

도시데이터 시각화

도시 데이터 시각화 (Urban Data Visualization)

- 도시를 이해하기 위해서는 공공기관 및 여러 기업으로부터 생산되는 도시데이터를 분석할 필요가 있음
- 도시 데이터 시각화는 이러한 도시데이터를 이해하고 분석하기 위한 중요한 도구 중 하나
- 도시 데이터 시각화를 통해 도시데이터를 여러 시각에서 분석하고 다양한 피드백을 통해 새로운 데이터에 대한 수요 창출
- 독립적으로 생산된 다양한 데이터를 함께 시각화함으로써 기존에는 인지하지 못했던 다양한 정보의 취득 가능

“Treepedia” by [MIT Senseable City Lab](http://senseable.mit.edu/treepedia)

- <http://senseable.mit.edu/treepedia>
- 자체 개발한 컴퓨터비전 알고리즘을 이용해 Google Street View의 사진들을 분석하여 각 도시의 녹색화 정도를 측정
- GVI(Green View Index)라는 측정단위를 자체개발하여 녹색화 정도를 수치화

“Solar Cities” by MIT Senseable City Lab

- <https://senseable.mit.edu/solar-cities>
- video: <https://www.facebook.com/watch/?v=1600555620110980>
- 각 도시의 빌딩의 모양, 태양의 궤적 등을 통해 잠재적인 태양광 에너지의 사용량을 측정하여 이를 3차원으로 시각화

“Sonic Cicites” by MIT Senseable City Lab

- <https://senseable.mit.edu/sonic-cities>
- 같은 장소(예: 뉴욕 센트럴파크)에서 촬영된 COVID-19 이전의 영상으로부터 추출한 사운드 데이터와 COVID-19 lock-down 기간 동안 녹음된 사운드에서 각각 다양한 소리(새소리, 차소리, 목소리, 공사현장 소리 등)를 추출 및 비교하여 어떻게 변화되었는가를 분석
- 기계학습(machine learning)을 이용하여 다양한 소리를 추출

“Underworlds” by MIT Senseable City Lab

- <http://underworlds.mit.edu>
- 자체 개발한 로봇을 이용, 도시의 여러 하수구에서 하수를 채취, 분석하여 식습관, 유전적인 성향, 약물, 전염병 등을 분석 하고 이를 시각화

deck.gl을 활용한 도시데이터 시각화

- 목표: 서울시 공공데이터를 지리정보와 연계하여 3차원으로 시각화
- <서울 열린데이터광장> (data.seoul.go.kr)에서 데이터를 취득
 - 동별 통계데이터: “동별”을 키워드로 검색하여 취득
 - 위치정보 데이터: “위치”를 키워드로 검색하여 취득. 위도&경도 형식의 위치데이터를 활용
- deck.gl의 두 예제를 수정하여 각 데이터를 시각화
 - GeoJsonLayer (Polygons): <https://deck.gl/examples/geojson-layer-polygons>
 - HexagonLayer: <https://deck.gl/examples/hexagon-layer>

활용기술

- deck.gl
 - WebGL2 기반의 JS 시각화 프레임워크
 - React, Mapbox GL, Google maps 등과의 통합 제공
- Mapbox – JS Map API
- React – JS front-end framework by Facebook
- NPM (Node.js) – JS package manager
- webpack – JS bundler

예제 프로그램 실행과정

- Mapbox (<https://mapbox.com>)에 가입하여 MapAccessToken 취득
- Mapbox token을 “MapboxAccessToken” 환경변수에 저장
- Node.js 설치 (<https://nodejs.org>)
- 각 예제 폴더(population 혹은 wifi)에서 “npm install” 실행 → “node_modules” 폴더가 생성되고 필요한 JS package들이 설치됨
- “npm start” → 예제 실행

예제1: 서울시 (동별)주민등록인구 시각화

- 인구데이터
 - <https://data.seoul.go.kr/dataList/10043/S/2/datasetView.do>
 - csv형식으로 작성된 txt파일을 다운로드하여 사용
 - papaparse 라이브러리 함수를 사용하여 parsing
 - euc-kr로 인코딩되어 있는 파일을 UTF8로 변환하여 사용
- 동별 행정구역 데이터: <https://github.com/vuski/admdongkor>
 - 최신 파일 (*.geojson) 다운로드
 - 파일을 fetch한 후 서울시 데이터만 추출
- 두 개의 데이터를 읽은 후 인구데이터를 동별행정구역 object에 추가하여 GeoJsonLayer에 넘겨 시각화
- 두 데이터의 동이름에 오류가 있는 부분을 수정
- Tooltip으로 세부정보를 표시

예제2: 서울시 공공와이파이 위치정보

- 공공와이파이 위치정보
 - <https://data.seoul.go.kr/dataList/OA-1218/S/1/datasetView.do>
 - csv와 json 형식으로 제공
- 파일을 fetch한 후 (위도,경도) 데이터만을 추출하여 리스트로 변환한 후 HexagonLayer로 넘겨 시각화 → 육각형 패턴의 히스토그램으로 시각화

디버깅

- 브라우저에서 "view source"를 수행하면 webpack에 의해 변환된 소스를 보여주기 때문에 디버깅이 어려움
- Chrome의 경우, 아래와 같이 JS 소스를 열어 디버깅할 수 있음
 - (in "Sources" tab of Chrome devtools) "top" -> "App" -> "(unnamed folder)" -> "." -> "app.js"