

観光振興を目的としたクイズ作成システムの提案

広瀬研究室 4 年
C1201871 富樫雄斗

令和 5 年度

概要

朝日地域では世帯数と人口が大きく減少しており、自然や文化など地域資源を活用した観光振興が求められる。また、市民参加のまちづくりは急速に広まりつつあり、状況に応じて情報を編集して媒体を適切に選ぶ必要がある。近年、観光デジタルトランスフォーメーションが推進され、旅の魅力がデジタルの力でさらに高まっている。そこで本研究では、山形県鶴岡市朝日地域の観光振興を目的とし、マップと連動させたクイズ作成システムを提案する。(205文字)

目次

第 1 章	はじめに	5
1.1	背景	5
1.2	目的	5
第 2 章	先行研究と類似サービス	7
2.1	先行研究	7
2.2	類似サービス	7
2.3	事例の課題	7
第 3 章	提案	9
3.1	課題の解決案	9
3.2	本システムのメリット	9
第 4 章	システム設計	11
4.1	要件定義	11
4.2	使用する技術と要素	11
4.3	システムの流れ	12
第 5 章	システム開発	15
5.1	Leaflet によるマップ生成	15
5.2	緯度と経度の取得	16
5.3	Google Apps Script による入力フォームの生成	17
5.4	Google スプレッドシートの取得と反映	19
5.5	クイズの内容の切り替え	21
5.6	解答の判定	22
5.7	アイコンの色変更	23
第 6 章	実験	25
6.1	実験の概要	25
6.2	実験結果	25
6.3	考察	26
第 7 章	結論	27

謝辭	29
參考文獻	31

第1章

はじめに

本章では研究の背景をまとめ、それを踏まえた目的について説明する。

1.1 背景

山形県鶴岡市朝日地域は豊かな森林資源に恵まれ自然と調和した歴史と文化に育まれてきたものの、山間地が多く有数の豪雪地帯である上に学校や病院等の施設への距離も遠く、平野部に比較して気象・風土・地形の不安要素から厳しい定住環境にあり、市全体でも人口減少や少子高齢化が進む中特にその傾向が顕著で、地域資源を活用した観光振興が求められる [1]。

そのため、市民参加のまちづくりは急速に広まりつつあり、歴史的環境の保全活動、密集市街地における防災まちづくり、公共施設のデザイン、商店街の活性化、町並み景観づくりなど、多く取り組まれている。まちづくり情報を編集して発信していくためには、意識的な記録・整理と目標や状況に応じた編集を行い、発信する媒体を適切に選ぶ必要がある [2]。

近年、デジタル化での効率化と収集されるデータの分析によって、ビジネス戦略の再検討や新たなビジネスモデルの創出といった変革を行う、観光デジタルトランスフォーメーションが推進されている。旅行者の利便性や観光産業における生産性が向上して旅の魅力がデジタルの力でさらに高まり、旅行者や地域をより豊かにすることができる [3]。

以上のことから、地域の魅力を効果的に伝えられるマップと連動させたクイズ作成システムを構築する。

1.2 目的

本研究では、専門的な知識がない人でも容易に編集可能なクイズ作成システムを提案する。それにより、朝日地域の自然、文化、食、歴史などに興味を持ってもらえるシステムを目指す。

第2章

先行研究と類似サービス

本章では先行研究と類似サービスを挙げてその課題について考察する。

2.1 先行研究

観光分野にクイズを活用している事例を取り上げる。

ご当地グルメを題材としたクイズの事例

王怡青らは、全国各地の地域活性化に貢献することを目的とし、旅の感覚を味わえるアプリケーションを提案している [4]。まずゲーム開始時に初期所持金が与えられ、これを 0 円にすることをゲームの終了条件とした。クイズモードで 4 択クイズを解き、正解であればノンクイズモードに遷移し、関連する観光情報を提示する。このモードにおいてグルメやグルメに関連する場所を選択し、その対価の支払いによって所持金を減らすことができる。ご当地グルメ・コンテンツの情報源については、Wikipedia と旅行ガイドブックを使用している。

クイズラリープラットフォームの事例

明石拓弥らは、ゲーミフィケーションに基づく地域理解動機付けのためのクイズラリープラットフォームを提案している [5]。スマートフォンの GPS 機能を利用し、参加者はそのスポットを訪れてはじめてタスクをこなせるようにして、住民が自分の足でその場所を訪れるきっかけを作る。また、スポットにちなんだクイズや謎解きでその場所の学びのきっかけを作り、地域理解に向けた内的動機づけを行う。さらに、タスクをこなすとポイントを獲得でき、ポイントに応じたコンテンツやステージ、プレゼント応募などの外的動機づけによって参加者のモチベーションの維持を図る。

2.2 類似サービス

QuizGenerator

learningBOX 株式会社では、誰でも無料でオリジナルのクイズや問題を作成できるツールを提供している [6]。択一問題や正誤問題、記述問題や並び替え問題などの形式で作ることができる。作成方法にはフォーム・テキスト・エクセルの 3 つがあり、限定公開か一般公開かも選べるようになっている。

2.3 事例の課題

2.1 節を踏まえ課題について考察する。

- Wikipedia の特徴として、膨大な量のコンテンツが掲載されているものの、不特定多数の人が記事を編集しているため、情報源としての信頼性に欠ける部分がある。
- クイズのジャンルがご当地グルメに限られており、その他の分野で使用することができない。

第3章

提案

2.2 節で挙げた先行研究と類似サービスの課題を解決できるようなシステムを構築する。

3.1 課題の解決案

クイズの情報源については、その土地のガイドブックまたは公式 Web サイトに限定し信頼性の向上を図る。ご当地グルメだけでなく、歴史・文化・生態系・自然など様々な分野に適用できるようにする。また、作成する媒体として Web ページを設定し、PC 及び携帯端末どちらの使用でも閲覧可能にする。

3.2 本システムのメリット

本システムにおいて以下のメリットが挙げられる。

- 地点の場所やクイズの内容を選択・入力し、観光地の情報を現地の方でも容易に追加してもらえる。
- コンテンツが PC では横に、携帯端末では縦に並び、使用するデバイスに適して表示される。

第4章

システム設計

システムの要件を定義し使用した要素を挙げる。

4.1 要件定義

システムとして複雑ではなく扱いやすくなるような要件を定義する。

- 操作説明の常時確認
画面下部にマーカーの追加方法やクイズの答え方について記載し、いつでも操作説明を確認できるようにする。
- 専門的な知識を必要としない編集
地点の内容を入力したり選択したりすることによって、位置・説明・問題文などを容易に追加できるようにする。
- 地点の詳細の表示
マップ上のマーカーをクリックすることで、名称・概要・住所・URL がポップアップで見られるようにする。
- 正解状況の反映
正解した問題のマーカーを青から赤に変更し、ブラウザ上に情報を保存してリロードしても保持できるようにする。

4.2 使用する技術と要素

以下の技術と要素を使用しバージョンとともに記載する。

4.2.1 HTML

HTML (HyperText Markup Language) とは、ウェブサイトのコンテンツの構造を作るために使うマークアップ言語である [7]。バージョンを HTML Living Standard とし、クイズとスタンプラリーのページに使用する。

4.2.2 CSS

CSS (Cascading Style Sheets) とは、スタイルシート言語で、HTMLなどで記述された文書の体裁や見栄えを表現するために用いられる [8]。バージョンを CSS3 とし、タブやスタンプの配置など、Web ページのデザインに使用する。

4.2.3 JavaScript

JavaScript とは、ウェブサイトに対話性を追加するプログラミング言語である [9]。バージョンを JavaScript ES2023 とし、マップ上の地点のアクションやクイズの正誤判定に使用する。

4.2.4 Leaflet

Leaflet とは、携帯端末で閲覧しやすく相互作用的なオープンソース JavaScript ライブラリである。わずか約 42KB で、ほとんどの開発者が必要とする全てのマッピング機能を備えている [10]。バージョンを Leaflet 1.3.0 とし、マップ上の地点へのマーカーの配置に使用する。

4.2.5 Google Apps Script

4.2.6 Google スプレッドシート

4.3 システムの流れ

本システムの大まかな流れについて説明する。(図.4.1)

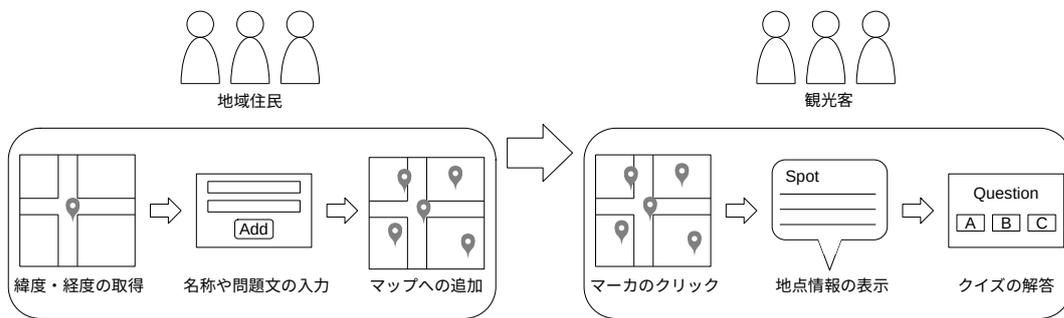


図 4.1 システムの流れ

第5章

システム開発

4章の要件定義を踏まえ本システムを設計する。前半は編集画面、後半はプレイ画面について説明する。

5.1 Leaflet によるマップ生成

鶴岡市周辺のマップになるよう中心とズームレベルを設定する。OpenStreetMap と国土地理院のタイルも選択できるようにし、プレイ画面も同様に行う。Leaflet の機能として、ドラッグによるマップの移動やマウスホイールによる拡大・縮小ができる (図.5.1)。

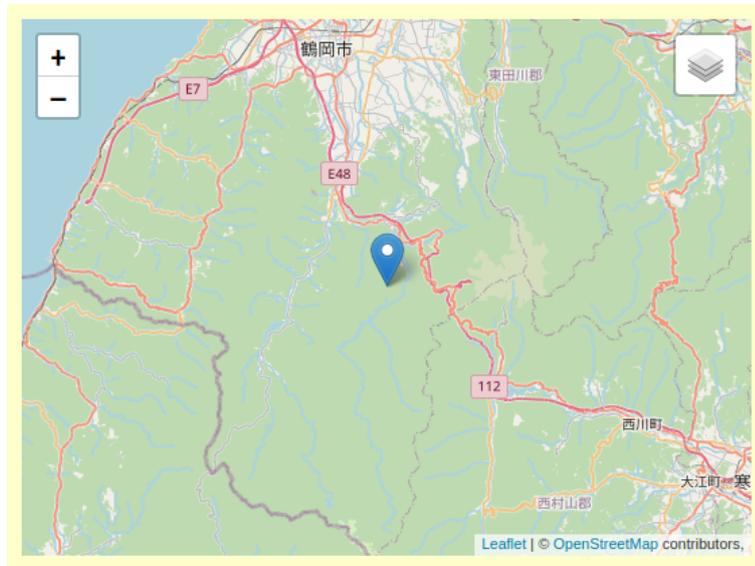


図 5.1 Leaflet で生成したマップ

JavaScript Leaflet によるマップの生成

```
var editmap = L.map('editmap');
var mpoint = [38.5350, 139.8875];
editmap.setView(mpoint, 10);
// OpenStreetMap のタイルを設定
var osmTile = L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="http://openstreetmap.org">
    OpenStreetMap</a> contributors,'
});
// 国土地理院のタイルを設定
var gsiTile = L.tileLayer('//cyberjapandata.gsi.go.jp/xyz/std/{z}/{x}/{y}.png', {
  attribution: '<a href="http://maps.gsi.go.jp/development/ichiran.html">
    国土地理院</a>'
});
var marker = L.marker(mpoint).addTo(editmap);
osmTile.addTo(editmap);
var baseLayers = {'OpenStreetMap': osmTile, '国土地理院': gsiTile};
L.control.layers(baseLayers, null).addTo(editmap);
```

5.2 緯度と経度の取得

.latlng.lat でマーカーがある地点の緯度, .latlng.lng で経度を取得する。その緯度と経度を CSS で非表示にした input 要素に代入し, 入力フォームでの送信時に参照できるようにする。これらの操作を click イベントにより, マップ上をクリックすることで行う (図.5.2)。



図 5.2 緯度と経度を取得するマーカー

JavaScript 緯度と経度を取得するマーカーの設置

```
function moveIcon(e) {  
  // クリックした地点の緯度と経度を取得  
  var clicklat = e.latlng.lat;  
  var clicklng = e.latlng.lng;  
  marker.setLatLng(e.latlng).bindPopup(  
    "緯度：" + clicklat + "<br>経度：" + clicklng  
  ).openPopup();  
  // id が lat の値に緯度、id が lng の値に経度を代入  
  document.getElementById("lat").value = clicklat;  
  document.getElementById("lng").value = clicklng;  
  
  editmap.on('click', moveIcon);  
}
```

5.3 Google Apps Script による入力フォームの生成

名称、概要、住所、Web サイトの URL、Google Drive の画像の共有 URL、問題文、選択肢 A、選択肢 B、選択肢 C、解答を入力・選択する欄を設ける。そしてその下に入力内容の送信ボタンとリセットボタンを配置する。Google Apps Script で取得し、(図.5.3)。



The image shows a web form with the following elements:

- 名称:
- 概要:
- 住所:
- WebサイトのURL:
- 画像:
- 問題文:
- 選択肢A:
- 選択肢B:
- 選択肢C:
- 正解となる解答:
- Buttons:

図 5.3 名称や問題文などの入力欄

GAS 入力した内容を Google スプレッドシートへ追加

```
function doPost(e){
    var sheet_url = "Google スプレッドシートの URL";
    var sh = SpreadsheetApp.openByUrl(sheet_url);
    //input 要素の内容を取ってくる
    var name = e.parameters.name.toString();
    var lat = e.parameters.lat.toString();
    var lng = e.parameters.lng.toString();
    var description = e.parameters.description.toString();
    var address = e.parameters.address.toString();
    var url = e.parameters.url.toString();
    var quiz = e.parameters.quiz.toString();
    var choiceA = e.parameters.choiceA.toString();
    var choiceB = e.parameters.choiceB.toString();
    var choiceC = e.parameters.choiceC.toString();
    var answer = String(e.parameters.answer);
    var image = e.parameters.image.toString();
    //これらを配列に入れ, Google スプレッドシートの最終行に追加
    var array = [name, lat, lng, description, address, url, quiz,
    choiceA, choiceB, choiceC, answer, image.replace("file/d/",
    "uc?id=").replace("\view?usp=sharing", "")]; //img 要素用の URL に変換
    sh.appendRow(array);
    //送信後に返されるページ
    var resultpage = HtmlService.createTemplateFromFile("result");
    return resultpage.evaluate();
}
```

送信ボタンがクリックされた後に返されるページを同じ Google Apps Script 内に作成する (図.5.4)。

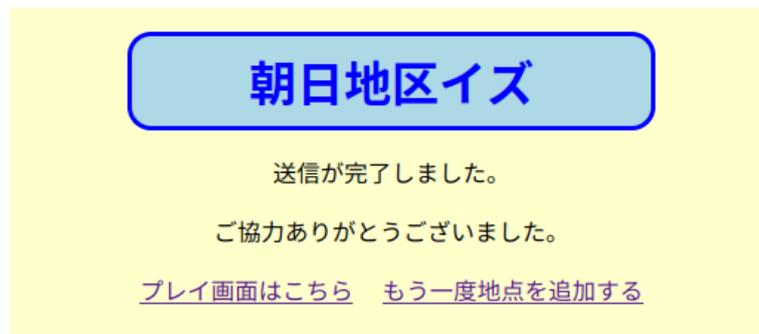


図 5.4 送信完了ページ

HTML 送信完了ページ

```

<!DOCTYPE html>
<html lang="ja">
<html>
  <head>
    <base target="_top">
    <title>送信完了 | 朝日地区イズ</title>
  </head>
  <body>
    <h1>朝日地区イズ</h1>
    <p>送信が完了しました。</p>
    <p>ご協力ありがとうございました。</p>
    <p><a href="プレイ画面の URL">プレイ画面はこちら</a>
      <a href="編集画面の URL">もう一度地点を追加する</a></p>
  </body>
</html>

```

5.4 Google スプレッドシートの取得と反映

左側のマップにマーカーを設置し、クリックすることで情報がポップアップで表示されるようにする。右側には地点の画像、クイズの問題文と選択肢のボタンを設ける。正解であれば赤文字で「正解！」と表示させてマーカーの色を赤に変え、不正解であれば青文字で「不正解…」と表示させる。(図.5.5)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	name	lat	lng	description	address	url	quiz	choiceA	choiceB	choiceC	answer	image	
2	月山ダム	38.583247	139.895997	月山ダムは、山	〒997-0405 山形	http://www.thr.m	月山ダムは、庄	梵字川	内川	大山川	a		
3	行沢とちの宴会	38.586912	139.82951	行沢地区には昔	〒997-0403 山形	https://www.asai	収穫したトノ	塩水にさらす	木灰で煮る	お酢でしめる	b		
4	庄内たがわ農業!	38.587097	139.874625	廃道となった国	〒997-0403 山形	https://mokeda	月山ワインの原	ほしぶどう	やまぶどう	うみぶどう	b		
5	松ヶ岡農場	38.692536	139.883058	松ヶ岡農場は、	〒997-0158 山形	http://matsuga	全農庄内の庄内	柿しぐれ	柿あられ	柿みぞれ	a		
6	大鳥池	38.362085	139.829049	朝日連峰の尾根	〒997-0622 山形	https://yamagata	朝日連峰・以東	イケタロウ	タキタロウ	カワタロウ	b		
7	湯殿山注連寺	38.602669	139.887049	湯殿山注連寺は	〒997-0531 山形	http://www.2.plala	七五三掛桜は5月	50年	200年	1000年	b		
8	湯殿山スキー場	38.572168	139.94197	雪質に優れた庄	〒997-0531 山形	https://www.tsuri	湯殿山スキー場	500m	1000m	2000m	c		
9	湯殿山総本寺 大	38.592548	139.900262	弘法大師が開山	〒997-0531 山形	https://www.tsuri	庄内地域には6位	真如海上人	本明海上人	鉄竜海上人	a		
10	菅原精肉店	38.611694	139.837407	先代から引き継	〒997-0404 山形	https://sugawara	朝日の人にとッ	チーズ味	キムチ味	カレー味	c		
11	湯殿山神社	38.541723	139.985616	「語るなかれ」	〒997-0532 山形	https://www.tsuri	湯殿山神社本宮	草履を履く	草履を履く	裸足になる	c		
12	小学校										a		
13	小学校										a		
14	小学校										a		
15	あさひ	38.84264076	139.8727772								a		
16	あさひ	38.84264076	139.8727772								a		

図 5.5 Google スプレッドシート

GAS Google スプレッドシートの情報の取得

```
const SPREAD_SHEET_ID = '18th7jHfAqERhzeT7vD1ztJlvUNzJMTSzWnByLiNdwh8';
const SHEET_NAME = 'sheet1';
function doGet(e) {
    const app = SpreadsheetApp.openById(SPREAD_SHEET_ID); //IDを取得
    const sheet = app.getSheetByName(SHEET_NAME); //シート名を取得
    const values = sheet.getDataRange().getValues(); //入力内容を全て配列で取得
    const data = [];
    //シートの入力内容をオブジェクトに詰め替え
    for(let i=0; i<values.length; i++){
        if(i === 0)continue; //ヘッダー部(1行目)はスキップ
        const param = {};
        for(let j=0; j<values[i].length; j++){
            param[values[0][j]] = values[i][j];
        }
        data.push(param);
    }
    const result = ContentService.createTextOutput(); //返却情報を生成
    result.setMimeType(ContentService.MimeType.JSON);
    result.setContent(JSON.stringify(data)); //JSON テキストをセット
    return result;
}
```

JavaScript 取得した情報をマップに反映

```
var marker = [];  
  
const endpoint = "Google Apps Script のデプロイした URL";  
fetch(endpoint)  
.then(response => response.json())  
// 成功した処理  
.then(data => {  
  // JSON から配列に変換  
  const markerList = data;  
  for (var i in markerList) {  
    var mk = markerList[i];  
    // スプレッドシートのデータを変数に代入  
    let name = mk.name, lat = mk.lat, lng = mk.lng,  
        description = mk.description, address = mk.address, url = mk.url,  
        image = mk.image, quiz = mk.quiz, choiceA = mk.choiceA,  
        choiceB = mk.choiceB, choiceC = mk.choiceC, answer = mk.answer;  
    let num = i;    // 問題番号  
    // ポップアップに表示させる内容  
    let popup = "<h3>" + name + "</h3><p>" + description  
        + "</p><p>" + address + "</p><a href='" + url + "'>" + url + "</a>"  
  }  
});
```

5.5 クイズの内容の切り替え

JavaScript クリックしたマーカーの情報への切り替え

```
function clickMarker(image, quiz, a, b, c, ans, num){  
  // 画像や問題文などの HTML の中身を変更  
  document.getElementById("image").setAttribute("src", image);  
  document.getElementById("quiz").innerHTML = quiz;  
  document.getElementById("a").innerHTML = a;  
  document.getElementById("b").innerHTML = b;  
  document.getElementById("c").innerHTML = c;  
  document.getElementById("result").innerHTML = " ";  
  document.getElementById("answer").innerHTML = ans;  
  document.getElementById("num").innerHTML = num;  
}
```

5.6 解答の判定

JavaScript 解答の判定

```
function judgeAnswer(abc) {
  // 選択肢と解答と問題番号を取得
  var select = document.getElementById(abc).value;
  var answer = document.getElementById("answer").textContent;
  var num = document.getElementById("num").textContent;
  // 選んだ選択肢と解答が正しいとき
  if (select == answer) {
    document.getElementById("result").innerHTML = "正解!!";
    document.getElementById("result").setAttribute("class", "correct");
    // アイコンの色を赤に変更
    marker[num].setIcon(redIcon);
    // ローカルストレージで問題番号に応じた変数に true
    localStorage.setItem('judge${num}', true);
  } else {
    document.getElementById("result").innerHTML = "不正解...";
    document.getElementById("result").setAttribute("class", "incorrect");
  }
}
```

画面右上に画像が切り替わるスペースを設ける。その下にクイズの問題文と選択肢を書き、正誤の判定が出るようにする。

JavaScript 解答の選択と解答状況のリセット

```
// 選択肢 A・B・C がそれぞれクリックされたとき
document.getElementById("a").addEventListener("click", () => {
  judgeAnswer("a");
});
document.getElementById("b").addEventListener("click", () => {
  judgeAnswer("b");
});
document.getElementById("c").addEventListener("click", () => {
  judgeAnswer("c");
});
```

5.7 アイコンの色変更

JavaScript 解答状況に応じた色変更

```
// ローカルストレージの judgeX の値が true のときは、赤いアイコンを設定
if(localStorage.getItem('judge${num}')=="true"){
    marker.push(L.marker([lat, lng], {title: name}).bindPopup(popup)
    .setIcon(redIcon).addTo(map)
    .addEventListener("click", () => {
        clickMarker(image, quiz, choiceA, choiceB, choiceC, answer, num);
    }));
// ローカルストレージの judgeX 自体が存在しないときは、青いアイコンのまま
}else{
    marker.push(L.marker([lat, lng], {title: name}).bindPopup(popup).addTo(map)
    .addEventListener("click", () => {
        clickMarker(image, quiz, choiceA, choiceB, choiceC, answer, num);
    }));
}
```

JavaScript ローカルストレージのデータクリア

```
document.getElementById("resetButton").addEventListener("click", () => {
    // ローカルストレージ全体のデータ削除
    localStorage.clear();
    // マーカーの色を全て青に変更
    for (var i in marker) {
        marker[i].setIcon(blueIcon);
    }
});
```


第 6 章

実験

本章では作成したシステムを使って実験を行い、名称や問題文などの情報が正確に追加されマップとクイズが機能するか検証する。実験の概要と手順を以下にまとめ、その実験結果から専門的な知識がない人でも扱える機能性と、容易に編集することができる利便性について考察する。

6.1 実験の概要

実験では、地点のクリックや内容の入力と選択によって、専門的な知識がない人でも容易に編集できるかどうか検証する。被験者が追加する項目は、地点の緯度・経度・名称・概要・住所・Web サイトの URL・画像、そしてクイズの問題文・選択肢 3 つ・正解となる解答である。以下にマップとクイズの追加手順を示す。

1. マップ上の追加したい地点をクリックして緯度と経度を取得する。
2. 名称・概要・住所・Web サイトの URL を入力する。
3. Google Drive の画像の共有 URL を入力する。
4. クイズの問題文と選択肢 A～C を入力する。
5. 正解となる解答を選択肢 A～C のの中から選択する。
6. 「地点を追加する」ボタンをクリックして内容を送信する。

6.2 実験結果

朝日中学校の親子行事に参加した生徒 25 名にお願いし、5 地点追加してもらうことができた。しかし、「小学校」が 3 つと「あさひ」が 2 つでいくつか重複し、クイズの問題文や選択肢が入力されていなかった。その内 3 地点は緯度・経度が入力されておらず、地図への反映が行われていなかった。

また、かたくり温泉の市場のあさひマルシェに来たお客さんをお願いし、あさひ小学校とかたくり温泉ほんぼの 2 地点してもらうことができた。こちらも緯度・経度と名称のみで、問題文や選択肢が入力されていなかった。

鶴岡サイエンスパークまつりでは、ジュニアドクター鳥海塾の第二段階の方にひまわり温泉ゆらら、大蔵村、の地点を入力してもらった。

	名称	緯度	経度
1 地点目	小学校		
2 地点目	あさひ	38.8426407646932	139.87277717369
3 地点目	あさひ小学校	38.656024370347	139.887572710567
4 地点目	かたくり温泉ぼんぼ	38.6341537844051	139.849672550979
5 地点目	ひまわり温泉ゆらら	38.3412859362341	140.275502800941
6 地点目	大蔵村	38.7539112048661	140.200652419026

6.3 考察

実験結果から考察する。

第7章

結論

選択した地点と入力した問題の内容を反映させることはできたが、保存ができていないのが現状である。今後はその情報をデータベースで管理し、正解した問題の状況も保存可能にする必要がある。その結果、朝日地域への興味と関心を向上させ、観光振興に繋がるシステムになるよう進めていく。

謝辞

本研究を進めるにあたり、多くの方々にご支援いただきました。指導教官の広瀬雄二教授からは多くのご指導ご協力を賜りました。ここに深謝の意を表します。また、実験にご協力いただいた朝日中央コミュニティセンターの皆様におきましては、業務に専念している中で大変貴重なお時間を頂戴することとなり、お詫びと共にご協力いただいたことに感謝いたします。

参考文献

- [1] 鶴岡市. “朝日地域振興計画”.
https://www.city.tsuruoka.lg.jp/shisei/sogokeikaku/tiikinosinkou/tiiki01o.files/shinko_asahi.pdf,
(参照 2023-05-09).
- [2] 佐藤滋・志村秀明・内田奈芳美・饗庭伸・川原晋・真野洋介・有賀隆. まちづくりデザインゲーム. 株式会社 学芸出版社. 2005, p.09-10, 76-77.
- [3] 観光庁. “観光 DX (デジタルトランスフォーメーション) の推進”.
https://www.mlit.go.jp/kankocho/shisaku/kankochi/digital_transformation.html, (参照 2023-01-10).
- [4] 王怡青・土井俊弥・井上祐輔・宇津呂武仁. “ご当地グルメを題材とするクイズ・コンテンツの作成”.
<https://db-event.jpn.org/deim2016/papers/185.pdf>, (参照 2022-11-15).
- [5] 明石拓弥・大浦秀喜・大藺隼人・成松智輝・山名莉央・酒匂大輝・中井哲也・中村匡秀. “ゲーミフィケーションに基づく地域理解動機付けのためのクイズラリープラットフォームの試作”.
<https://cs27.org/achieve/data/pdf/1450.pdf>, (参照 2023-11-14).
- [6] learningBOX 株式会社. “QuizGenerator — 無料で使えるクイズ・問題作成ツール”.
<https://quizgenerator.net/>, (参照 2023-08-19).
- [7] MDN. “HTML の基本 - ウェブ開発を学ぶ”.
https://developer.mozilla.org/ja/docs/Learn/Getting_started_with_the_web/HTML_basics,
(参照 2022-12-16).
- [8] MDN. “CSS: カスケーディングスタイルシート”.
<https://developer.mozilla.org/ja/docs/Web/CSS>, (参照 2022-12-16).
- [9] MDN. “JavaScript - ウェブ開発を学ぶ”.
https://developer.mozilla.org/ja/docs/Learn/Getting_started_with_the_web/JavaScript_basics, (参照 2022-12-16).
- [10] Volodymyr Agafonkin. “Leaflet - a JavaScript library for interactive maps”.
<https://leafletjs.com/>, (参照 2023-02-06).