Project for social good

- Game using google cloud services
- Multiplayer, players interact with each other some way
- Use pygaming to write code for game
- Puzzle game, at least 2 people, games like tag or hide and seek.
    - If not enough players in lobby, invite random players

**Name:** Capital Adventure

- Consists of business men/women who try to steal financial assets from a guard dragon.
- Like cops and robbers, but business men/women have to collect stock options scattered around the castle that two dragons defend
- The two dragons are controlled by only one player while the other players occupy only one business men/women per player..
- The dragons have to stop the business men/women from collecting all stock options.
- As the players collect more stock options, their briefcase becomes heavier, causing the players to move slower, allowing for the dragon to catch up to them. The players can choose to drop stock options in order to move faster.
- The players have three lives shared among them, once they get caught, they drop all stock options, are given x amount of invincibility seconds and speed to get away from the dragon that caught them. They lose a life in the process.

To Do

Research programming lang for web game development (python based pref)

-

Look into multiplayer options with language

- Some computer languages don't work

Project Name + Description

- Name:
- A web-based multiplayer game used as a way for people to interact from different physical locations (look up ____) (played in real time)
    - Coding
        - Collision detection
        - Speed manipulation
        - Lives counter
        - Item pickups and debuffs
        - Movement on an xy-axis
    - Map description
        - S

What technology to use/research

- Google cloud (plus register domain)
    - https://cloud.google.com/python/docs/getting-started
    - https://cloud.google.com/solutions/gaming
    - https://cloud.google.com/docs/tutorials?authuser=2#python
    - Apply for a certification to support https
    - A linux server (nginx)
    - Using php / python to process data from different players
    - User authentication
        - https://cloud.google.com/python/docs/getting-started/authenticate-users

- Pyjamas
- Pyjaco
- https://www.youtube.com/watch?v=McoDjOCb2Zo → Online Multiplayer Python

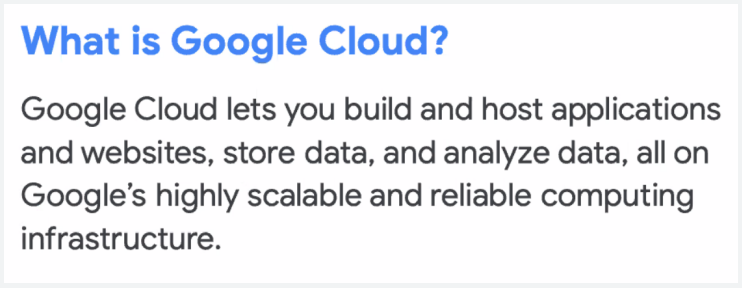
Research Notes

Google Cloud Project (w/ Python)

- Billing : https://cloud.google.com/billing/docs/how-to/modify-project?authuser=2
- Create a Firestore database in Native mode
    - When creating a project with firebase, they ask what organization to attach to the project. No organization means you have a limited amount of projects they allow you to make, so I'm attaching it to the school (ucdavis.edu)
    - In the Cloud Console, go to the **Firestore viewer** page
    - From the **Select a Cloud Firestore mode** screen, click **Select Native Mode**.
    - Select a location for your Firestore database.
    - Click **Create Database**.
- Enable the App Engine Admin, Cloud Storage, Cloud Logging, and Error Reporting APIs
    - https://console.cloud.google.com/flows/enableapi?apiid=appengine.googleapis.com%2Cstorage-api.googleapis.com%2Clogging.googleapis.com%2Cclouderrorreporting.googleapis.com&redirect=https%3A%2F%2Fconsole.cloud.google.com&authuser=2&_ga=2.16985337.1985377488.1610820712-1134590378.1610820712
- In Cloud Shell, open the app's source code
    - Cloud Shell provides command-line access to your Google Cloud resources directly from the browser.
- Cloud Shell (to run the app)
    - In Cloud Shell, install your app dependencies using pip
        - pip3 install -r requirements.txt --user
        - pip3 install gunicorn --user
        - ~/.local/bin/gunicorn -b :8080 main:app

- https://cloud.google.com/python/docs/getting-started?authuser=2#deploying-your-app-to-app-engine
    - For this example, you use App Engine to deploy a scalable app to Google Cloud. With zero-configuration deployments and zero server management, App Engine lets you focus on writing code. Plus, App Engine automatically scales to support sudden traffic spikes.
- **Monitoring your app using Google Cloud's operations suite**
    - https://cloud.google.com/python/docs/getting-started?authuser=2#monitoring-your-app-using-google-clouds-operations-suite
    -

`Google Cloud Panel Notes`




- Find data sets in the resource section and append them as a variable in query
- add member to the resource and set role to communicate live data between topics
    - Add openaq-streaming@openaq-streaming.iam.gserviceaccount.com with Pub/Sub Publisher role.
- http://bit.ly/gcloud-big-data-workshop-pubsub-register
    - Take project ID and put in google form
    - Topic name into google form
- Should see data streaming to your topic in terminal
- Add subscriptions to topic to view messages and pull data from other subscribers
- Pub Schema, event time stamp
    - http://bit.ly/cloud-big-data-workshop-pubsub-schema
    - timestamp AS sample_timestamp
    - CURRENT_TIMESTAMP() AS dataflow_timestamp
- https://cloud.google.com/vision/docs/drag-and-drop

Take input from player server side

Appl

Backend (Server Side)

- Store data from other players()
- Website server is created (powered by ubuntu 20.04 LTS with nginx)
    - Ssh: host: 34.94.236.123  username:root  passwd: capventure

Frontend

- https://docs.google.com/presentation/d/1q98yFrK1rCTsmVFU68wL9ucPXKgrgOTFLi7-9YBXB10/edit
- https://github.com/include-davis/hackdavis2021workshop
- You can embed python!
- But GitHub pages only works for static sites
- So you would have to do an extra step and link your GitHub to another site
- Heroku, Netlify, AWS, are some examples