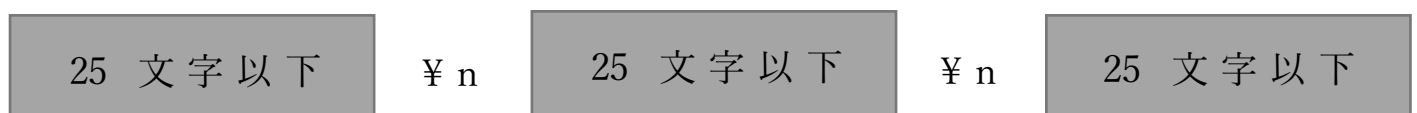


図27 ○×ゲーム問題文のイメージ

```
1 問題0,問題文は25文字以下を¥nで区切り三つまでに収める,JMd
2 問題1,計算機工学実験室は、主に1年生が利用し、¥nパソコンは1人1台使用することができる,o
3 問題2,この教室で学ぶ授業を通して、Webサイトを¥n作ることができるようになる。o
4 問題3,この教室ではC言語やJava等の¥nプログラミング言語の授業が行われる。x
5 問題4,この教室では、数学の授業はない。o
6 問題5,情報技術科で学んだことを基に勉強すれば、¥n1年生の時点からITパスポートや基本情報技術者等の¥n資格取得を狙うことができる。o
7 null,null,null
```

図28 実際の○×ゲームの問題文

実際の日本語対応のコード

```
function aframeMutlByte() {
  document.querySelectorAll('[mb-text]:empty').forEach(mb_text => {
    const text = mb_text.dataset.text
    const text_cnt = text.length //テキストの長さ
    const width = text_cnt * 1.4 //テキストの幅
    const height = 1.6 //テキストの高さ
    let cvs = document.createElement('canvas') //canvas 生成
    let ctx = cvs.getContext('2d')
    cvs.width = width * 100 //幅の設定
    cvs.height = height * 100 //高さの設定
    ctx.fillStyle = "rgb(0, 0, 0)" //テキストのカラーを指定
    ctx.font = '100pt Arial' //テキストのフォントを指定
    ctx.fillText(text, 0, 125)
    const base64 = cvs.toDataURL("image/png") //base64
    mb_text.innerHTML = `
```

4.4 読み込みの高速化

サイトが完成し,読み込み速度を調べた結果,読み込み速度が遅かったため,サイトの読み込み速度の改善を行った.原因はコードの量が多いことと,ファイル容量が大きいことが原因だと考えたため,以下に対策について記す.

4.4.1 JavaScript と CSS をインラインで記述

このように CSS は HTML タグにインラインで記述し,JavaScript も HTML タグの中に記述しインラインにして,別ファイルを読み込む手間をなくし,読み込み速度改善に取り組んだ.

```
<html>
  <body>
    <p style="color: red;">こんにちは</p>
  </body>

  <script>
    var cnt = 0;
  </script>
</html>
```

4.4.2 JavaScript と CSS を簡略化しコード量を削減

JavaScript を簡略化する前


```
$('.button').click(function(){ //通常の JavaScript 関数
  if($('.button').hasClass('on')){ //クラスが存在するかどうか
    $('.button').removeClass('on'); //クラスの削除
  }else{
    $('.button').addClass('on'); //クラスの追加
  }
})
})
```

JavaScript を簡略した後

```
$('.button').click(function(){
  $('.button').toggleClass('on'); //jQuery の toggleClass を使用
  //この関数はクラスがあれば削除し,なければ追加するので一つにまとめることができる.
})
```

CSS の簡略化する前


```
.button-right{
  background-color: #bfbfbf;
  color: #E54848;
}
.button-left{
  background-color: #bfbfbf;
  color: #2A579A;
}
```



二つの要素に同じ背景を指定

CSS の簡略化した後

```
.button{
  background-color: #bfbfbf;
  &.right{
    color: #E54848;
  }
  &.left{
    color: #2A579A;
  }
}
```



Sass を使うことで一つにまとめることができる

4.4.3 JavaScript と CSS ファイルの圧縮

ソースコードの行数が少ない方がファイルの読み込み速度が早くなるため,Minify を使ってファイルを圧縮した。



図29 Minify の画像

圧縮後は,ファイルの拡張子に min という拡張子が追加される。

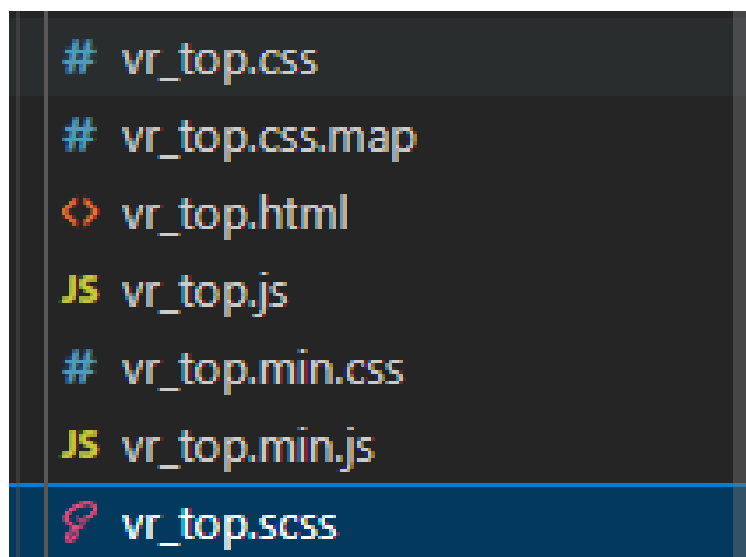


図30 実際に圧縮した後のファイル一覧

4.4.4 高速化の結果

PageSpeed Insights を用いて,サイトの読み込み速度を調べたところ,15 秒から 5 秒ほどに縮めることができた。

4.5 レスポンシブ対応

レスポンシブ対応のブレイクポイントは,JavaScript を用いて閲覧している端末を判定し,その端末にあったサイズに調整している.

今回のブレイクポイントは,端末の OS 情報を設定し,判定している.

```
if (back_hantei || (navigator.userAgent.indexOf('iPhone') > 0 && navigator.userAgent.indexOf('iPad')  
    == -1) || navigator.userAgent.indexOf('iPod') > 0 || navigator.userAgent.indexOf('Android') > 0 ||  
    navigator.userAgent.indexOf('') > 0) {}
```

CSS の方で,ブレイクポイントを設定するやり方もあるが,JavaScript でもレスポンシブ対応の処理をしなければいけなかったため,CSS と JavaScript の二つでブレイクポイントを設定するよりも,JavaScript でまとめた方が処理が減ると考え,JavaScript でブレイクポイントを設定した.

サイトのデザインやサイズだけでなく,スマートフォンではマウスオーバーが使えないため,クリックに変更をした.

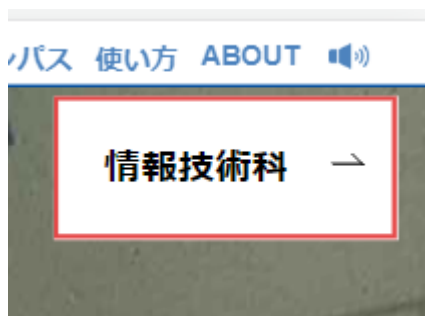


図31 教室一覧リストクリック前

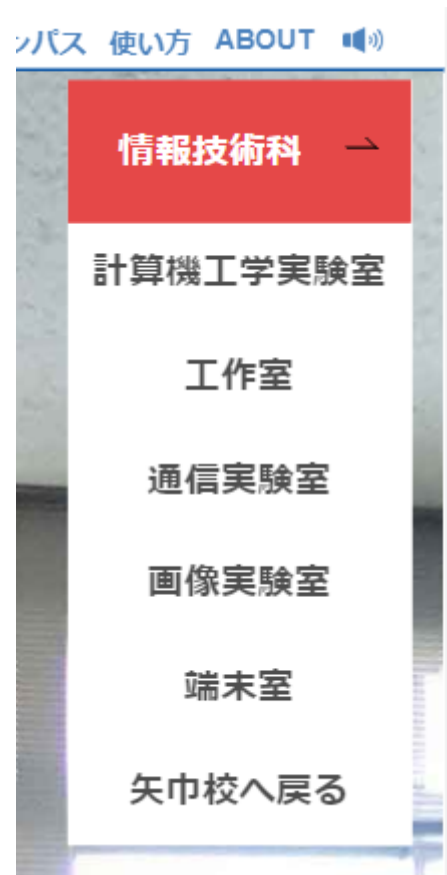
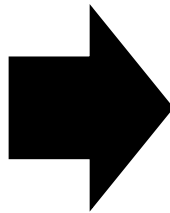


図32 教室一覧リストクリック後

元々はマウスオーバーだったが,レスポンス対応でクリックに変更した.


実際のレスポンス対応コード

JavaScript コード

```
$(".navbar-nav").addClass("res"); //ヘッダーの部分にレスポンス対応するため,  
$(".nav-item").addClass("res");   クラスを追加している.
```

CSS コード

```
.navbar-nav {  
    display: inline-flex;  
    height: 22px;  
    &.res {  
        padding-left: 0px; //レスポンス対応で要素の幅の調整  
    }  
    .nav-item {  
        list-style: none;  
        margin-left: 50px; //レスポンス対応の前の要素の幅  
        transition: .3s;  
        position: relative;  
        cursor: pointer;  
        &.res {  
            margin-left: 20px; //レスポンス対応で要素の幅を調整している.  
            &::after {  
                content: " ";  
                opacity: 0;  
            }  
        }  
    }  
}
```



4.6 ジャイロ機能制限

スマートフォンのジャイロ機能がオンになっていると、トップページや〇×ゲームの画面で、動いてしまうためジャイロをオフにする処理を加えた。

対応ページは、トップページと〇×ページである。

```
<a-entity id="camera_phone"
  camera look-controls="enabled: false;"
  magicWindowTrackingEnabled="false"> //このプロパティを無効にすることでジャイロを無効に
                                     できる
</a-entity>
```

このようにほかにも様々なプロパティがあるので、いろいろカスタマイズ可能である。

プロパティは以下のとおりである。

Property	Description	Default Value
enabled	Whether look controls are enabled.	true
reverseMouseDown	Whether to reverse mouse drag.	false
reverseTouchDrag	Whether to reverse touch drag.	false
touchEnabled	Whether to use touch controls in magic window mode.	true
mouseEnabled	Whether to use mouse to move the camera in 2D mode.	true
pointerLockEnabled	Whether to hide the cursor using the Pointer Lock API .	false
magicWindowTrackingEnabled	Whether gyroscope camera tracking is enabled on mobile devices.	true

参考 [A-Frame 公式](#)

4.7 CSV ファイルでの管理

矢中校と水沢校を合わせると,各科各教室の数がとても多いため,更新や管理がしやすいように,CSV ファイルにまとめ管理することにした.

```
1 科ごとのid,科の教室のid,教室名
2 1,1,メカトロニクス技術科\n機械加工実習場,mekatoro_kikaikakou,こちらはメカトロニクス技
3 1,2,メカトロニクス技術科\nCAD/CAM室,mekatoro_cadcam,こちらはメカトロニクス技術科の、
4 1,3,メカトロニクス技術科\n制御工学実験室,mekatoro_seigyo,こちらはメカトロニクス技術科
5 1,4,メカトロニクス技術科\n電子工学実験室,mekatoro_densi
6 2,1,電子技術科\nコンピュータ実習室,densi_computer,こちらは電子技術科の、コンピュータ
7 2,2,電子技術科\n総合実験室,densi_sougouziken,
8 2,3,電子技術科\n無響室,densi_mukyo,こちらは電子技術科の、無響室です。外部からの雑音が
9 3,1,建築科\n建築実習場,kentiku_zisyuzyou,こちらは建築科の建築実習場です。木造を中心
```


上の写真のように,科を判別する ID と教室を判別する ID の二つをつけている.

ID は 1 列目が科を判別する ID で,2 列目が教室を判定する ID となっている.

判別処理の仕組みについて

手順1 URL に二つの ID を付与しているので,その URL から ID を二つ取得する.

•URL : vrpage/main.html?2=3



id_1=2 id_2=3

手順2 取得した,科を判別するための ID_1 から科を判別する.

1列目を2行目から,id_1 と一致する数字を探していく

id_1 = 2



上から探していく

1	科ごとのid, 科の教室のid, 教室名
2	1, 1, メカトロニクス技術科\n機械加工実習場,
3	1, 2, メカトロニクス技術科\nCAD/CAM室, meka
4	1, 3, メカトロニクス技術科\n制御工学実験室,
5	1, 4, メカトロニクス技術科\n電子工学実験室,
6	2, 1, 電子技術科\nコンピュータ実習室, densi
7	2, 2, 電子技術科\n総合実験室, densi_sougou
8	2, 3, 電子技術科\n無響室, densi_mukyo, こち

手順3 取得した,教室を判別するための ID_2 から教室を判別する.

次は, 2列目から,id_2 と一致する数字を探していく.



電子技術科の無響室を取得することができる



1	科ごとのid, 科の教室のid, 教室名
2	1, 1, メカトロニクス技術科\n機械加工実習場,
3	1, 2, メカトロニクス技術科\nCAD/CAM室, meka
4	1, 3, メカトロニクス技術科\n制御工学実験室,
5	1, 4, メカトロニクス技術科\n電子工学実験室,
6	2, 1, 電子技術科\nコンピュータ実習室, densi
7	2, 2, 電子技術科\n総合実験室, densi_sougou
8	2, 3, 電子技術科\n無響室, densi mukvo, こち

4.8 A-Frame の日本語対応

A-Frame では現在、標準で日本語フォントに対応していないので、テキストを表現する `<a-text>` タグの `value` に日本語を記述しても文字化けするので、今回は HTML Shader と `html2canvas` を使い実装した。

HTML Shader とは、その名の通り 2 次元の DOM 要素をマテリアルとして、3 次元の A-Frame オブジェクト上に貼り付けることができるコンポーネントで、それは `html2canvas` というライブラリ上に成り立っている。

`html2canvas` とは任意の HTML 要素に描画された内容を canvas に描画して canvas 要素として返してくれるライブラリである。

以下に図を用いて解説する。

手順1 対象の日本語テキストを生成した canvas に描画

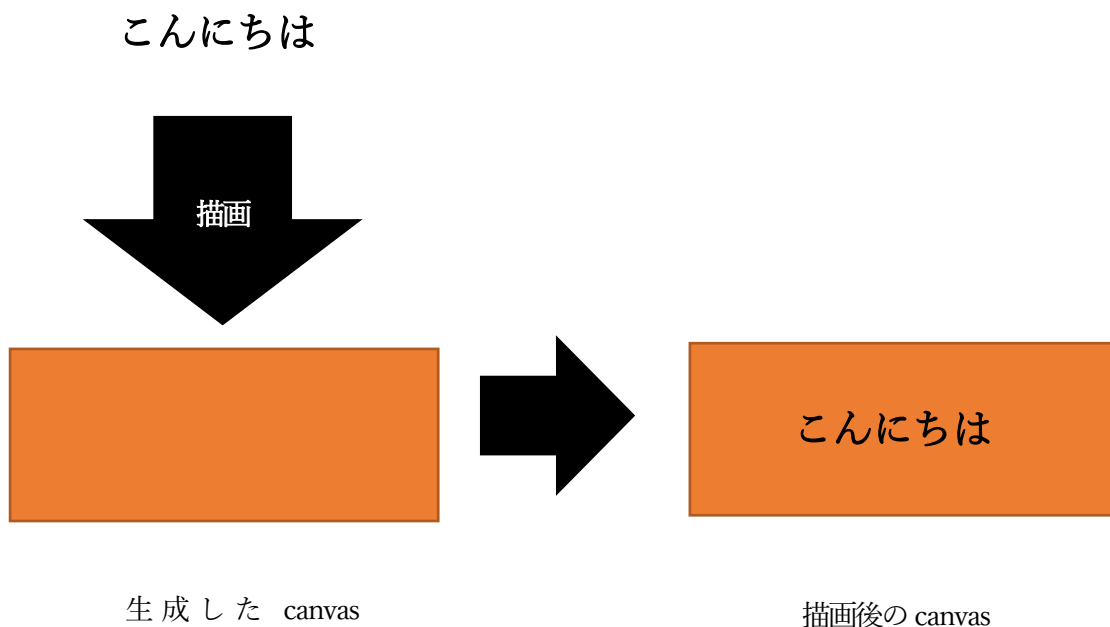


図 33 対象の日本語テキストを生成した canvas に描画

手順2 canvas 情報を base64 に変換

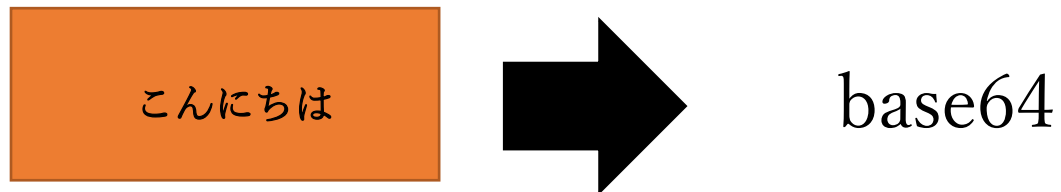


図 34 canvas 情報を base64 に変換

Base64 とは、64 種類の文字（アルファベット「a～z, A～Z」と数字「0～9」、一部の記号「+ と /」）と末尾の記号「=」を用いて一定の規則に基づきデータを変換するエンコード方式（データを他の形式に変換する際の方式）の一つである。

- ・手順3 <a-image>タグの src に base64 を指定

<a-image>タグは A-Frame で画像を表現するためのタグである.

<a-image src="base64">

HTMLShader

こんにちは

HTMLShader を使い,canvas で作成した
二次元のテキストを,3次元の A-Frame
オブジェクト上に貼り付ける.

a-image 要素

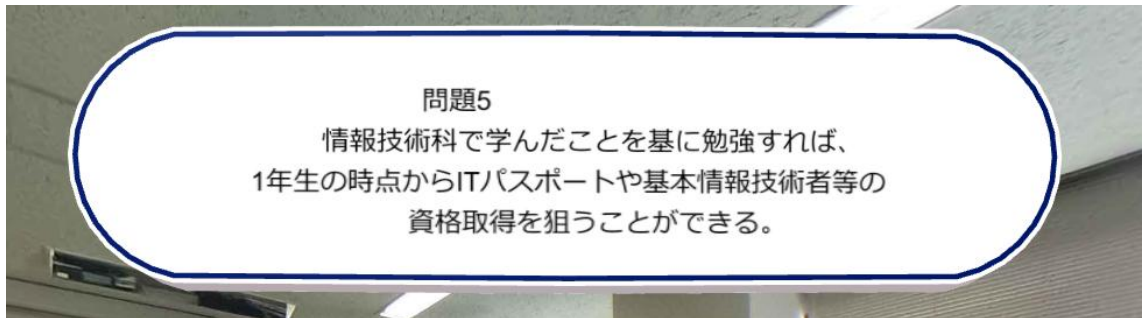
こんにちは

以上の手順で,日本語対応を実現することができる.

図 35 <a-image>タグの src に base64 を指定

4.8.1 改行処理

○×ゲームの問題文の文字量が多く,改行しなければ画面に収まらないため,改行処理を加えることとした.



一つ前の項目で説明した通り,A-Frame の日本語対応は,テキストではなく画像扱いなため,通常の改行処理が効かない.そのため JavaScript で改行する処理を実装した.

情報技術科で学んだことを基に勉強すれば,¥n 1 年生の時点から IT パスポートや基本情報技術者等の ¥n 資格取得を狙うことができる.

実際の問題文

改行処理のしくみ

※例

A-Frame ¥n を用いた ¥n ○×ゲーム

csv ファイルで問題文を作成するときに,問題文を区切りたい場所に ¥n を入れる.

A-Frame ¥n を用いた ¥n ○×ゲーム

この問題文を ¥n で区切り三つに分ける.

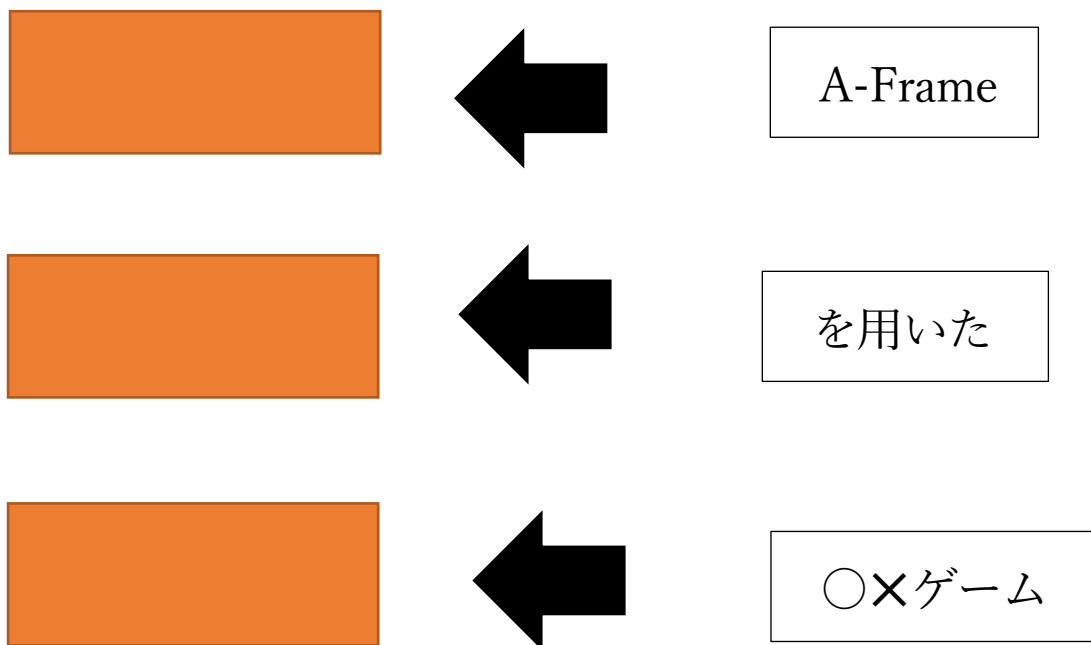


図 36 改行処理の解説図

三つに分けたものを一つずつ日本語対応処理を行うことで改行処理を実装した.

4.9 3D モデルの製作

4.9.1 3D モデルの概要

Blender を用いて岩手県立産業技術短期大学の矢巾キャンパス,水沢キャンパスを作成した.これらを glb ファイルとして出力したものを A-Frame を用いてサイトで使用する.地面は敷地分を再現し,校庭や駐車場など平面が広がる部分には,オブジェや産技短のロゴ等の装飾を設置した.

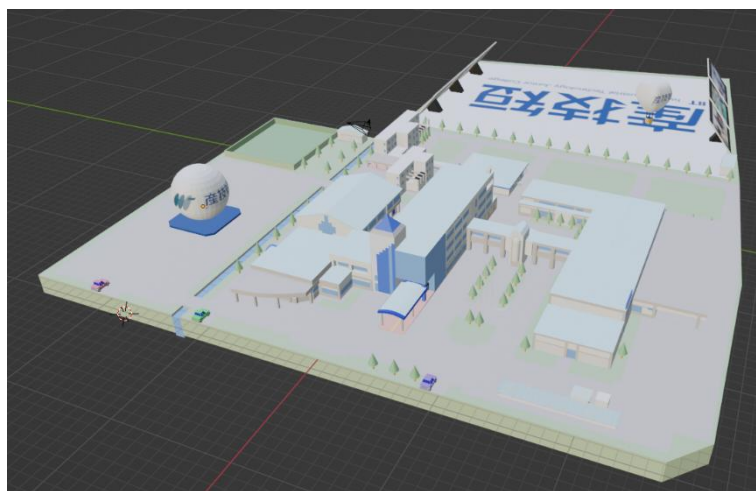


図37 矢巾キャンパスの3Dモデル



図38 水沢キャンパスの3Dモデル

4.9.2 3D モデルの製作上の留意点

学校の 3D モデルを製作する上で以下の 3 つの点に留意した。

- ① サイトの読み込みを早くするために、モデルの容量を出来るだけ抑えること
- ② 見栄えの良いモデルにすること
- ③ サイトに反映させると設定していた色より濃い色で表示されるため、
目標とする色よりも薄い色を付ける必要があること

の 3 つのことを考えた。

まず①では、容量削減のために点・辺の溶解と削除(モデルの形はそのままに頂点数を減らすこと)を行った。頂点数が多いことは容量が大きくなる原因の一つであるため、この作業で約 1,000KB 以上の削減に成功した。

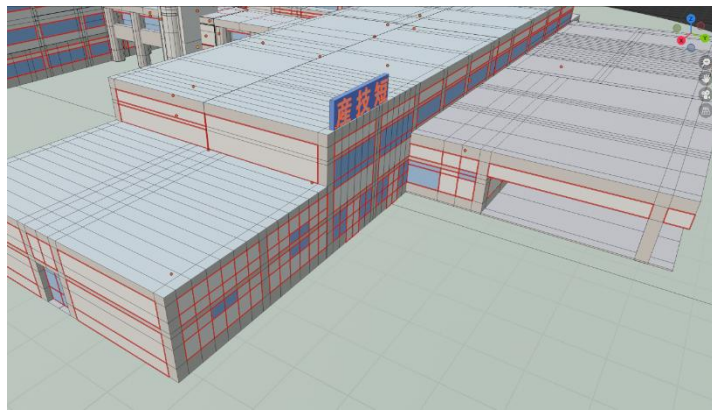


図 39 製作直後の矢巾キャンパスの実習棟

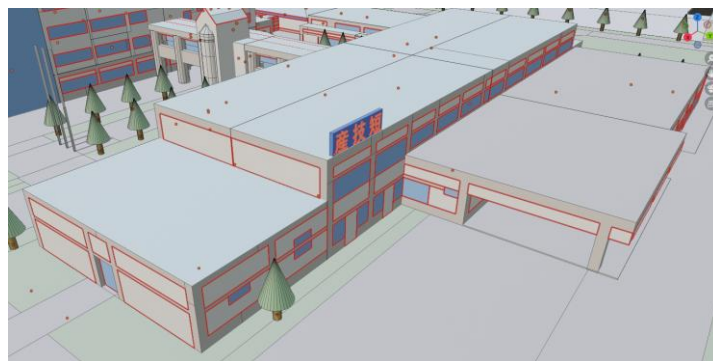


図 40 点・辺の溶解・削除後の矢巾キャンパスの実習棟

また,②では,矢巾キャンパスの駐車場と校庭,水沢キャンパスの校庭に以下のように装飾を設置し,モデルの見栄えを良くすることに努めた.

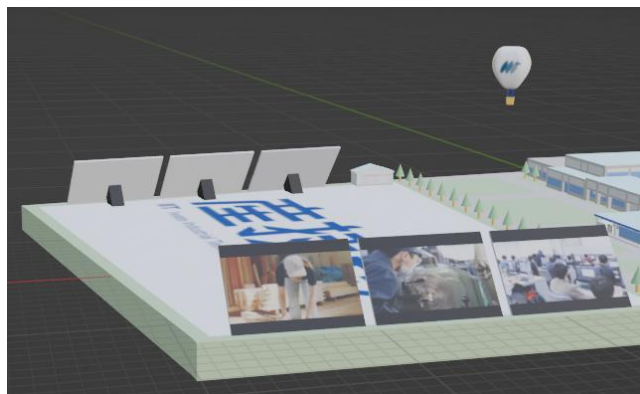
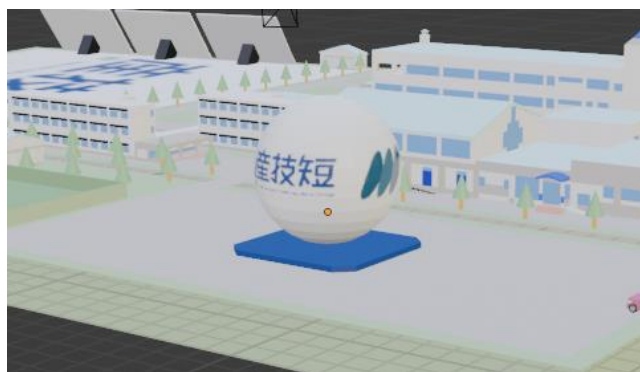
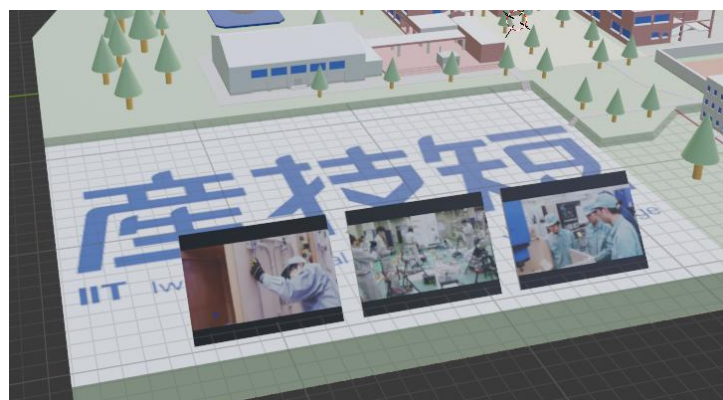


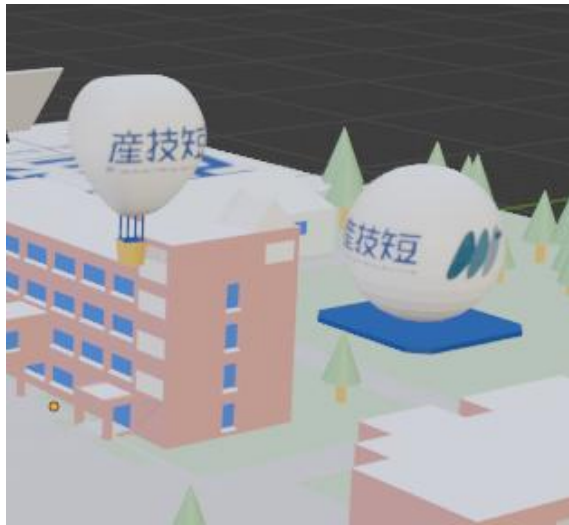
図 41 矢巾キャンパスの装飾①



矢巾キャンパスの装飾②



水沢キャンパスの装飾①



水沢キャンパスの装飾②

4.10 A-Frame に 3D モデルを反映

4.10.1 glb ファイルについて

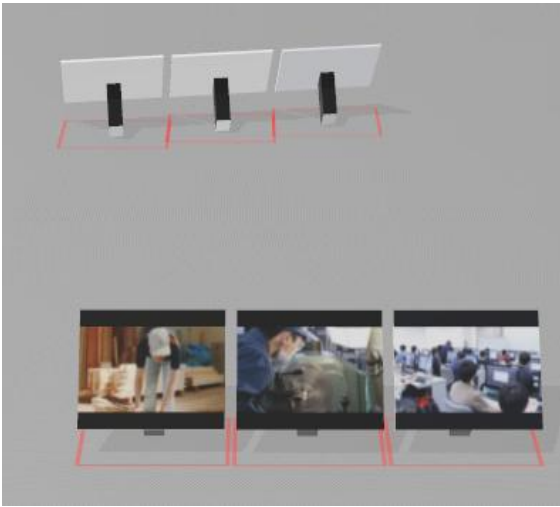
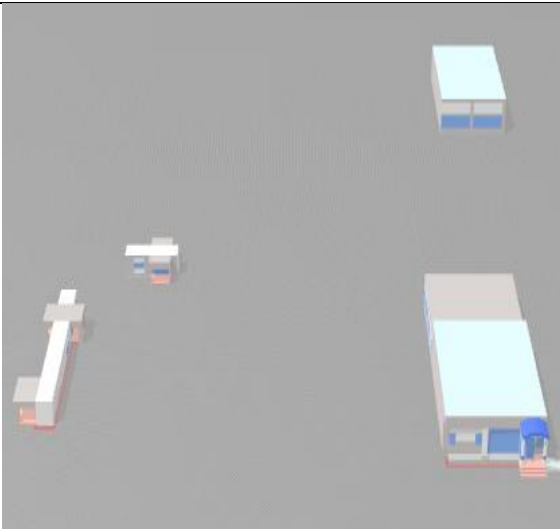
A-Frame 上において,Blender ファイルではサイトで表示することが出来なかった.しかし,obj ファイルと glb ファイルのどちらかに出力したものを使用すれば表示することが出来た.その後,サイトの動作を軽くするために容量の節約について考えた結果,glb ファイルとして出力したものを使用することに決定した.

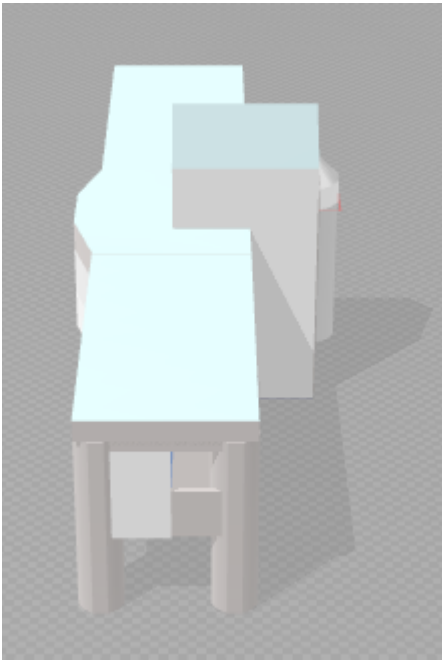
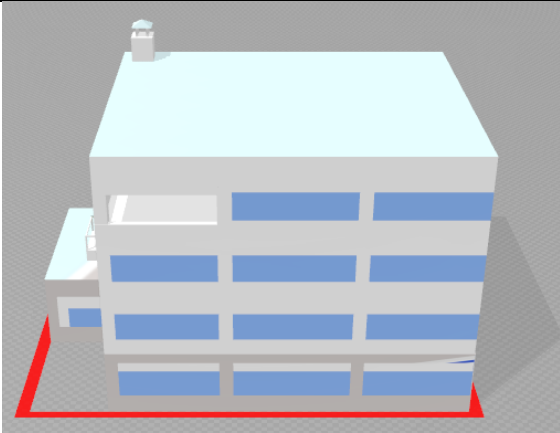
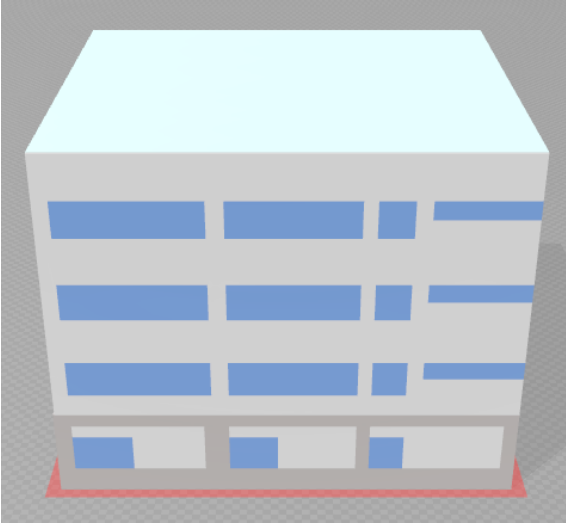
4.10.2 分割した glb ファイルの種類

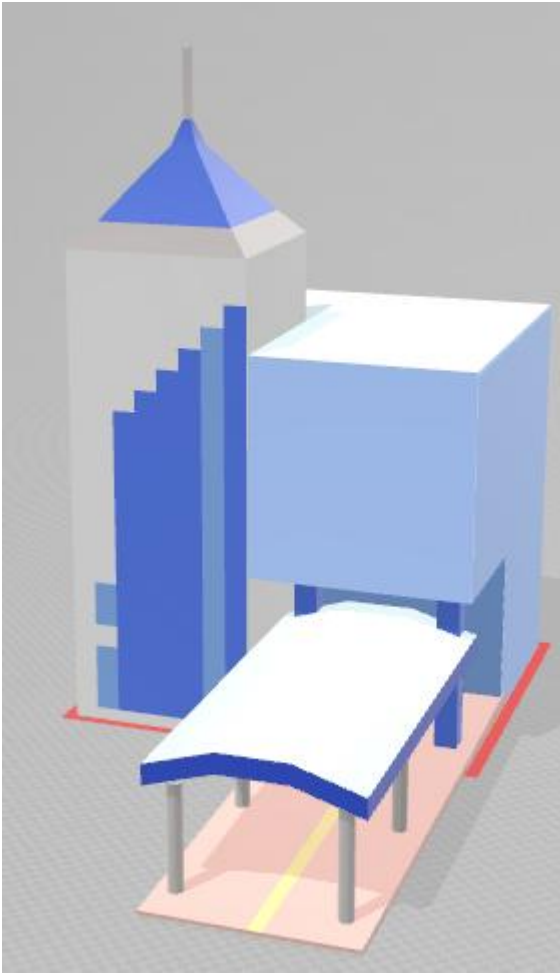
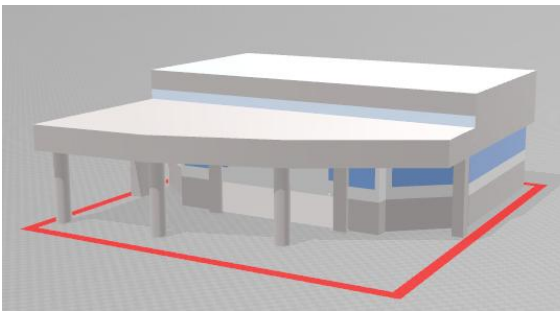
3D モデルをサイトに反映させる際は,クリックで校舎の写真に飛ぶモデルの glb ファイルを分割する必要がある.現在の分割したものを以下に示す.

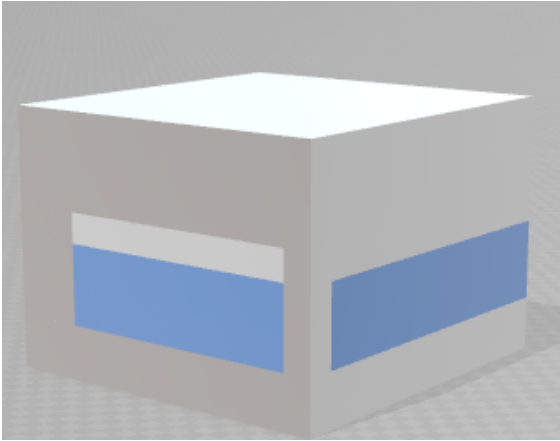
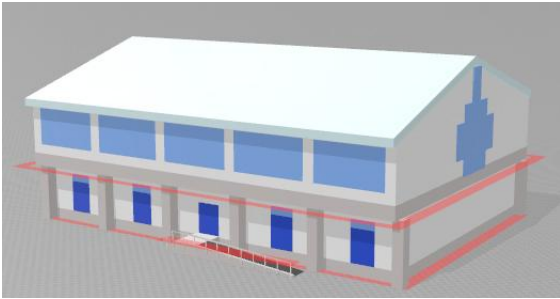
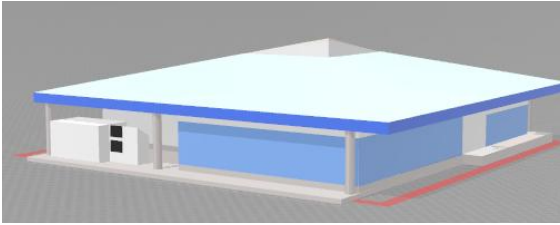
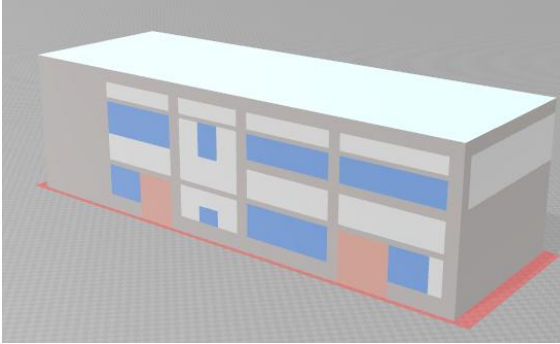
矢巾キャンパス

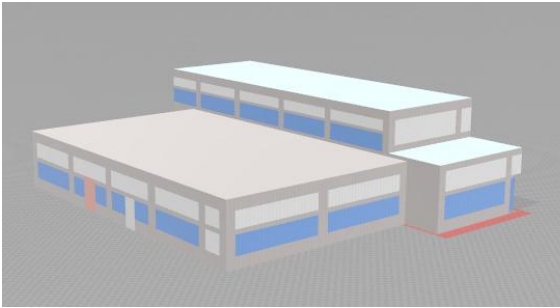
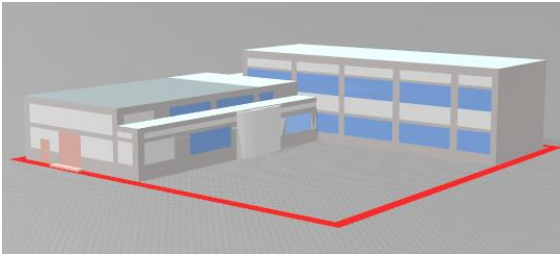
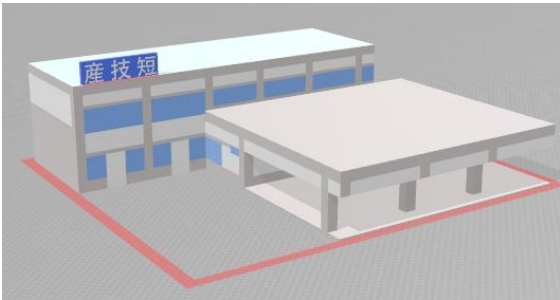

glb ファイル名	場所	写真
yahaba_dodai	地面と,装飾①	
yahaba_dodai2	装飾②	

yahaba_picture	装飾③	
yahaba_tuuro	通路や教室 (クリックで 360 度写真のページへ遷移しないもの)	

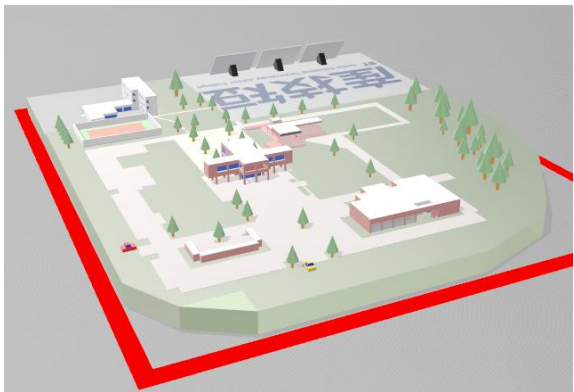
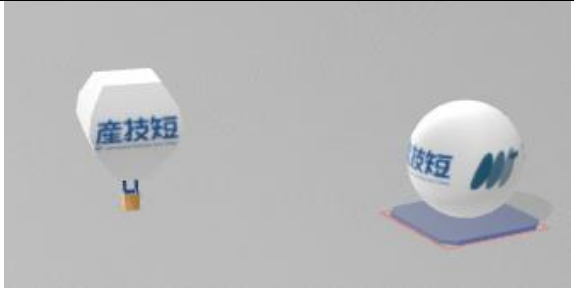
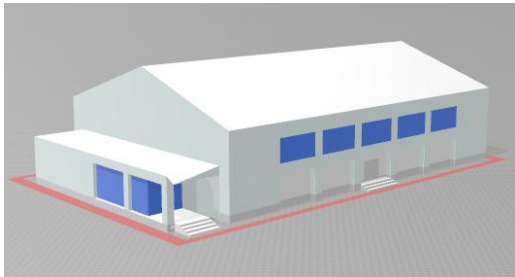
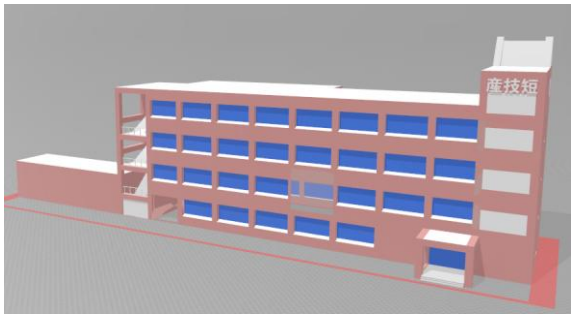
<p>yahaba_tuuro_</p> <p>2</p>	<p>通路②（校舎間の渡り廊下）</p>	
<p>zyouhou</p>	<p>本館棟を3分割（奥）</p>	
<p>densi</p>	<p>本館棟を3分割（真ん中）</p>	

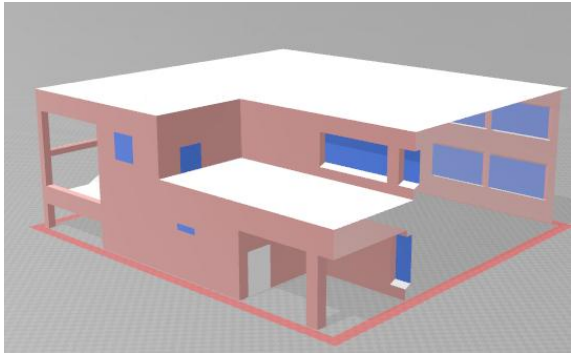
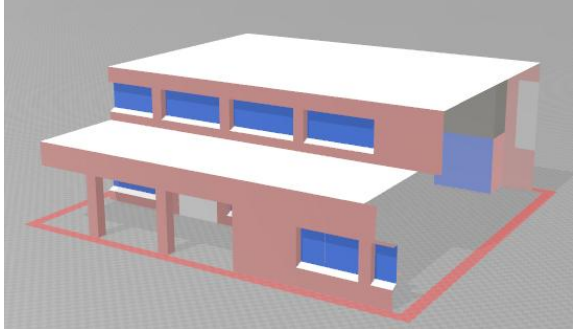
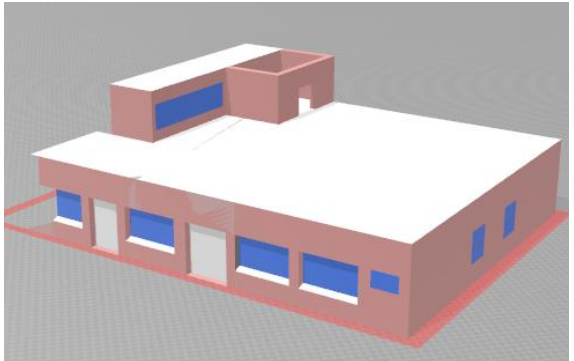
<p>honkan_genka n</p>	<p>本館棟を3分割 (前) クリックすると多目的ホール</p>	
<p>tamoku</p>	<p>多目的ホール</p>	

tosyo	図書室	
taikukan	体育館	
syoku	学生ホール(食堂)	
senkou	<p>実験研究棟を十字に 割って4分割(上から見て 左下)</p> <p>クリックすると専攻 科</p>	

kentiku	<p>実験研究棟を十字に 割って4分割(上から見て 右上)</p> <p>クリックすると建築 科</p>	
sande	<p>実験研究棟を十字に 割って4分割(上から見て 左上)</p> <p>クリックすると産業 デザイン科</p>	
mekatoro	<p>実験研究棟を十字に 割って4分割(上から見て 右下)</p> <p>クリックするとメカ トロニクス技術科</p>	
ryou	<p>学生寮</p>	

水沢キャンパス

glb ファイル名	場所	写真
mizusawa_dodai	地面や装飾,教室 (クリックで360 度写真のページへ 遷移しないもの)	
mizusawa_dodai2	装飾②	
taikukan_m	体育館	
honkan_m	本館棟	

denkigizyutu	<p>実験研究棟を3 分割（奥）</p> <p>クリックすると 電気技術科</p>	
seisangizyutu	<p>実験研究棟を3 分割（真ん中）</p> <p>クリックすると 生産技術科</p>	
kentikusetubi	<p>実験研究棟を3 分割（前）</p> <p>クリックすると 建築設備科</p>	

第5章 完成したWEBサイト

完成したWebサイトをページごとに以下に示す。

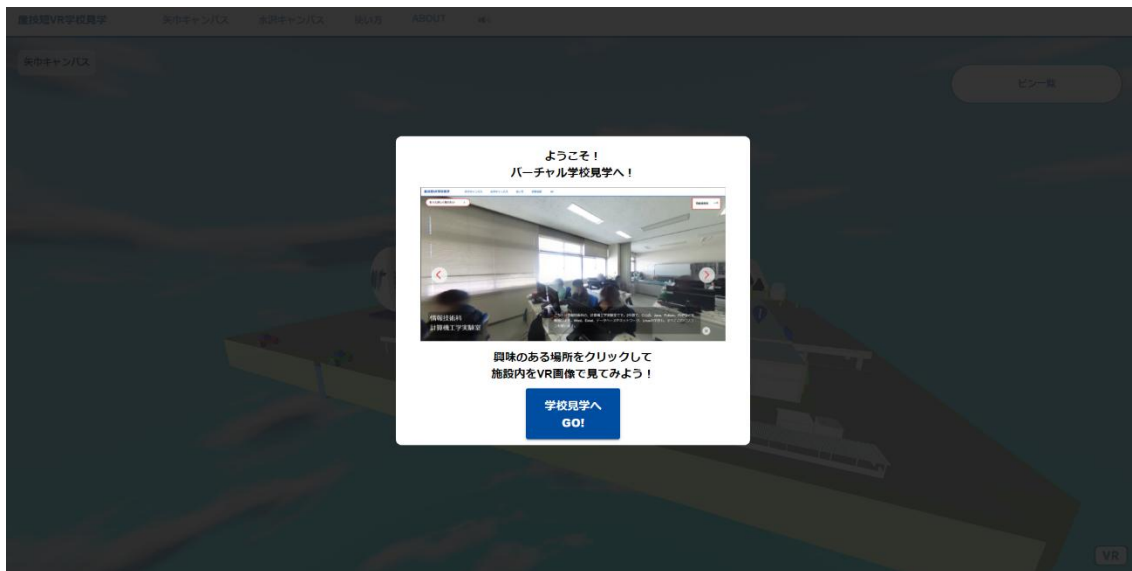


図42 モーダルウィンドウ



図43 矢巾校の3Dオブジェクト

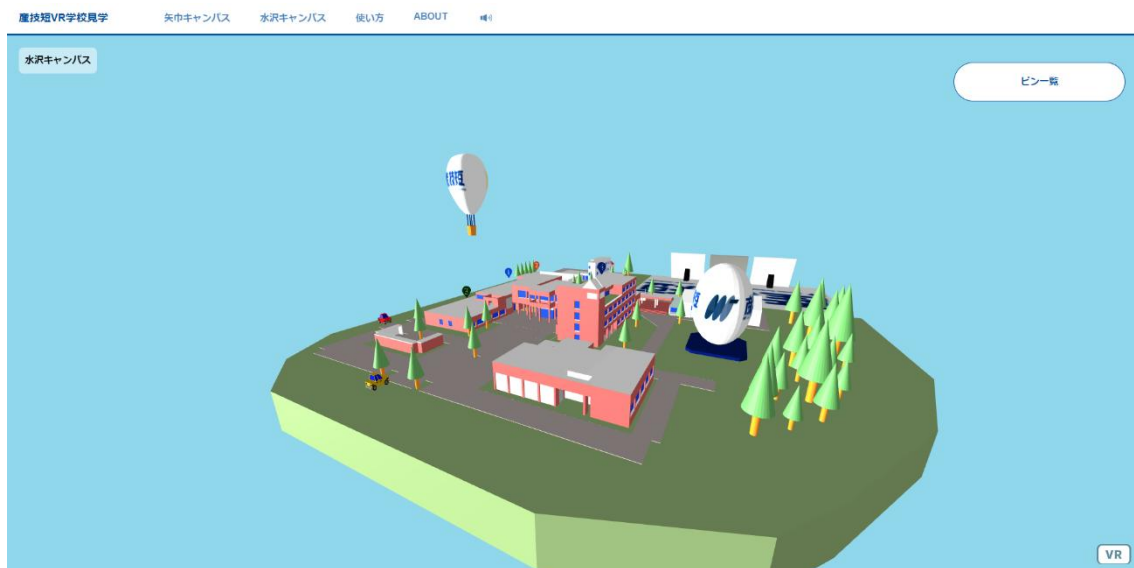


図 44 水沢校の 3D オブジェクト

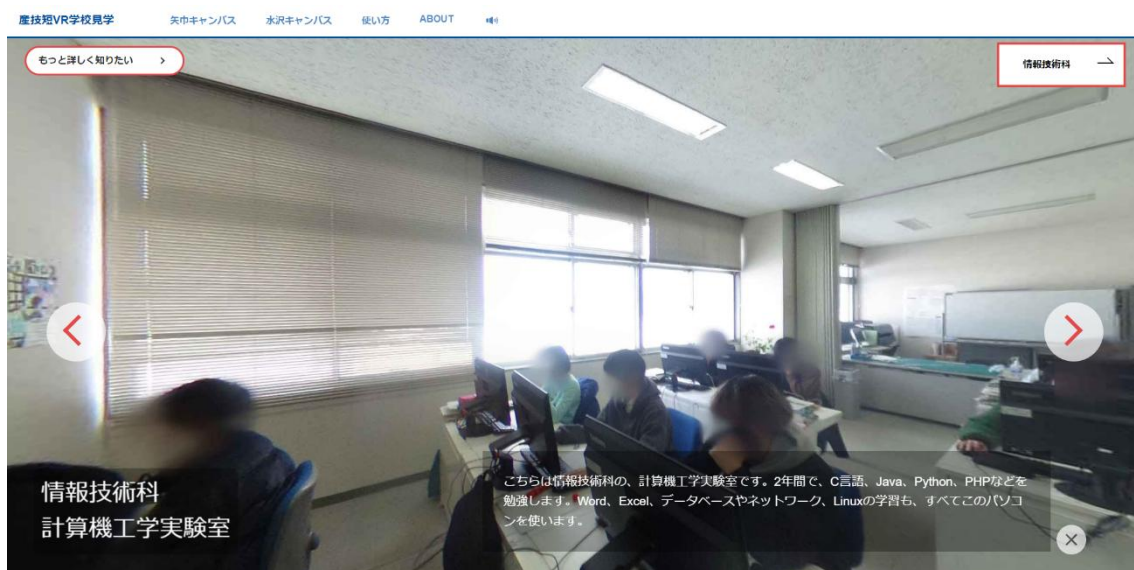


図 45 紹介ページ



図46 〇×ゲーム スタート画面

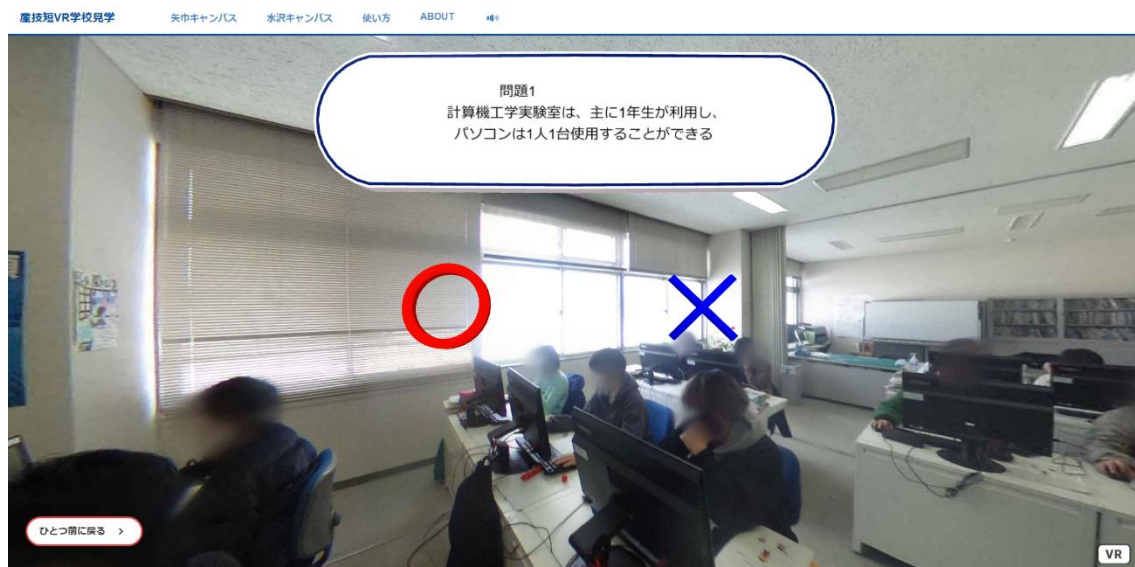


図47 〇×ゲーム プレイ画面

第6章 おわりに

6.1 評価

6.1.1 良かった点

A-Frame など授業で習わなかった新しい技術に触れ,それを基にして,やりたいことを形にすることが出来た.

2023 年実施の今回の産技短展の展示では,実際にサイトに触れて学校を見て回り「面白い」と言ってくださった方が多くいらしやった.よって,目的としていた「使用感と見栄えの良いサイトを作る」ことは達成できたと感じた.「より多くの人に産技短のことを知ってもらう」という目標については,これから実際に公式サイトに載せられてからのことなので,期待したい.

6.1.2 悪かった点

- ・○×ゲームの問題を全教室分実装出来なかった.
- ・元々は各科でモチーフを分けた色々なゲームを用意する計画だったが,出来なかった.
- ・3D モデルをサイトに反映させた際に,クリック判定が置かれていない地面や装飾等のモデルが縦に伸びてしまう問題があり,解決出来なかった.

6.1.3 今後について

本研究は,3DCG と 360 度写真で分かりやすく紹介することで,受験生に当校のことを知ってもらいたいということと,その上で使用感と見栄えの良いサイトを目指すということを念頭に,今回の卒業研究である A-Frame を用いた Web オープンキャンパスサイトの作成に取り組んだ.私たちが作成したサイトが,受験生の方が少しでも深く産技短のことを知れる助けになれば幸いである.

また,今後この研究が引き継がれることがあれば,前述した悪かった点の改善を目指してほしいと共に,サイトの見栄えをさらに良くしていった欲しい.

参考文献・引用

A-Frame: Hello WebVR. 2023 年 1 月 1 日. <https://aframe.io/> 2023 年 3 月 6 日

CSS: CSS リファレンス - CSS: カスケーディングスタイルシート. CSS リファレンス. 2022 年 10 月 3 日

<https://developer.mozilla.org/ja/docs/Web/CSS/Reference> 2023 年 3 月 6 日

JavaScript: JavaScript リファレンス - MDN Web Docs. JavaScript リファレンス

2022 年 12 月 7 日

<https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference> 2023 年 3 月 6 日

akito fujii. A-Frame で日本語を表示(スクリプトで実装) – Qiita. A-Frame で日本語を表示(スクリプトで実装) 2022 年 7 月 18 日.

<https://qiita.com/afujii/items/d1885710acaf7b35fe03> 2023 年 3 月 6 日

nchujimiyama. SCSS の基本的な書き方 – Qiita. SCSS の基本的な書き方

2020 年 02 月 15 日 <https://qiita.com/nchujimiyama/items/8a6aad5abead39d1352a>

2023 年 3 月 6 日

[JavaScript 公式サイト](#)

[A-Frame 公式サイト](#)

[A-Frame 日本語対応](#)

[SCSS の書き方参考](#)

謝辞

本研究の作成にあたり、准教授石舘先生、講師安倍先生には、ご指導ご鞭撻を賜りました。ここに深謝の意を表します。

広報部会の方々、各科の先生方、学生の方々には、本研究の360度写真の撮影にあたり、ご協力を頂きました。厚く御礼申し上げます。