LEO PAN

✉ **leo.hl.pan@gmail.com** ☎ **(647)-262-9693** 📍 **Markham, Ontario** **leohlpan**

TECHNICAL SKILLS

Experienced	C++, C, Java, Python, MATLAB
Familiar	JavaScript, HTML, CSS, C#
Software/Libraries	React, React Native, OpenGL, Android Studio, Git, LaTeX

EDUCATION

University of Waterloo *September 2015 - April 2019*
Bachelor of Computer Science

PROJECTS

Piece of Pi *2019*

- Android app made built with React Native which breaks the digits of pi into smaller blocks. Designed to aid those who wish to memorize pi. Digits rendered using performance-efficient FlatList component.
- User settings allow users to set preferences such as starting digit. Also able to display Euler's number or tau instead of pi. AsyncStorage used to save preferences for future use.

Catch of the Day *2019*

- Website designed using React to model a fish market with customer independent shopping carts.
- Used props to update inventory and customer orders from nested components such that information presented to the user synced across the entire website.
- Stored inventory on Firebase for easy access on all connected devices. Restricted write access via user authentication to ensure only authorized users can freely edit inventory.

Advanced Ray Tracer *2018*

- Ray tracer designed to render a 3-dimensional scene as a 2-dimensional image.
- Features include: bounding boxes, reflection, refraction, adaptive anti-aliasing, soft shadows - all of which can be disabled to improve render time.
- Overcame difficulties in using complex mathematical formulas to model behaviour of light.
- Multithreading reduced render time by up to 93%. Able to render simple scenes instantly even with all features enabled.

Flag Trivia *2018*

- Android app built with Android Studio which tests the user's knowledge of country flags.
- Learned to transition and share information between different activities. Selected answers are shared and retained when navigating between pages. They are passed to a final activity to display a score.

Dungeon Crawler *2016*

- A dungeon crawler game on C++ built with various design patterns to avoid duplicate code and ensure better interaction between classes.
- Observer pattern used to interact with surrounding spaces for player and enemy movement. Enemy interaction during combat is also overseen by the observer pattern.
- Various types of power-up potions designed via decorator pattern. Enemies and player types also designed via decorator pattern because they share many commonalities.