

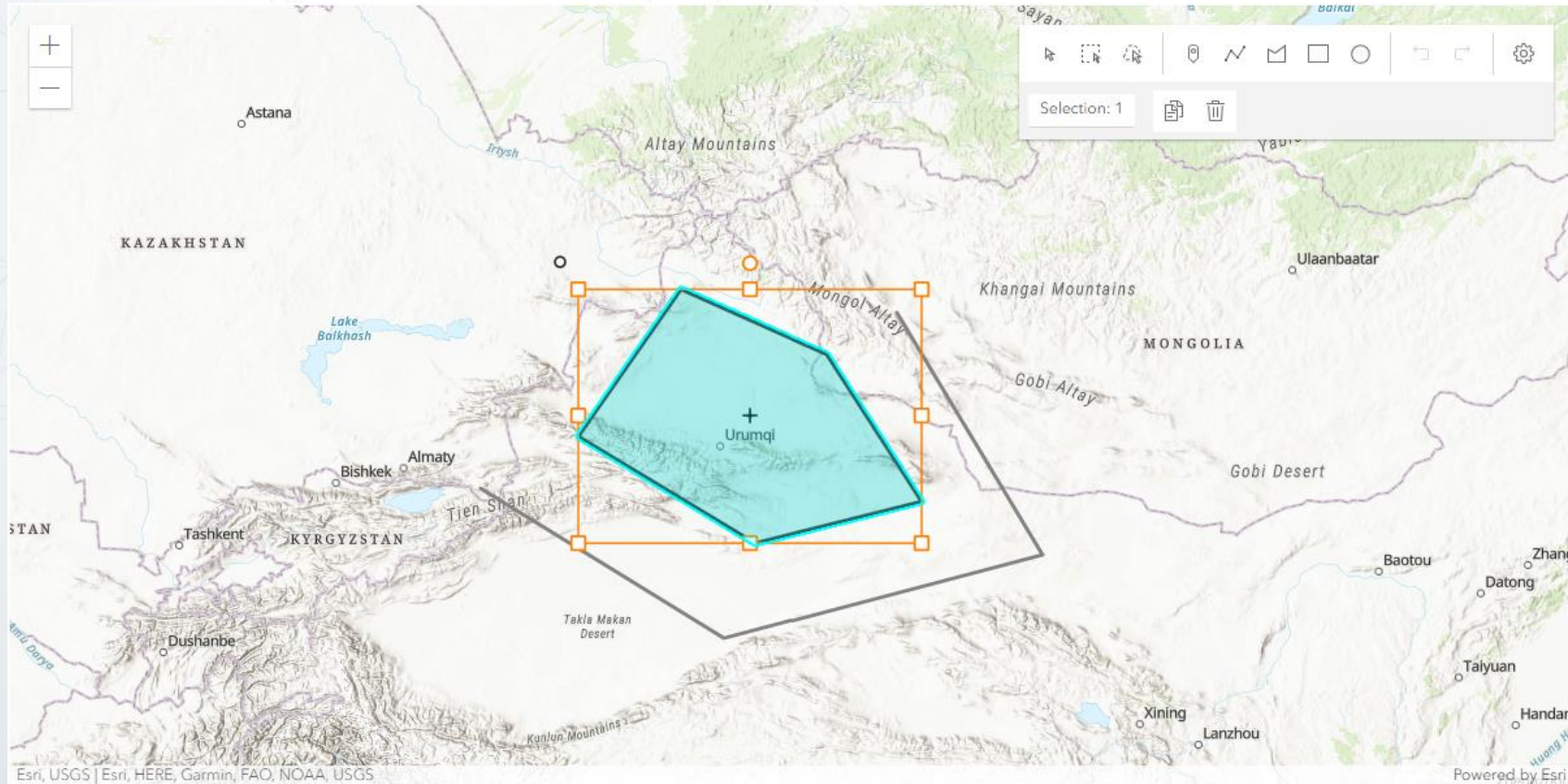
A large, abstract graphic on the left side of the slide. It consists of a grid of squares and rectangles in various shades of green and teal. Some squares contain smaller images, including a topographic map, a city street map, and a landscape with trees. The background of the graphic is a light gray network of interconnected lines and dots, resembling a network or data structure.

Assignment 3: Draw Tools & Apply Edits

Web Intermediate Course

Sketch Widget

- Sketch เป็น widget ชนิดหนึ่ง ที่เราสามารถเพิ่มลงใน map เพื่อวาด Point / Polyline / Polygon graphic ลงบน map พร้อมทั้งสามารถปรับรูปร่าง graphic ได้ค่อนข้างอิสระ มาพร้อม UI ที่พร้อมใช้งาน



Sketch Widget (ต่อ)

- Sketch widget ต้องการ MapView และ GraphicsLayer ในการใช้งาน
- จำเป็นต้องเซต event ต่างๆ ด้วย Sketch.on() เพื่อการใช้งานที่สมบูรณ์

```
import Sketch from "@arcgis/core/widgets/Sketch";
```

```
let sketch = new Sketch({  
  layer: this.graphicsLayer,  
  view: this.mapView  
});  
  
// Listen to sketch widget's create event.  
sketch.on("create", (event) => {  
  if (event.state === "complete") {  
    // Do something when finished drawing  
  }  
});  
  
this.mapView.ui.add(sketch, "top-right")
```

Draw

- Draw เป็น class สำหรับใช้วาด graphic ลงบน map ได้เช่นกัน แต่ต้อง custom เองเยอะกว่าเนื่องจากไม่มี UI พร้อมใช้งานเหมือน Sketch
- ต้องการ MapView ในการใช้งาน
- เริ่มการวาดโดยใช้คำสั่ง `Draw.create(drawAction)` โดย `drawAction` คือสิ่งที่จะวาด ("point", "polyline", "polygon" เป็นต้น)
- ควบคุมการแสดงผลด้วย `DrawAction.on(action, handlerFunction)` โดย event ที่ส่งเข้ามายัง handler จะมี property ชื่อ `vertices` เป็นตัวระบุจุดยอดมุมที่กำลังวาดอยู่ ซึ่งต้องนำ property ดังกล่าวไปสร้าง `geometry` และ `graphic` อีกทีหนึ่ง

Draw (ต่อ)

```
import Draw from "@arcgis/core/views/draw/Draw";
```

```
// Create Draw instance somewhere before using it (e.g. in ngOnInit)
this.draw = new Draw({
  view: this.mapView
});
```

```
// Call create() whenever you need to draw something
drawPolygon(){
  const action = draw.create("polygon")

  // fires when a vertex is added
  action.on("vertex-add", (evt) => {
    this.updatePolygon(evt.vertices);
  });

  // fires when the pointer moves
  action.on("cursor-update", (evt) => {
    this.updatePolygon(evt.vertices);
  });
  ...
}
```

Draw (ต่อ)

```
...
// fires when a vertex is removed
action.on("vertex-remove", (evt) => {
  this.updatePolygon(evt.vertices);
});

// fires when the drawing is completed
action.on("draw-complete", (evt) => {
  this.updatePolygon(evt.vertices);
  // Also do anything else after finished drawing
});
}

// This method is for updating Polygon graphic on the map
updatePolygon(vertices){
  this.mapView.graphics.removeAll()
  const polygon = new Polygon({ rings: vertices })
  const polygonGraphic = new Graphic({ geometry: polygon, symbol: this.fillSymbol })
  this.mapView.graphics.add(polygonGraphic)
}
```

Apply Edits

- นอกจากการ query feature แล้ว เรายังสามารถทำการ add/edit/delete feature ต่างๆ ของ FeatureLayer ได้ โดยการใช้งาน FeatureLayer.applyEdits(edits)
- addFeatures ต้องส่งค่าเป็น Array ของ Graphic ซึ่งระบุทั้ง geometry และ attributes ที่ต้องการ add ลงไปใน FeatureLayer
- updateFeatures ต้องส่งค่าเป็น Array ของ Graphic เช่นเดียวกับกรณี addFeatures แต่ใน attributes จำเป็นต้องระบุ objectid ของ feature ที่ต้องการแก้ไขข้อมูลด้วย
- deleteFeatures ต้องส่งค่าเป็น Array ของ Graphic ซึ่งระบุ objectid ลงไปใน attributes ด้วย หรือจะส่งเป็น Object ที่มีแต่ objectid ก็ได้
- ผลลัพธ์จากการทำ applyEdits เช็คได้จาก response.addFeatureResults, response.updateFeatureResults และ response.deleteFeatureResults

Apply Edits (တဲ့)

```
const addFeature = new Graphic({
  geometry: pointGeometry,
  symbol: pointSymbol,
  attributes: { name: "my-point" }
})

const updateFeature = new Graphic({
  geometry: pointGeometry,
  symbol: pointSymbol,
  attributes: { objectid: 123456, name: "my-point" }
})

const deleteFeature = { objectid: 987654 }

featureLayer.applyEdits({
  addFeatures: [addFeature],
  updateFeatures: [updateFeature],
  deleteFeatures: [deleteFeature]
}).then((response) => {
  console.log(response) // Check add/update/delete results
})
```

Assignment 3

จงสร้าง Web Page ต่อไปนี้ด้วย Angular ตามรายละเอียดต่อไปนี้

- แสดงแผนที่บนหน้าเว็บ พร้อม Sketch widget ที่มุมขวาบนของแผนที่ โดย Sketch widget ต้องวาดได้แค่ Polygon เท่านั้น
- เมื่อวาด Polygon เสร็จสิ้นแต่ละครั้ง ให้ทำการเรียก FeatureLayer.applyEdits เพื่อแทรก feature ใหม่เข้าไปใน FeatureLayer

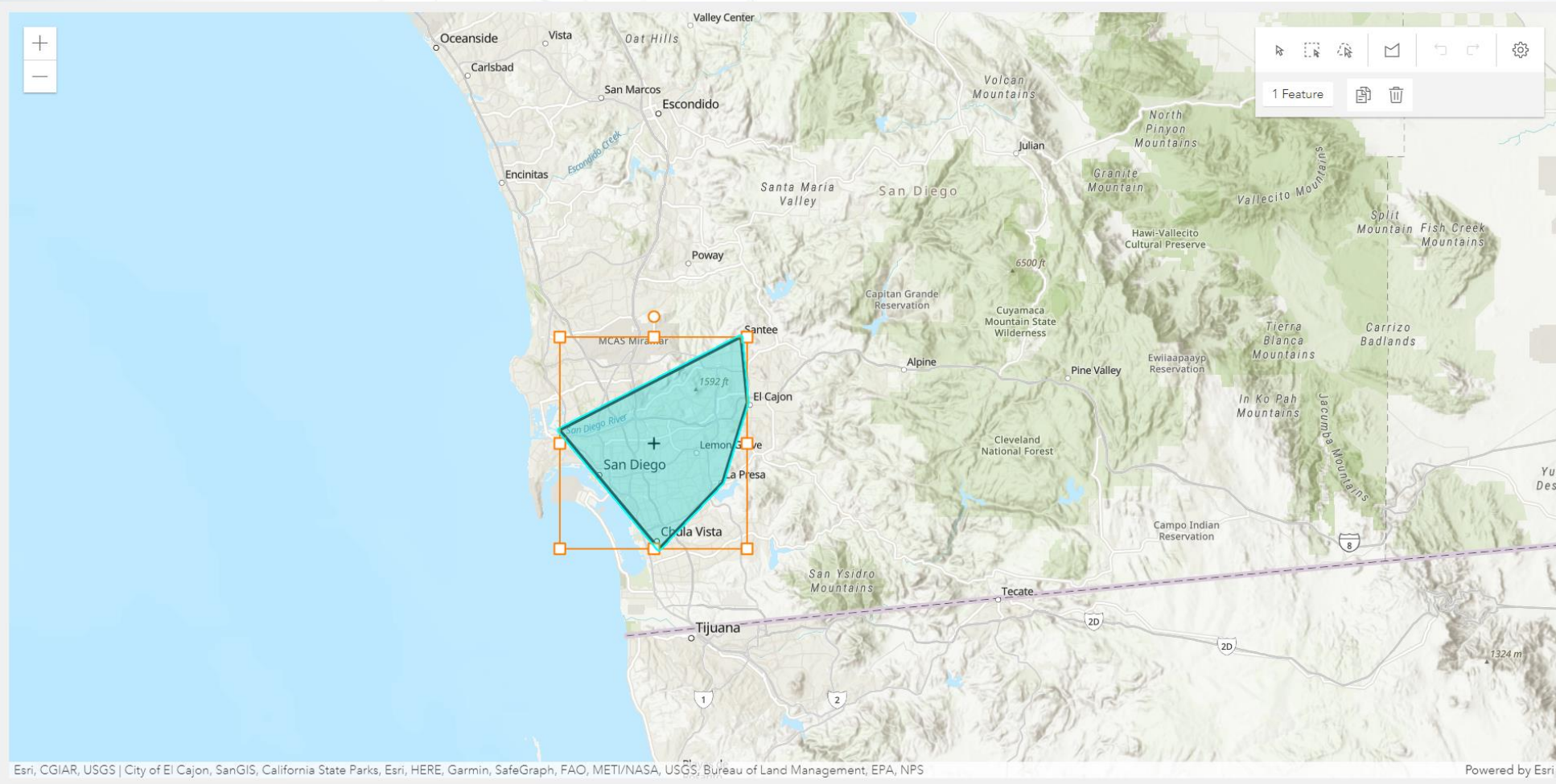
<https://sampleserver6.arcgisonline.com/arcgis/rest/services/Wildfire/FeatureServer/2>

- ในการ add feature แต่ละครั้ง ให้แนบ attribute ต่อไปนี้ไปกับ Graphic ที่ส่งไปด้วย
 - symbolid: รูปร่างของ polygon มีค่าระหว่าง 0 – 3 (เลือกใช้ค่าใดค่าหนึ่ง)
 - description: คำอธิบายเพิ่มเติม ใส่ text อะไรก็ได้ให้รู้ว่าเป็น feature ที่เรา add มา

Assignment 3 (ต่อ)

- เมื่อทำการคลิกเลือก Polygon ใดๆ ที่วาดไปแล้วบนแผนที่ให้นำ geometry ของ graphic ที่เลือกไปทำ spatial query เพื่อดึง objectid ของ polygon ที่กำลัง edit
- เมื่อทำการย้ายตำแหน่ง หรือปรับรูปร่างของ Polygon เสร็จสิ้น ให้ทำการเรียก FeatureLayer.applyEdits เพื่ออัปเดต geometry ของ feature ที่เพิ่ง edit ไป
- เมื่อทำการลบ Polygon ออกจากแผนที่ด้วยปุ่มลบของ Sketch widget ให้ทำการเรียก FeatureLayer.applyEdits เพื่อลบ feature ออกจาก FeatureLayer ด้วย

Assignment 3 (ต่อ)



Assignment 3 (ต่อ)

สามารถตรวจสอบว่า Polygon ที่เพิ่มเข้าไปไปปรากฏใน FeatureLayer แล้วหรือยัง ได้ที่ <https://www.arcgis.com/home/webmap/viewer.html?url=https%3A%2F%2Fsampleserver6.arcgisonline.com%2Farcgis%2Frest%2Fservices%2FWildfire%2FFeatureServer%2F2&source=sd>

The screenshot displays the ArcGIS web map viewer interface. The top navigation bar includes 'ArcGIS My Map', 'Open in Map Viewer', 'Modify Map', and 'Sign In'. Below the navigation bar are 'Details', 'Edit', and 'Basemap' tabs. A search bar on the right contains the text 'Find address or place'. The main map area shows a topographic view of San Diego with a red hatched polygon overlaid, representing a wildfire response area. The polygon is located in the central-eastern part of the city, covering areas like San Clemente Canyon and parts of San Diego. The left sidebar, titled 'Add Features', lists 'Wildfire - Wildfire Response Polygons' with four options: 'IR Heat Perimeter', 'IR Intense Heat Area', 'Maximum Manageable Area (MMA)', and 'Temporary Flight Restriction (TFR)'. The bottom of the interface includes 'UNDO' and 'REDO' buttons, a scale bar (0 to 6 km), and a footer with 'Trust Center', 'Legal', 'Contact Esri', and 'Report Abuse'.

Hint

- ดูตัวอย่างการสร้างและใช้งาน Sketch widget ได้ที่ <https://developers.arcgis.com/javascript/latest/sample-code/sketch-geometries/>
- เนื่องจาก FeatureLayer ดังกล่าวเก็บเฉพาะ feature ที่เป็น Polygon เท่านั้น ดังนั้นตอนสร้าง Sketch widget ให้ปรับค่า property availableCreateTools เพื่อ fix ให้วาดได้แค่ Polygon อย่างเดียว
- ใน Sketch.on("create") ให้เช็ค event.state ว่าเป็น complete หรือยัง ถ้า complete แล้ว ให้เพิ่ม attributes เข้าไปที่ event.graphic ก่อนที่จะส่งไป FeatureLayer.applyEdits (ส่งไปที่ addFeatures)
- ใน Sketch.on("update") เมื่อ event.state มีค่าเป็น "start" ให้นำ geometry ของ polygon ที่เพิ่งคลิกเลือกไป ไปทำ spatial query เอา objectid มาเก็บใส่ตัวแปรไว้ และเมื่อ event.state มีค่าเป็น "complete" ค่อยนำ geometry ที่อัปเดตแล้วส่งไปที่ FeatureLayer.applyEdits พร้อมกับ objectid ที่ดึงเก็บไว้ (ส่งไปที่ updateFeatures)
- ใน Sketch.on("delete") ให้นำ objectid ที่เก็บค่าไว้ ส่งไป au feature ออกโดยเรียก FeatureLayer.applyEdits (ส่งไปที่ deleteFeatures)

Reference

- <https://sampleserver6.arcgisonline.com/arcgis/rest/services/Wildfire/FeatureServer/2>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-widgets-Sketch.html>
- <https://developers.arcgis.com/javascript/latest/sample-code/sketch-geometries/>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-layers-FeatureLayer.html#applyEdits>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-views-draw-Draw.html>

End of Assignment