

# GSImap 機能仕様書

---

## 1. プロジェクト概要

### 1.1 アプリケーション名

**GSImap** - 国土地理院地形図ビューア+画像オーバーレイツール

### 1.2 目的

国土地理院の地形図をベースに、PNG画像のオーバーレイ表示、GPSデータの可視化、ポイント・ルートデータの管理を行うWebベースの地図アプリケーション。

### 1.3 技術構成

- **フロントエンド**: HTML5, JavaScript (ES6+ Modules), CSS3
- **地図ライブラリ**: Leaflet.js v1.9.4
- **ファイル処理**: SheetJS (Excel), File API, FileReader API
- **地図タイル**: 国土地理院タイルサーバー
- **モジュール**: ネイティブESモジュール (import/export)
- **ブラウザ対応**: モダンブラウザ (Chrome, Firefox, Edge, Safari)

## 2. 主要機能

### 2.1 編集モード切り替えシステム

#### 2.1.1 画像・GPS編集モード

- **機能**: PNG画像オーバーレイ、GPSデータ表示、GeoJSON読み込み
- **対象ファイル**: PNG画像、Excel(.xlsx)、GeoJSON(.json)
- **操作**: 画像のドラッグ・リサイズ、透過度調整、座標設定

#### 2.1.2 ポイント・ルート編集モード

- **機能**: JSONファイルから既存のポイント・ルートデータを読み込み表示
- **対象ファイル**: ポイントJSON、ルートJSON
- **表示**: 座標データの地図上への可視化

### 2.2 レイアウトシステム

#### 2.2.1 オーバーレイレイアウト

- **構成**: 全画面地図表示、右上にフローティングコントロールパネル
- **背景**: 半透明白 (95%透明度) + ブラーエフェクト
- **モード切り替え**: ラジオボタンによる編集モード選択
- **レスポンシブ**: 画面サイズに応じた自動調整

### 2.3 画像オーバーレイ機能

### 2.3.1 画像読み込み・表示

- **対応形式:** PNG画像のみ
- **表示方法:** Leaflet ImageOverlayを使用した地図上重ね表示
- **初期配置:** 箕面大滝座標（34.853667, 135.472041）中心
- **アスペクト比:** 元画像の縦横比を維持

### 2.3.2 インタラクティブ編集

- **スケール調整:** 数値入力による表示倍率制御（0.1～）
- **透過度制御:** 0-100%のスライダー調整
- **中心座標設定:** 座標指定または地図クリックによる位置設定
- **ドラッグハンドル:** 四隅のリサイズハンドルによる画像サイズ調整
- **中心マーカー:** 画像中心のドラッグによる位置移動

## 2.4 GPS データ処理機能

### 2.4.1 Excel ファイル読み込み

- **対応形式:** .xlsx形式のExcelファイル
- **データ構造:** ID, 緯度(8桁DMS), 経度(9桁DMS)の3列構成
- **座標変換:** 度分秒形式から10進数形式への自動変換
- **フォーマット例:**
  - 緯度: 12345678 → 12°34'56.78"
  - 経度: 123456789 → 123°45'67.89"

### 2.4.2 GPS データ表示

- **マーカー:** 緑色円形マーカー（半径5px、緑枠2px）
- **ポップアップ:** ポイントID、座標値、元データ情報
- **自動フィット:** 全GPSポイントが表示されるよう地図範囲調整

## 2.5 GeoJSON 機能

### 2.5.1 ファイル読み込み

- **対応形式:** .geojson, json形式
- **処理方法:** JSON.parse()による解析後、L.geoJSON()で地図表示
- **エラーハンドリング:** 無効なGeoJSONファイルの検出と通知

### 2.5.2 表示スタイル

- **ライン/ポリゴン:** オレンジ色（#ff7800）、線幅3px、透明度0.8
- **ポイント:** 円形マーカー（半径6px、オレンジ塗り、黒枠1px）
- **プロパティ表示:** name属性がある場合はポップアップ表示

## 2.6 ポイント・ルート データ管理

### 2.6.1 ポイントJSON読み込み

- **データ構造:** totalPoints, imageInfo, points配列を含むJSON
- **座標系:** 画像絶対座標からLeaflet座標への変換
- **マーカー:** 赤色円形マーカー（半径4px、白枠1.5px）
- **ID表示:** ポイントIDをポップアップで表示

### 2.6.2 ルートJSON読み込み

- **データ構造:** routeInfo, imageInfo, points配列を含むJSON
- **中間点表示:** 青色円形マーカー（半径3px、白枠1px）
- **ルート描画:** 中間点を結ぶ青色ライン（線幅3px、透明度0.7）
- **情報表示:** 開始・終了ポイント、中間点数の情報

## 3. ユーザーインターフェース

### 3.1 コントロールパネル構成

#### 3.1.1 編集モード選択

- **位置:** オーバーレイパネル上部
- **形式:** ラジオボタン（「画像・GPS編集」「ポイント・ルート編集」）
- **動作:** 選択に応じたパネル内容の動的切り替え

#### 3.1.2 画像・GPS編集パネル

- **表示倍率:** number入力（0.1～、0.1刻み、初期値0.3）
- **透過度:** number入力（0-100%、初期値50%）
- **中心座標:** テキスト表示（緯度・経度、読み取り専用）
- **ファイルボタン:** 画像(PNG)読込、GPS値(Excel)読込、GeoJSON読込
- **中心座標:** 座標設定モード切り替えボタン

#### 3.1.3 ポイント・ルート編集パネル

- **ファイルボタン:** ポイントJSON読込、ルートJSON読込
- **シンプル構成:** 最小限のコントロール

### 3.2 視覚的フィードバック

#### 3.2.1 ドラッグハンドル

- **表示:** 四隅の赤色円形ハンドル（脈動アニメーション）
- **ホバー効果:** スケール1.3倍、赤色発光エフェクト
- **ツールチップ:** 各角の機能説明（「左上角をドラッグしてリサイズ」等）
- **カーソル:** リサイズ方向に応じたカーソル変更

#### 3.2.2 情報表示

- **リサイズ情報:** ドラッグ中の幅・高さ表示（km単位）
- **座標更新:** リアルタイムの座標値更新
- **エラーメッセージ:** モーダルダイアログによる統一された通知

### 3.3 スタイリング仕様

- **フォント**: システムフォント、13px基本サイズ
- **パネル背景**: rgba(255, 255, 255, 0.95) + backdrop-filter: blur(10px)
- **ボーダー**: 8px角丸、box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15)
- **レスポンシブ**: min-width: 240px, max-width: 300px

## 4. 技術仕様

### 4.1 モジュール構成

```
js/
├─ app-main.js           # メインアプリケーション統合
├─ map-core.js           # Leaflet地図コア機能
├─ image-overlay.js      # 画像オーバーレイ+ドラッグ機能
├─ gps-data.js           # GPSデータ処理 (Excel読み込み)
├─ geojson-loader.js     # GeoJSON読み込み機能
├─ point-route-editor.js # ポイント・ルート編集機能
└─ mode-switcher.js     # モード切り替え制御
```

### 4.2 主要クラス・メソッド

#### 4.2.1 GSIMapApp (統合クラス)

- **init()**: 全モジュールの初期化と連携設定
- **setupEventHandlers()**: ファイル入力イベントの統合管理
- **showErrorMessage()**: 統一エラー表示システム

#### 4.2.2 MapCore (地図コア)

- **constructor()**: Leaflet地図初期化、国土地理院タイル設定
- **getMap()**: 地図インスタンスの提供
- **専用ペイン作成**: dragHandles (z-index: 650) , centerMarker (z-index: 700)

#### 4.2.3 ImageOverlay (画像処理)

- **loadImage()**: PNG画像読み込み (Promise対応)
- **createDragHandles()**: リサイズハンドル生成・配置
- **updateImageBounds()**: アスペクト比維持リサイズ処理
- **moveImageToPosition()**: 中心座標による画像移動

#### 4.2.4 GPSData (GPS処理)

- **loadGPSData()**: Excelファイル解析 (SheetJS使用)
- **parseCoordinate()**: DMS→10進数変換 (8桁緯度・9桁経度対応)
- **addGPSMarkersToMap()**: GPSポイントマーカー配置

#### 4.2.5 ModeSwitcher (モード制御)

- **switchMode()**: 編集モード切り替え処理
- **showCurrentModePanel()**: パネル表示状態管理

4.3 座標系仕様

4.3.1 基準座標系

- **地理座標系**: WGS84 (EPSG:4326)
- **投影法**: Web Mercator (EPSG:3857) - Leafletデフォルト
- **初期中心**: 34.853667°N, 135.472041°E (箕面大滝)

4.3.2 座標変換処理

- **DMS形式**: 度分秒の固定長文字列解析
- **精度**: 小数点以下6桁 (約1m精度)
- **画像座標**: Canvas座標からLeaflet座標への変換

4.4 ファイル処理仕様

4.4.1 サポート形式

- **画像**: PNG形式のみ (JPEG, GIF等是非対応)
- **GPS**: Excel .xlsx形式 (.xls, CSV等是非対応)
- **地理**: GeoJSON形式 (.geojson, json拡張子)
- **座標**: ポイント・ルート専用JSON形式

4.4.2 処理制限

- **ファイルサイズ**: ブラウザメモリ制限内
- **セキュリティ**: すべてクライアントサイド処理
- **並行処理**: 同時ファイル読み込み非対応

5. データ仕様

5.1 GPSデータ構造 (Excel)

列番号	データ型	説明	例
A	文字列	ポイントID	"P001"
B	文字列(8桁)	緯度(DDMMSS.ss)	"34512345"
C	文字列(9桁)	経度(DDDMMSS.ss)	"135283456"

5.2 ポイントJSONデータ構造

```
{
  "totalPoints": 10,
  "imageInfo": {
    "width": 1920,
```

```
    "height": 1080
  },
  "points": [
    {
      "index": 1,
      "id": "A001",
      "x": 640,
      "y": 480,
      "isMarker": false
    }
  ],
  "exportedAt": "2025-08-10T12:34:56.789Z"
}
```

### 5.3 ルートJSONデータ構造

```
{
  "routeInfo": {
    "startPoint": "START",
    "endPoint": "GOAL",
    "waypointCount": 5
  },
  "imageInfo": {
    "width": 1920,
    "height": 1080
  },
  "points": [
    {
      "type": "waypoint",
      "index": 1,
      "x": 320,
      "y": 240
    }
  ],
  "exportedAt": "2025-08-10T12:34:56.789Z"
}
```

## 6. 動作環境

### 6.1 推奨環境

- **OS**: Windows 10/11, macOS 10.15+, Ubuntu 18.04+
- **ブラウザ**: Chrome 90+, Firefox 88+, Edge 90+, Safari 14+
- **画面解像度**: 1024×768以上推奨
- **メモリ**: 4GB以上推奨（大容量画像処理時）

### 6.2 必要機能

- **ES6 Modules**: import/export構文サポート

- **File API:** FileReader, Blob API
- **Canvas API:** 画像処理用
- **CORS:** ローカルファイルアクセス用

## 6.3 開発環境

- **サーバー:** 静的ファイルサーバー (`python -m http.server`等)
- **デバッグ:** ブラウザ開発者ツール
- **バージョン管理:** Git推奨

# 7. エラーハンドリング

## 7.1 ファイル読み込みエラー

- **画像形式エラー:** PNG以外選択時の警告
- **ファイル破損:** 読み込み失敗時のエラー表示
- **GPS形式エラー:** 座標解析失敗時の詳細情報
- **JSON形式エラー:** 不正なJSON構造の検出

## 7.2 操作エラー

- **モジュール読み込み:** ESモジュール読み込み失敗
- **座標変換:** 無効な座標値の処理
- **メモリ不足:** 大容量ファイル処理時の警告

## 7.3 表示方式

- **統一UI:** モーダルダイアログによる一貫したエラー表示
- **詳細情報:** エラーメッセージと解決方法の提示
- **ログ出力:** Console.logによる開発者向け詳細情報

# 8. 今後の拡張可能性

## 8.1 機能拡張案

- **ポイント編集:** 地図上でのポイント作成・編集機能
- **ルート作成:** ドラッグによるルート描画機能
- **データ出力:** 編集したデータのJSON/GPX出力
- **印刷機能:** 地図とオーバーレイの印刷対応
- **測定機能:** 距離・面積計算機能

## 8.2 技術改善案

- **TypeScript化:** 型安全性の向上
- **WebWorker:** 大容量ファイル処理の非同期化
- **PWA対応:** オフライン利用とインストール機能
- **データベース:** IndexedDBによるデータ永続化
- **クラウド連携:** オンラインストレージとの連携

# 9. 制約事項

## 9.1 技術的制約

- **ブラウザ依存:** ES6 Modules対応ブラウザが必須
- **ローカル実行:** HTTPSまたはローカルサーバーでの動作が必要
- **メモリ制限:** 大容量画像処理時のブラウザメモリ制限
- **同期処理:** ファイル読み込みの逐次処理

## 9.2 機能制約

- **画像形式:** PNG形式のみサポート
- **座標系:** 日本国内の座標系に最適化
- **データ永続化:** ブラウザセッション終了時にデータ消失
- **多言語:** 日本語UIのみ対応

---

**作成日:** 2025年8月10日

**バージョン:** 2.0

**更新内容:** モジュール化アーキテクチャ、編集モード切り替え、オーバーレイレイアウト対応

**作成者:** Claude Code