

WebGISアプリ向けデータ可視化機能

詳細設計書

データ取得変換ツール

1. 地理空間情報データ変換ツール(GeoJSONからMVTへの変換)

概要

収集した国土数値情報のGeoJSONファイルをMVT形式のファイルに変換する

内容

- 1.引数を受け取る
- 2.引数の数が3であるとき、
tippecanoe -e [引数1] -pC -z[引数2] -aN [引数3]
を実行する
- 3.引数の数が3でないとき
Usage: data_preparation_tool_mvt.sh dst_dir max_zoom_level src_geojson
を表示する

シェルスクリプトファイル名

data_preparation_tool_mvt.sh

引数

引数1	dst_dir	string	MVTを出力したいフォルダ名のパス
引数2	max_zoom_level	string	作成したい最大のズームレベル
引数3	src_geojson	string	MVTファイルに変更したいGeoJSONデータのパス

返り値

なし

2.3次元建物データ変換ツールの利用(GeoJSONからMVTへの変換)

概要

- 1.引数を受け取る
- 2.引数の数が4であるとき、
tippecanoe -pC -ad -an -ps -z[引数1] -e [引数2] -l [引数3] -ai [引数4]
を実行する
- 3.引数の数が4でないとき
Usage: data_preparation_tool_3d_mvt.sh max_zoom_level dst_dir layer_name src_geojson
を表示する

シェルスクリプトファイル名

data_preparation_tool_mvt.sh

引数

引数1	max_zoom_level	string	作成したい最大のズームレベル
引数2	dst_dir	string	MVTを出力したいフォルダ名のパス
引数3	layer_name	string	レイヤ名
引数4	src_geojson	string	MVTファイルに変更したいGeoJSONデータのパス

返り値

なし

WebAPI_アメダスデータ

概要

リクエストパラメータに記入された地図の領域と日時に該当するアメダスデータを取得し、GeoJsonとして返却する機能。

基本情報

名称	アメダスデータ
APIパス	amedasData
公開URL	/api/t_amedas_data
レスポンス形式	GeoJson

引数（リクエストパラメータ）

変数名	説明	フォーマット	必須	例
point_1	領域における北西の位置情報	カンマ区切りの数値（経度,緯度）	●	139.4,35.7
point_2	領域における北東の位置情報	カンマ区切りの数値（経度,緯度）	●	139.5,35.75
point_3	領域における南東の位置情報	カンマ区切りの数値（経度,緯度）	●	139.6,35.8
point_4	領域における南西の位置情報	カンマ区切りの数値（経度,緯度）	●	139.3,35.85
currentDate	対象とする年月日時分	YYYYMMDDHH24MI	●	202201271512 ⇒15時10分のデータ取得

処理詳細（データ取得）

データ取得元テーブル	t_amedas_code_data, t_amedas_data
取得処理	下記は同一クエリ文ではなく、それぞれ順次実行する。 1. t_amedas_code_dataテーブルのlocationがリクエストパラメータのpoint_1,point_2,point_3,point_4で囲まれた領域内（CRSは4326）であるデータを取得。 2. t_amedas_dataテーブルのprefNumberが、上記1. で取得したprefNumberであり、かつdatetimeがリクエストパラメータのcurrentDateと一致するデータを取得 ・datetimeは10分刻であるため、currentDateの最後の1桁目を無視し0に変換して等号させる

レスポンス(JSON)

・正常時

項目名	内容																																																																						
type	"FeatureCollection"																																																																						
features	子項目名	データ内容																																																																					
	geometry	t_amedas_code_dataテーブルのlocation ・経度,緯度に入れ替えて出力 ・CRSは4326																																																																					
	properties	<table><tr><th>子項目名</th><th>データ内容</th></tr><tr><td>prefNumber</td><td>t_amedas_code_dataテーブルの地域コード</td></tr><tr><td>altitude</td><td>t_amedas_code_dataテーブルの高度</td></tr><tr><td>type</td><td>t_amedas_code_dataテーブルの種別</td></tr><tr><td>elems</td><td>t_amedas_code_dataテーブルの属性値</td></tr><tr><td>kjName</td><td>t_amedas_code_dataテーブルの漢字名</td></tr><tr><td>knName</td><td>t_amedas_code_dataテーブルのカタカナ名</td></tr><tr><td>enName</td><td>t_amedas_code_dataテーブルの英名</td></tr><tr><td>datetime</td><td>t_amedas_dataテーブルのデータ取得日時</td></tr><tr><td>observationNumber</td><td>t_amedas_dataテーブルの観測値コード</td></tr><tr><td>pressure</td><td>t_amedas_dataテーブルの現地気圧</td></tr><tr><td>normalPressure</td><td>t_amedas_dataアールの海面更正気圧</td></tr><tr><td>temp</td><td>t_amedas_dataアールの気温</td></tr><tr><td>humidity</td><td>t_amedas_dataアールの湿度</td></tr><tr><td>sun10m</td><td>t_amedas_dataアールの 1 0 分間日照時間</td></tr><tr><td>sun1h</td><td>t_amedas_dataアールの 1 時間日照時間</td></tr><tr><td>precipitation10m</td><td>t_amedas_dataアールの前 1 0 分間降水量</td></tr><tr><td>precipitation1h</td><td>t_amedas_dataアールの前 1 時間降水量</td></tr><tr><td>precipitation3h</td><td>t_amedas_dataアールの前 3 時間降水量</td></tr><tr><td>precipitation24h</td><td>t_amedas_dataアールの前 2 4 時間降水量</td></tr><tr><td>windDirection</td><td>t_amedas_dataアールの風向</td></tr><tr><td>wind</td><td>t_amedas_dataアールの風速</td></tr><tr><td>maxTempTime</td><td>t_amedas_dataアールのこの時間までの日最高気温が出た時刻</td></tr><tr><td>maxTemp</td><td>t_amedas_dataアールのこの時間までの日最高気温</td></tr><tr><td>minTempTime</td><td>t_amedas_dataアールのこの時間までの日最低気温が出た時刻</td></tr><tr><td>minTemp</td><td>t_amedas_dataアールのこの時間までの日最低気温</td></tr><tr><td>gustTime</td><td>t_amedas_dataアールの日最大風の時刻</td></tr><tr><td>gustDirection</td><td>t_amedas_dataアールの日最大風速の風向</td></tr><tr><td>gust</td><td>t_amedas_dataアールの日最大風速</td></tr><tr><td>snow</td><td>t_amedas_dataアールの積雪深</td></tr><tr><td>snow1h</td><td>t_amedas_dataアールの 1 時間降雪量</td></tr><tr><td>snow3h</td><td>t_amedas_dataアールの 3 時間降雪量</td></tr><tr><td>snow6h</td><td>t_amedas_dataアールの 6 時間降雪量</td></tr><tr><td>snow12h</td><td>t_amedas_dataアールの 1 2 時間降雪量</td></tr><tr><td>snow24h</td><td>t_amedas_dataアールの 2 4 時間降雪量</td></tr></table>	子項目名	データ内容	prefNumber	t_amedas_code_dataテーブルの地域コード	altitude	t_amedas_code_dataテーブルの高度	type	t_amedas_code_dataテーブルの種別	elems	t_amedas_code_dataテーブルの属性値	kjName	t_amedas_code_dataテーブルの漢字名	knName	t_amedas_code_dataテーブルのカタカナ名	enName	t_amedas_code_dataテーブルの英名	datetime	t_amedas_dataテーブルのデータ取得日時	observationNumber	t_amedas_dataテーブルの観測値コード	pressure	t_amedas_dataテーブルの現地気圧	normalPressure	t_amedas_dataアールの海面更正気圧	temp	t_amedas_dataアールの気温	humidity	t_amedas_dataアールの湿度	sun10m	t_amedas_dataアールの 1 0 分間日照時間	sun1h	t_amedas_dataアールの 1 時間日照時間	precipitation10m	t_amedas_dataアールの前 1 0 分間降水量	precipitation1h	t_amedas_dataアールの前 1 時間降水量	precipitation3h	t_amedas_dataアールの前 3 時間降水量	precipitation24h	t_amedas_dataアールの前 2 4 時間降水量	windDirection	t_amedas_dataアールの風向	wind	t_amedas_dataアールの風速	maxTempTime	t_amedas_dataアールのこの時間までの日最高気温が出た時刻	maxTemp	t_amedas_dataアールのこの時間までの日最高気温	minTempTime	t_amedas_dataアールのこの時間までの日最低気温が出た時刻	minTemp	t_amedas_dataアールのこの時間までの日最低気温	gustTime	t_amedas_dataアールの日最大風の時刻	gustDirection	t_amedas_dataアールの日最大風速の風向	gust	t_amedas_dataアールの日最大風速	snow	t_amedas_dataアールの積雪深	snow1h	t_amedas_dataアールの 1 時間降雪量	snow3h	t_amedas_dataアールの 3 時間降雪量	snow6h	t_amedas_dataアールの 6 時間降雪量	snow12h	t_amedas_dataアールの 1 2 時間降雪量	snow24h
子項目名	データ内容																																																																						
prefNumber	t_amedas_code_dataテーブルの地域コード																																																																						
altitude	t_amedas_code_dataテーブルの高度																																																																						
type	t_amedas_code_dataテーブルの種別																																																																						
elems	t_amedas_code_dataテーブルの属性値																																																																						
kjName	t_amedas_code_dataテーブルの漢字名																																																																						
knName	t_amedas_code_dataテーブルのカタカナ名																																																																						
enName	t_amedas_code_dataテーブルの英名																																																																						
datetime	t_amedas_dataテーブルのデータ取得日時																																																																						
observationNumber	t_amedas_dataテーブルの観測値コード																																																																						
pressure	t_amedas_dataテーブルの現地気圧																																																																						
normalPressure	t_amedas_dataアールの海面更正気圧																																																																						
temp	t_amedas_dataアールの気温																																																																						
humidity	t_amedas_dataアールの湿度																																																																						
sun10m	t_amedas_dataアールの 1 0 分間日照時間																																																																						
sun1h	t_amedas_dataアールの 1 時間日照時間																																																																						
precipitation10m	t_amedas_dataアールの前 1 0 分間降水量																																																																						
precipitation1h	t_amedas_dataアールの前 1 時間降水量																																																																						
precipitation3h	t_amedas_dataアールの前 3 時間降水量																																																																						
precipitation24h	t_amedas_dataアールの前 2 4 時間降水量																																																																						
windDirection	t_amedas_dataアールの風向																																																																						
wind	t_amedas_dataアールの風速																																																																						
maxTempTime	t_amedas_dataアールのこの時間までの日最高気温が出た時刻																																																																						
maxTemp	t_amedas_dataアールのこの時間までの日最高気温																																																																						
minTempTime	t_amedas_dataアールのこの時間までの日最低気温が出た時刻																																																																						
minTemp	t_amedas_dataアールのこの時間までの日最低気温																																																																						
gustTime	t_amedas_dataアールの日最大風の時刻																																																																						
gustDirection	t_amedas_dataアールの日最大風速の風向																																																																						
gust	t_amedas_dataアールの日最大風速																																																																						
snow	t_amedas_dataアールの積雪深																																																																						
snow1h	t_amedas_dataアールの 1 時間降雪量																																																																						
snow3h	t_amedas_dataアールの 3 時間降雪量																																																																						
snow6h	t_amedas_dataアールの 6 時間降雪量																																																																						
snow12h	t_amedas_dataアールの 1 2 時間降雪量																																																																						
snow24h	t_amedas_dataアールの 2 4 時間降雪量																																																																						

・リクエストパラメータエラーまたはDBエラーの場合

空欄のJSONを返却する

WebAPI_人流データ

概要

リクエストパラメータに記入された地図の領域と日時などに該当する人流データを取得し、GeoJsonとして返却する機能。

基本情報

名称	人流データ
APIパス	peopleFlowData
公開URL	/api/t_people_flow_data
レスポンス形式	GeoJson

引数（リクエストパラメータ）

変数名	説明	フォーマット	必須	例
point_1	領域における北西の位置情報	カンマ区切りの数値（経度,緯度）	●	139.4,35.7
point_2	領域における北東の位置情報	カンマ区切りの数値（経度,緯度）	●	139.5,35.75
point_3	領域における南東の位置情報	カンマ区切りの数値（経度,緯度）	●	139.6,35.8
point_4	領域における南西の位置情報	カンマ区切りの数値（経度,緯度）	●	139.3,35.85
currentDate	対象とする年月日時分秒	下記のように最小単位は「秒」で最大単位は「年月」 ----- YYYYMMDDHH24MISS, YYYYMMDDHH24MI, YYYYMMDDHH24, YYYYMMDD, YYYYMM	●	日付の場合：20180404
device	対象とする子機デバイスID	文字列（先頭ゼロも含める）	-	0011

処理詳細（データ取得）

データ取得元テーブル	t_people_flow_data
取得条件	下記はすべてAND条件 1. t_people_flow_dataテーブルのlocationをX,Yを入れ替えて生成したgeometryが、リクエストパラメータのpoint_1,point_2,point_3,point_4で囲まれた領域内である。（CRSは4326） 2. t_people_flow_dataテーブルのdatetimeがリクエストパラメータcurrentDateと一致する。 ただし、リクエストパラメータcurrentDateの桁数（文字列長さ）と、比較するdatetimeの桁数（左から）は同一にする。 3. リクエストパラメータdeviceが指定された場合、t_people_flow_dataテーブルのdeviceと一致する。

レスポンス(JSON)

・正常時

項目名	内容		
type	"FeatureCollection"		
features	子項目名	データ内容	
	geometry	t_people_flow_dataテーブルのlocation ・経度,緯度に入れ替えて出力 ・CRSは4326	
	properties	子項目名	データ内容
		id	t_people_flow_dataテーブルのid
		people_flow_base_id	t_people_flow_dataテーブルのpeople_flow_base_id
		device	t_people_flow_dataテーブルの子機デバイスID
		rsssi	t_people_flow_dataテーブルの受信強度
		datetime	t_people_flow_dataテーブルの観測日時

・リクエストパラメータエラーまたはDBエラーの場合

空欄のJSONを返却する

タイムスライダー機能

Webページに時間軸バーを組み込むことができるjQueryプラグイン

詳細は <http://k2go.jp/public/Timeline/> を参照のこと

1. タイムスライダーUI

タイムスライダーUIについて以下に示す。

- ・ 画面下部に横方向の時間軸バーおよび摘みを描画する。
- ・ 時間軸バーでホイールを操作すると時間軸バー目盛の縮尺を変更する。
- ・ 摘みを左右にドラッグすると現在日時を移動する。
- ・ 自動再生パネルの再生ボタンをクリックすると、自動再生開始時刻から、自動再生終了時刻まで、指定した時間間隔で自動的に時刻を移動する。自動再生に合わせてスライダーのつまみも移動する。

2. API

以下のI/Fを同期対象のアプリケーションに実装することで、アプリケーション間で情報を同期することができる。

引数の詳細は <http://k2go.jp/public/STARS/controller/> を参照。

create

説明

指定した日時情報で時間軸バーを再描画します。

構文

```
$(セレクト).k2goTimeline("create"[, 引数]);
```

戻り値

jqueryオブジェクト

zoomIn

説明

摘み（ポインタ）の位置を基点に時間軸バーをズームインします。

構文

```
$(セレクト).k2goTimeline("zoomIn");
```

戻り値

jqueryオブジェクト

zoomOut

説明

摘み（ポインタ）の位置を基点に時間軸バーをズームアウトします。

構文

```
$(セレクト).k2goTimeline("zoomOut");
```

戻り値

jqueryオブジェクト

showRangeBar

説明

指定した日時情報でレンジバーを描画します。

構文

```
$(セレクト).k2goTimeline("showRangeBar"[, 引数]);
```

戻り値

jqueryオブジェクト

hiddenRangeBar

説明

レンジバーを非表示にします。

構文

```
$(セレクト).k2goTimeline("hiddenRangeBar");
```

戻り値

jqueryオブジェクト

start

説明

時間軸バーの自動スクロールを開始します。

構文

`$(セレクト).k2goTimeline("start", 引数);`

戻り値

jqueryオブジェクト

stop

説明

時間軸バーの自動スクロールを停止します。

構文

`$(セレクト).k2goTimeline("stop");`

戻り値

jqueryオブジェクト

getOffsetFromTime

説明

日時から時間軸バー上のオフセット座標を取得します。

構文

`$(セレクト).k2goTimeline("getTimeFromOffset", 引数);`

戻り値

指定したオフセット座標（X座標）の日時（Dateオブジェクト）

getOffsetFromTime

説明

日時から時間軸バー上のオフセット座標を取得します。

構文

`$(セレクト).k2goTimeline("getOffsetFromTime", 引数);`

戻り値

指定した日時のオフセット座標（X座標）

setOptions

説明

オプションの設定変更を行います。

構文

`$(セレクト).k2goTimeline("setOptions", 引数);`

戻り値

jqueryオブジェクト

getOptions

説明

オプション値を取得します。

構文

`$(セレクト).k2goTimeline("getOptions");`

戻り値

現在のオプション値を含むObject

formatDate

説明

日時を指定した書式で文字列変換します。

構文

`$(セレクト).k2goTimeline("formatDate", 日時, 書式[, オフセット]);`

戻り値

日時文字列

ViewURL

表示している時空間情報及び表示データ情報をURL（URI）で表示（一時記録）できる機能。

1. WebGIS情報の一時保存機能

API

createURL(baseUrl, param)

baseUrl アプリ自身のURL

URL経由で新たに開くアプリに引き継ぐ情報。

param

param[key]=value の形式で設定する。

valueはencodeURIComponent()でURLとして利用可能な文字列に変換している。

また特定のkeyがある場合は、以下の通りに変換する。

current_time : cT

bar_left_time : bLT

bar_right_time : bRT

autorun_start : aS

autorun_end : aE

map_latitude : mLT

map_longitude : mLN

map_direction : mD

map_pitch : mP

map_zoom : mZ

window_top : wT

window_left : wL

window_width : wW

window_height : wH

引数から以下の文字列を生成する。

baseUrl?key1=value1&key2=value2& ... keyN=valueN

処理詳細

1. baseUrl(WebアプリのURL)のチェックを行う。

baseUrlがnull、もしくは空文字列の場合は "base_url is not defined."をコンソール出力し、エラーを投げる。

baseUrlの値がstringタイプでない場合は、 "base_url type is not string." をコンソール出力し、エラーを投げる。

2. param[key]=value について、keyとvalueを変換する。

keyが特定の文字列の場合は、対応する別の文字列に変換し、そうでなければencodeURIComponentでURLに利用可能な文字列に変換する。

valueはencodeURIComponentでURLに利用可能な文字列に変換する。

3. 2で変換したkey、valueから "key=value" の文字列に連結し、配列に格納する。

4. baseUrlと "key=value"形式の文字列から "baseUrl?key1=value1&key2=value2& ... keyN=valueN" の形式のURLを生成する。

5. URLが2083文字以内ならそれを返却する。2084文字以上なら "URL length is overflow."をコンソール出力し、エラーを投げる。

2. WebGIS情報の一時保存機能

API

parseURL(url)

url createURLで生成したURL文字列をパースし、param[key]=valueの形式で格納する。

valueはdecodeURIComponentで復元し、createURLで変換された特定のkeyも逆変換する。

処理詳細

1. urlのチェックを行う。

? が含まれていれば、?以降のクエリを & で分割し、配列に格納する。

? が含まれていない場合は、空の配列を返却する。

2. 1で配列に格納した文字列を、さらに = で分割する。

1番目の文字列が特定の文字列であれば変換し、それ以外であればdecodeURIComponentで変換する。

2番目の文字列はdecodeURIComponentで変換する。

3. 2で分割、変換したそれぞれの文字列について、1番目をkey、2番目をvalueとしparam[key]=valueの形式で格納する。

4. urlを最後まで処理したらparamを返却する。

時空間同期機能

複数のWebGISアプリケーションの時空間情報を同期させる機能

1. StarsControllerの同期処理

StarsControllerは、複数のWebアプリケーションを同期させるためのライブラリで、ライブラリを組み込んだアプリケーション（親アプリ）から、事前に同期用のI/Fを実装した同期対象のアプリケーション（子アプリ）を複数起動し、親アプリ側から同期用のI/Fを呼ぶことで、同期処理を実現する。以下、同期処理の概略を示す。

1. 親アプリを起動
2. 同期対象の子アプリをすべて起動
3. すべての子アプリが起動完了するまで、親アプリが監視
4. すべての子アプリの起動が完了次第、フォーカスが当たっている子アプリの時空間情報を、同期用のI/F経由で親アプリが収集
5. 親アプリが収集した情報を、同期用のI/F経由でフォーカスされていない子アプリへ配信

また、MapboxベースのWebGISアプリとiTownsbベースのWebGISアプリの間での同期処理では仰俯角とズームの扱いが異なるため、以下の通りの変換処理を実施している。

仰俯角：iTownsbの仰俯角 = $\text{Max}(90 - \text{Mapboxの仰俯角}, 60)$

ズーム率：iTownsbのズーム値(カメラから地表までの距離)： $80000000 * e^{(\text{Mapboxのズーム率} * -0.712)}$

2. 同期用API

以下のI/Fを同期対象のアプリケーションに実装することで、アプリケーション間で情報を同期することができる。

引数、戻り値などの詳細はStarsControllerの <http://k2go.jp/public/STARS/controller/> を参照。

STARScntroller_getDate

STARScntrollerがWebアプリケーションの現在日時情報を取得するため定期的にコールします。

戻り値としてWebアプリケーションの現在日時情報を返して下さい。

構文

```
STARScntroller_getDate()
```

引数

なし

戻り値

Webアプリケーションの現在日時情報を保持するJSONオブジェクト

STARScntroller_setDate

STARScntrollerがWebアプリケーションの現在日時を同期させる際にコールします。

当該メソッドがコールされたらWebアプリケーション側で日時変更に伴う必要な処理を行って下さい。

構文

```
STARScntroller_setDate(pDate)
```

引数

Webアプリケーションの現在日時情報を変更する為のJSONオブジェクト

戻り値

なし

STARScntroller_getPosition

STARScntrollerがWebアプリケーションの現在日時情報を取得するため定期的にコールします。

戻り値としてWebアプリケーションの現在日時情報を返して下さい。

構文

```
STARScntroller_getPosition()
```

引数

なし

戻り値

Webアプリケーションの現在位置情報を保持するJSONオブジェクト

STARScntroller_setPosition

STARScntrollerがWebアプリケーションの現在位置を同期させる際にコールします。

当該メソッドがコールされたらWebアプリケーション側で位置変更に伴う必要な処理を行って下さい。

構文

```
STARScntroller_setPosition(pPosition)
```

引数

Webアプリケーションの現在位置情報を保持するJSONオブジェクト

戻り値

なし

STARSController_isReady

Webアプリケーションが日時情報や位置情報の取得・設定が可能な状態かを判別する為、STARSControllerが定期的にコールします
戻り値としてWebアプリケーションの現在の状態を返して下さい。

構文

```
STARSController_isReady()
```

引数

なし

戻り値

Webアプリケーションの現在の状態を表す論理値 (true/false) 。

連続画像キャプチャ

データを時間方向に自動再生しながら、タイムラインとの画像を静止画像として複数枚キャプチャする機能。

1. キャプチャ処理詳細

MediaStream#getDisplayMediaを利用して、WebGISアプリケーションの表示画面をキャプチャする。

1. 連続画像キャプチャが可能かどうかの判定を時刻及び、コールバック関数の結果により決定する。
2. MediaStream#getDisplayMediaで画面キャプチャのStreamを取得する。
3. CanvasRenderingContext2D.drawImageでWebGISアプリの表示画面を画像オブジェクトとして作成する
4. HTMLCanvasElement.toBlobで画像をメモリに書き出し、URLを生成する。
5. 引数に設定されたダウンロード前処理を実行し、Navigator.msSaveBlobで指定のファイル名でキャプチャ画像をダウンロードする。
6. 1に戻り、キャプチャ停止条件に当てはまれば連続画像キャプチャを停止する。

2. API

multiCapture

連続画像キャプチャを行う

構文

multiCapture(currentTime, startTime, endTime, stepTime, plsContinueCallback, pPreCaptureCallback, pNameFormatCallback)

引数

currentTime	現在時刻
startTime	連続画像キャプチャの開始時刻
endTime	連続画像キャプチャの終了時刻
stepTime	連続画像キャプチャの一回のキャプチャごとに進む時間ステップ
plsContinueCallback	連続画像キャプチャを継続するかをチェックするためのコールバック関数
pPreCaptureCallback	画像キャプチャの前処理のためのコールバック関数
pNameFormatCallback	キャプチャ画像の名前を返却するコールバック関数

戻り値

なし

(ダウンロードフォルダにキャプチャ画像が格納される。)

360度画像保存

1.初期化处理

概要

360度画像作成に必要なDomを作成し、viewerを初期化する。

内容

- 1.ドキュメントのbodyにJPEGのエンコーダを追加する。
- 2.スフィアマップの画像レンダリング用のDomおよび、画像保存用のDomを作成し、ドキュメントに追加する。
また、各種ボタンと入力エリアを配置する。
- 3.キャプチャ取得用のサブビューの初期化を行う。
- 4.サブビューの更新を行う。
- 5.メインビューとサブビューを同期させる。
- 6.イベントの登録を行う。
- 3.スフィアマップのレンダラーを初期化する。

関数名

CreateSphereMap()

引数

GlobalView	view	メインビュー
Function	initBackGroundLayersFunc	サブビューの初期化時に呼び出される関数。引数としてサブビューのviewが渡される。
GuiTools	GUITools	指定した場合、指定したGUIToolがviewに反映される。

返り値

なし

2.360度画像用サブビューの表示/非表示切り替え機能

概要

キューブマップを作成しているサブビューの表示と非表示を切り替える。

内容

- 1.サブビューの描画領域を取得する。
- 2.表示の場合、サブビューの描画領域の透過率を1に変更する。
非表示の場合、サブビューの描画領域の透過率を0.00000000001に変更する。

関数名

toggleShowBackGroundView()

引数

なし

返り値

なし

3.360度画像度画像作成機能

概要

360度画像の作成を開始する。作成が終了すると、自動的にダウンロードを行う。

内容

- 非同期で以下の処理を実行する。
 - キューブマップの3Dカメラの角度を上ベクトルを軸にして回転する。
 - キューブマップの画像をキャプチャする。
 - 「360度画像用サブビュー作成の待ち時間」で設定されている時間後に次の処理を開始する。
- 手順1を「前」「後ろ」「右」「左」「上」「下」に対して行い、結合する。
- キャプチャした画像のダウンロードを行う。

関数名

downloadSphereMap()

引数

GlobalView	mainView	メインビュー
GlobalView	view	サブビュー

返り値

なし

4.360度画像の解像度設定機能

概要

解像度の設定を行う。単位は[px]で、最低1[px]、最大解像度は1024[px]とする。

内容

- 入力値が1未満の場合は1、1025以上の場合は1024に丸め込む。
- サブビューの更新を行う。

関数名

setDisplaySize()

引数

GlobalView	backGroundView	サブビュー
int	size	解像度

返り値

なし

5.360度画像用サブビュー作成の待ち時間設定機能

概要

キューブマップ作成時、現在の視点での読み込み予定のタイル画像がサーバ上に存在しない場合の、画像取得のタイムアウト時間を設定する。

内容

- タイムアウト値に入力値を設定する。

関数名

setTimeout()

引数

int	timeout	待ち時間
-----	---------	------

返り値

なし

点群表示機能

複数の地点に関する時系列データを点で地図上に可視化する機能

1. 点群表示機能の処理詳細

MapboxではDeck.glのMapboxLayerを利用して描画する。詳細を以下に記述する。

1. インプットデータとして、CSVファイルをD3.js利用で読み取る。
2. deck.glのMapboxLayerを利用して点群データ用のLayerを作成する。
3. 点群データ用のLayerを地図へ追加する。

更に時系列データ表示のために、以下の処理を行う。

0. 時系列データとして、対応する時刻をYYYYMMDDhhmmの形式にしたファイル名のデータを準備する
1. 表示済の点群データ用のLayerを削除する。
2. WebGISアプリの表示時刻に合わせて、データファイルを選択し、Layerを追加する。

iTownsでは、点群データの表示について、Potree形式のデータはiTowns標準のPotreeLayerを使用して描画する。

CSV形式のデータについては点群データ表示ライブラリ PointCloud.js を作成し、それを利用して点群データを表示する。

この実装は <https://github.com/SIPupstreamDesign/ChOWDER/> の実装を流用して作成している。

点群データ用のクラスとして itowns.GeometryLayer を継承した PointCloudLayer を実装している。

PointCloudLayerのコンストラクタで実施

1. itowns.FileSource を利用してCSVファイルを読み取り、コード内に実装したカラムでデータをパースする
2. 点群データを描画するための3次元モデルを作成する。

PointCloudLayer#updatePointCloudで実施

3. データをもとに、各地点に点群データを配置する。

更に時系列データ表示、UI操作による点群データの描画変更のために、以下の処理を行う。

0. 時系列データとして、対応する時刻をYYYYMMDDhhmmの形式にした名前のファイルを準備する
1. 表示済の点群データ用のLayerを削除する。
2. WebGISアプリの表示時刻に合わせてCSVファイルを選択し、Layerを作成、追加する。

2. API

iTowns向けライブラリのAPIを記載する。

CreatePointCloudLayer

点群データ用Layerを作成する。

構文

CreatePointCloudLayer(itownsView, config)

引数

itownsView itowns.GlobeViewのオブジェクト

config Layerの設定値、以下サンプル

```
{"id": "test", "url": "./test.csv", "column": ["lng", "lat", "val"], "size": 20, "scale": 0.2, "color": "0x00ff00", "opacity": 1.0}
```

戻り値

PointCloudLayer (点群データ用Layerオブジェクト)

PointCloudLayer#updatePointCloud

グラフの再描画を実施する。

構文

updatePointCloud()

引数

なし

戻り値

PointCloudLayer (点群データ用Layerオブジェクト)

3. データ形式

Mapbox

CSV形式

lng	地点の経度
lat	地点の緯度
value	地点に関するデータ

iTowns (CSV形式とPotree形式の2つが表示可能)

CSV形式

lng	地点の経度
lat	地点の緯度
value	地点に関するデータ

Potree形式(バイナリデータ)

以下のツール等を利用してptree形式のデータを作成する

<https://github.com/potree/PotreeConverter/>

面群表示機能

複数のエリアに関する時系列データを平面ポリゴンで地図上に可視化する機能

1. 面群表示機能の処理詳細

MapboxではDeck.glのMapboxLayerを利用して描画する。詳細を以下に記述する。

- 1. インプットデータとして、JSONファイルをD3.js利用で読み取る。
 - 2. deck.glのMapboxLayerを利用して面群データ用のLayerを作成する。
 - 3. 面群データ用のLayerを地図へ追加する。
- iTownsでは、標準のFileSource、ColorLayerを利用して表示するため省略する。

更に時系列データ表示のために、以下の処理を行う。

- 0. 時系列データとして、対応する時刻をYYYYMMDDhhmmの形式にしたファイル名のjsonを準備する
- 1. 表示済の点群データ用のLayerを削除する。
- 2. WebGISアプリの表示時刻に合わせて、JSONファイルを選択し、Layerを追加する。

2. データ形式

Mapbox

JSON形式

contour	エリアの頂点の緯度経度の配列 [[経度1, 緯度1], [経度2, 緯度2], ...]
value	エリアに関するデータ

以下にサンプルを示す。

```
[
  {
    "contour": [[139.76, 35.68], [139.77, 35.68], [139.77, 35.69], [139.76, 35.69]],
    "value": 100
  },
  {
    "contour": [[139.78, 35.68], [139.79, 35.68], [139.79, 35.69], [139.78, 35.69]],
    "value": 300
  },
  {
    "contour": [[139.79, 35.68], [139.80, 35.68], [139.80, 35.69], [139.79, 35.69]],
    "value": 400
  }
]
```

iTowns

GeoJSON形式

stroke	エリアの枠線の色
stroke-opacity	エリアの枠線の透過度
stroke-width	エリアの枠線の幅
fill	エリアの描画色
fill-opacity	エリアの描画部分の透過度
value	エリアに関する測定値など

ジオメトリは <https://datatracker.ietf.org/doc/html/rfc7946> に従う

以下にサンプルを示す。

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [[139.77, 35.68], [139.78, 35.68], [139.78, 35.69], [139.77, 35.69]]
        ]
      },
      "properties": {
        "stroke": "#ff0000",
        "stroke-opacity": 1,
        "stroke-width": 1.0,
        "fill": "#00c000",
        "fill-opacity": 0.3,
        "value": 100
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [[139.78, 35.68], [139.79, 35.68], [139.79, 35.69], [139.78, 35.69]]
        ]
      },
      "properties": {
        "stroke": "#ff0000",
        "stroke-opacity": 1,
        "stroke-width": 1.0,
        "fill": "#00d000",
        "fill-opacity": 0.3,
        "value": 100
      }
    }
  ]
}
```


立体グラフ表示機能

空間分布する時系列データを立体グラフで地図上に可視化する機能

1. 立体グラフの処理詳細

MapboxではTurf.jsとMapboxのfill-extrusionレイヤーを利用して描画する。詳細を以下に記述する。

1. インプットデータとして、CSVファイルをD3.js利用で読み取る。
2. Turf.jsのポリゴン作成機能(turf.circle)とMapboxのLayerを利用して立体グラフ用のLayerを作成する。
3. 立体グラフ用のLayerを地図へ追加する。

更に時系列データ表示、UI操作による立体グラフの描画変更のために、以下の処理を行う。

0. 時系列データとして、対応する時刻をYYYYMMDDhhmmの形式にした名前のファイルを準備する
1. WebGISアプリの表示時刻に合わせて、ファイルを選択し、Sourceデータを作成する。
2. 描画済みのLayerのsourceを新たに作成したsourceへ置き換える。ファイルがなければLayerを非表示にする。

iTownsでは、立体グラフ表示ライブラリ 3dBargraph.js を作成し、それを利用して立体グラフを表示する。詳細は以下に記述する。

この実装は <https://github.com/SIPupstreamDesign/ChOWDER/> の実装を流用して作成している。

立体グラフ用のクラスとして itowns.GeometryLayer を継承した BarGraphLayer を実装している。

BarGraphLayerのコンストラクタで実施

1. itowns.FileSource を利用してCSVファイルを読み取り、コード内に実装したカラムでデータをパースする
2. 立体グラフを描画するための立方体モデルを作成する。

BarGraphLayer#updateBarGraphで実施

3. データをもとに、各地点に立方体モデルを配置、指定された立体グラフの色、高さ、太さに合わせてモデルを変形する。

更に時系列データ表示、UI操作による立体グラフの描画変更のために、以下の処理を行う。

0. 時系列データとして、対応する時刻をYYYYMMDDhhmmの形式にした名前のファイルを準備する
1. 表示済の立体グラフ用のLayerを削除する。
2. WebGISアプリの表示時刻に合わせてCSVファイルを選択し、Layerを作成、追加する。

2. API

iTowns向けライブラリのAPIを記載する。

CreateBargraphLayer

立体グラフ用Layerを作成する。

構文

CreateBargraphLayer(itownsView, config)

引数

itownsView	itowns.GlobeViewのオブジェクト
config	Layerの設定値、以下サンプル { "id": "test", "url": "./test.csv", "column": ["name", "lng", "lat", "val"], "size": 20, "scale": 0.2, "color": "0x00ff00", "opacity": 1.0 }

戻り値

BargraphLayer (立体グラフ用Layerオブジェクト)

BargraphLayer#updateBarGraph

グラフの再描画を実施する。

構文

updateBarGraph()

引数

なし

戻り値

BargraphLayer (立体グラフ用Layerオブジェクト)

3. データ形式

Mapbox

CSV形式

name	地点の名称(サンプルアプリでは使用しない)
lng	地点の経度
lat	地点の緯度
val	地点に関するデータ

iTowns

CSV形式

name	地点の名称(サンプルアプリでは使用しない)
lng	地点の経度
lat	地点の緯度
val	地点に関するデータ

3次元オブジェクト表示機能

移動体のような位置情報を持つ時系列データを3次元オブジェクトで地図上に可視化する機能

1. 3次元オブジェクトの処理詳細

MapboxではTurf.jsとMapboxのfill-extrusionレイヤーを利用して描画する。詳細を以下に記述する。

1. インプットデータとして、CSVファイルをD3.js利用で読み取る。
2. Turf.jsのポリゴン作成機能(turf.polygon)とMapboxのLayerを利用して3次元オブジェクト用のLayerを作成する
3. 3次元オブジェクト用のLayerを地図へ追加する。

更に時系列データ表示、UI操作による3次元オブジェクトの描画変更のために、以下の処理を行う。

0. 時系列データとして、対応する時刻をYYYYMMDDhhmmの形式にした名前のファイルを準備する
1. WebGISアプリの表示時刻に合わせて、ファイルを選択し、Sourceデータを作成する。
2. 描画済みのLayerのsourceを、新たに作成したsourceへ置き換える。ファイルがなければLayerを非表示にする。

iTownsでは、3次元オブジェクト表示ライブラリ 3dObject.js を作成、それを利用して3次元オブジェクトを表示する。

この実装は <https://github.com/SIPupstreamDesign/ChOWDER/> の実装を流用して作成している。

3次元オブジェクト用のクラスとして itowns.GeometryLayer を継承した 3dObjectLayer を実装している。

3dObjectLayerのコンストラクタで実施

1. itowns.FileSource を利用してCSVファイルを読み取り、コード内に実装したカラムでデータをパースする
2. 3次元オブジェクトを描画するための3次元モデルを作成する。

3dObjectLayer#update3dObjectで実施

3. データをもとに、各地点に3次元オブジェクトを配置、指定されたの色で描画する。

更に時系列データ表示、UI操作による3次元オブジェクトの描画変更のために、以下の処理を行う。

0. 時系列データとして、対応する時刻をYYYYMMDDhhmmの形式にした名前のファイルを準備する
1. 表示済の3次元オブジェクト用のLayerを削除する。
2. WebGISアプリの表示時刻に合わせてCSVファイルを選択し、Layerを作成、追加する。

2. API

iTowns向けライブラリのAPIを記載する。

Create3dObjectLayer

3次元オブジェクト用Layerを作成する。

構文

Create3dObjectLayer(itownsView, config)

引数

itownsView itowns.GlobeViewのオブジェクト

config Layerの設定値、以下サンプル

{ "id": "test", "url": "./test.csv", "column": ["name", "lng", "lat", "angle"], "size": 20, "scale": 0.2, "color": "0x00ff00", "opacity": 1.0 }

戻り値

3dObjectLayer (3次元オブジェクト用Layerオブジェクト)

3dObjectLayer#update3dObject

グラフの再描画を実施する。

構文

update3dObject()

引数

なし

戻り値

3dObjectLayer (3次元オブジェクト用Layerオブジェクト)

3. データ形式

Mapbox

CSV形式

name 地点の名称(サンプルアプリでは使用しない)

lng 地点の経度

lat 地点の緯度

angle 3次元オブジェクトの角度

iTowns

CSV形式

name 地点の名称(サンプルアプリでは使用しない)

lng 地点の経度

lat 地点の緯度

angle 3次元オブジェクトの角度

相対位置関係判定機能

空間分布するレイヤー間の位置関係について地図上に可視化する機能

1. 面群データ内の点データ判定処理詳細

MapboxではTurf.jsを利用して面群データと点群データの位置関係を判定する。詳細を以下に記述する。

1. インプットデータとして、点群データ(点群表示機能)、面群データ(面群表示機能)をそれぞれ準備する。
 2. 点群データと面群データの読み込みを待ち、両方のデータが読み込まれたのちに、位置判定処理を開始する。
 3. Turf.jsのポリゴン内外判定処理(turf.inside)をして、点群データの1つ1つが面群データのいずれ化の内部かどうかを判定する。
 4. 面群データ内の点は赤、面群データ外の点は青を指定し、Layerを作成、追加する。
- 時間変更など、データの変更があった場合も、それぞれのデータ読み込み後に上記の判定を実施する。