

# Vincent Tantra

<https://vtantra.github.io/>

(510) 828-6581 | 2010 Milvia Street Unit 212, Berkeley, CA 94704 | [v\\_tantra@berkeley.edu](mailto:v_tantra@berkeley.edu)

*UC Berkeley undergraduate with 1 year of industry experience in web application development and game development engines.  
Based in the California Bay Area with interests in frontend, graphics, VR/AR and UX/UI.*

---

## EDUCATION

### University of California, Berkeley

Graduating Dec 2020

- B.S in Electrical Engineering & Computer Science, 3.34 GPA
- Coursework: Data Structures, Artificial Intelligence, Algorithms, Graphics, Computer Security, Image Processing

---

## SKILLS

- **Programming Languages:** Python, Lua, Java, JavaScript, TypeScript, C, SQL, HTML/CSS, Golang, C++; VCS experience with git
- **Protocols & Libraries:** React, JSON, NumPy, spaCy, XMLUtils, REST, OpenCV, GraphQL, PyTorch, Tensorflow
- **Frameworks:** Flask, React Native, Appium/Selenium, Ruby on Rails, Bootstrap

---

## EXPERIENCE

### Roblox Corporation – Software Engineering Intern

JUN 2020 – AUG 2020, SAN MATEO, CA

- Implemented 3D modeling tools for their proprietary game development engine & drove internal research on automation forward
- Developed a set dressing tool that enabled instant, controlled decoration on virtual terrain, outspeeding manual decoration by 10 times
- Spearheaded research on the engine's performance & scalability with the aforementioned tool & procedural generation, allowing simultaneous smart placement of around 500 high-fidelity game assets in 0.53 seconds

### Middesk (CodeBase Client) – CodeBase Software Developer

JAN 2020 – MAY 2020, BERKELEY, CA

- Created a fully-hosted, flexible business-onboarding flow that accelerated & streamlined the process of collecting business partner info
- Implemented a robust form using React, Regex, & PostgreSQL, assuring the successful collection of values such as tax ID field & files
- Introduced lasting UI improvements on the original flow mockup, such as adding indicators for required vs. optional fields

### Storr, Inc. (CodeBase Client) – CodeBase Software Developer

SEP 2019– DEC 2019, BERKELEY, CA

- Created a social backend service for Storr's mobile retail app that enabled users to send SMS & email invites to their contacts
- Utilized Twilio & SendGrid's API to implement the send-invite endpoint with phone number/email validation
- Developed a comprehensive suite of unit & integration tests & ensured isolated functionality for cases such as malformed data input

### 21 Labs, Inc. – Software Engineering Intern

JUN 2019 – AUG 2019 & DEC 2019, CAMPBELL, CA

- Improved the test maintenance experience for beta users of their web service, which provides autonomous QA testing for mobile apps
- Implemented DOM parsing & Android log processing endpoints, allowing other microservices to analyze an app's structure with ease
- Built a custom analyser that crawls an app's frontend structure & UI using tree search algorithms, XMLdiff & NLP libraries (spaCy)
- Accelerated test suite management by building a flow updater that detects changes in an app's screen flow & enables tests to self-adapt

---

## PROJECTS

### Virtual Reality Environment Simulator – C#

APR 2019 – MAY 2019

- Charted a mobile virtual reality application for the iPhone that allowed exploration of lighting in a pre-rendered 3D environment
- Incorporated a wide range of different technologies & applications, such as Unity3D, XCode, & the Google Cardboard SDK
- Tested realistic lighting practices in a VR environment, & used vector mathematics to implement other functionality, like movement

### Concurrent Cached File Server – Golang

APR 2019 – MAY 2019

- Developed a basic file server that allowed concurrent server accesses & implemented caching for improved performance
- Engineered solutions to common file server issues, such as timeout issues or read errors
- H&led security concerns such as traversal attacks by programming a file sanitization method using iterative string manipulation

### Physical Renderer & Pathtracer – C++

MAR 2019 – APR 2019

- Implemented the core routines of an image renderer using path-tracing algorithms, with graphic practices such as ray-scene intersection
- Utilized the bounding volume hierarchy as a core data structure to lessen ray intersection tests & reduce rendering time by around 70%
- Recreated illumination techniques combined with adaptive sampling, allowing for realistic lighting of 3D models in a contained box