GWiz
(The GPXWizard)

Katherine Duff, Ashley Kim, Elisa Luo, Rebecca Yao
Motivation

- So I can go on excessively long bike rides in the name of “research”
- Unleash that inner data junkie 😊
- Don’t let Strava steal your data*

*arguably the compromise of personal privacy is worth the gratification obtained from the social features
High-Level Overview

- Imperative
- Statically scoped
- Strongly typed
- Built in struct and list types
- GPX file analysis
GPX File

```xml
<gpx xmlns="http://www.topografix.com/GPX/1/1" creator="Garmin" version="1.1">
  <gpxmetadata></gpxmetadata>

  <trk>
    <trkseg>
      <trkpt lat="49.3552480" lon="123.1350690">
        <ele>299.6</ele>
        <time>2021-04-22T15:45:05Z</time>
        <extensions>
          <gpxtpx:TrackPointExtension>
            <gpxtpx:atemp>28</gpxtpx:atemp>
            <gpxtpx:hr>104</gpxtpx:hr>
          </gpxtpx:TrackPointExtension>
        </extensions>
      </trkpt>

      <trkpt lat="49.3552480" lon="123.1350690">
        <ele>298.0</ele>
        <time>2021-04-22T15:45:09Z</time>
        <extensions>
          <gpxtpx:TrackPointExtension>
            <gpxtpx:atemp>28</gpxtpx:atemp>
            <gpxtpx:hr>106</gpxtpx:hr>
          </gpxtpx:TrackPointExtension>
        </extensions>
      </trkpt>
    </trkseg>
  </trk>
</gpx>
```
Compiler Architecture

- **Lexer** (scanner.mll)
- **Parser** (gwizparse.mly)
- **AST** (ast.ml)
- **Semantic Checking** (semant.ml)
- **SAST** (sast.ml)

- **Source Code**
- **C Files** (list.c, struct.c, parse_gpx.c)
- **Linking**
- **Code Generation** (codegen.ml)
- **LLVM Executable**
GWiz's Evolution

Iteration 1
- Activity Class
  - Sounded cool but too cumbersome for user to define

Iteration 2
- Pass GPX file into a gwiz array of structs
  - Simple to pass pointers to and from C
  - Array of structs’ memory didn’t match up with C

Current iteration
- File pointer
  - Clean and easy to work with on the user end
  - Efficient
Key Language Features

- GPX file interpretation
  - File pointer
  - Built-in functions
  - Math functions
- Generate your own route*
  - Structs (Trackpoint)
  - Linked List

*may be a little cumbersome
Trackpoint is a struct of (double, double, int)
Used to represent latitude, longitude and time

```xml
<trkpt lat="49.3552480" lon="-123.1350690">
<ele>300.6</ele>
<time>2021-04-22T15:44:59Z</time>
</trkpt>
```

Trackpoint t;
int time;
double lat;
t=(4.5285, 1.4829, 562);
lat=getLat(t); //assigns lat to 4.5285
getTime(t); //assigns time to 562
Linked List

- Create Linked Lists of Ints and Trackpoints
- Append elements to the end of the list
- Access elements at specified index
- Leak memory

```
Trackpoint[] t;
Trackpoint a;
Trackpoint b;
Trackpoint c;
a=(4.5285, 1.4829, 562);
b=(34.897239, 8.2397, 1234);
t = [a]; //initialize list
t +<b>; //append b to end
c = <-0>; //access first element
```
In C: used libxml2, an XML parser, to select latitude, longitude, and time data for each trackpoint

Return a pointer to a list of trackpoints
GPX file is represented in GWiz as a file pointer
Handy built-in functions provide the building blocks for a variety of user-defined functions!

- GPX functions
  - void stat(file f) /* prints general stats for a quick overview */
  - int totTime(file f)
  - double totDist(file f)
  - double avgSpeed(file f)
- Math functions
  - intFloor(double d)
  - doubleFloor(double d)
  - squareRoot(double d)

```c
file f;
f = parseGPX("nameOfFile.gpx");
```
Code demo
Future Work

- Use graphs to see if 2 gpx file routes intersect
- Add elevation and heart rate to Trackpoint
- Modify and write to a new gpx file
- Conversion between double and int
  - Less safe, more convenient.
- Automatic garbage collection
  - Won’t run the risk of running out of memory
Thanks for Listening!