# Sebastian Greczek

Chicago, IL • 708-745-1475 • sebastiangreczek1@gmail.com
GitHub • Portfolio Website • Portfolio Website GitHub • LinkedIn

## Career Objective

Graduating Computer Science major with a strong passion for software development seeking to gain experience and grow as a developer.

## Education

**University of Illinois at Chicago (BS in Computer Science)**

- **Honors:** ***cum laude***
- **GPA: 3.73**

## Skills Summary **Programming:** C/C++, Java, Scala, SQL, JavaScript, HTML/CSS, Python

- **AWS Deployment:** EC2, Lambda, S3, API Gateway, Amplify
- **Software Development:** Scrum Software Development(Agile), Jira, Git
- **Technologies/Frameworks:** Spark, Kafka, Visual Studio, Eclipse, Intellij, Linux, Windows

## Project Experience

**Serverless REST API** ***Code: GitHub***

- Build a rest api that handles requests and returns logs within a specified time interval. Logs are generated on an EC2 instance and are stored on S3. Next, using AWS Lambda functions written in node.js, which check if logs exist in the given time interval and if they do another function searches for the logs within the time interval and returns the logs. These functions are exposed using Amazon API Gateway. The rest service is built using Akka http written in scala which communicates with Gateway to invoke the Lambda functions.

**Javascript Raytracer** ***Code: GitHub Live Demo: JSRaytracer***

- Developed a javascript raytracer that uses the Canvas API to draw six spheres to a html element. A path is traced through each pixel from the camera to the object. Once the ray reaches an object, the ray will bounce off the surface, towards the two light sources. Shadows are determined on whether the bounced off ray reaches the light sources or is intersected by another object first. The user has control over the different lighting components, depth of reflections, and size, color, and position of the six spheres.

**Graph Algorithms Visualization** ***Code: GitHub Live Demo: GraphAlgo***

- Creates a html grid that represents a graph data structure. The user can select cells on the grid to specify walls, set start cells, and set end cells. They can then specify a traversal or search algorithm to simulate. A css animation will play showing how each cell is traversed based on the specified algorithm. Mazes or walls can be randomly generated to show more interesting applications of graph algorithms.

**WebGL Shadow Mapping** ***Code: GitHub Live Demo: ShadowMap***

- Developed a javascript program that uses the WebGL API to render a city skyline.The user can control the direction of the camera, change the perspective, change light direction, and zoom in and out. The shadow mapping technique is used to calculate the shadows. First the scene is rendered from the perspective of the light, the depth texture is extracted, and saved as a texture. Then the scene is rendered from the perspective of the camera and it is compared to the depth texture to determine whether the object is in shadow or light.