GrabMap Khmer Render solution

PIC: Wei Chen

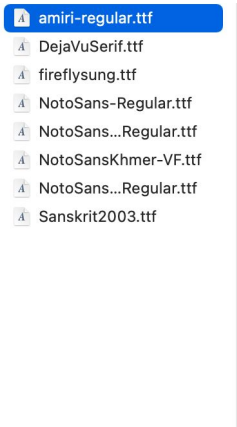
Render Khmer needs HarfBuzz+Freetype

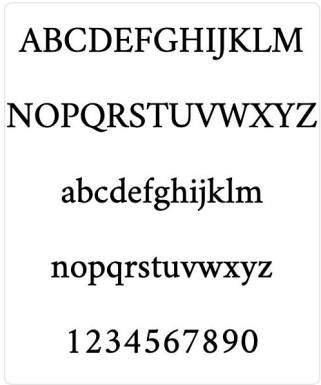
Freetype: font rasterization engine . Give a codepoint can render to a bitmap and draw parameters

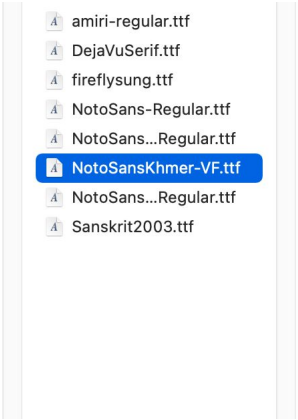
HarfBuzz: Char layout engine(shaping). Convert Unicode text to glyph indices and positions

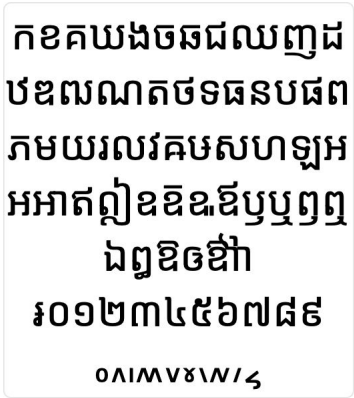
Font File

Font files contain one or more **fonts** that can be accessed by the operating system and applications.
Include char layout and char image.
Common font files. .ttf .otf .woff .fnt ...





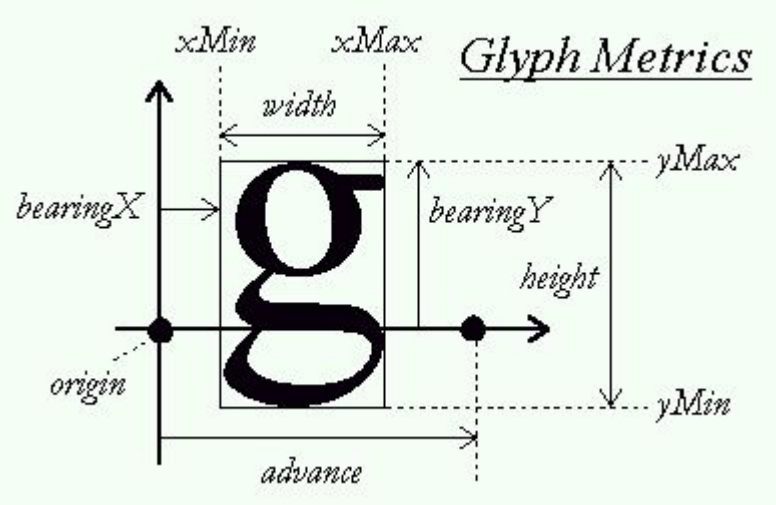


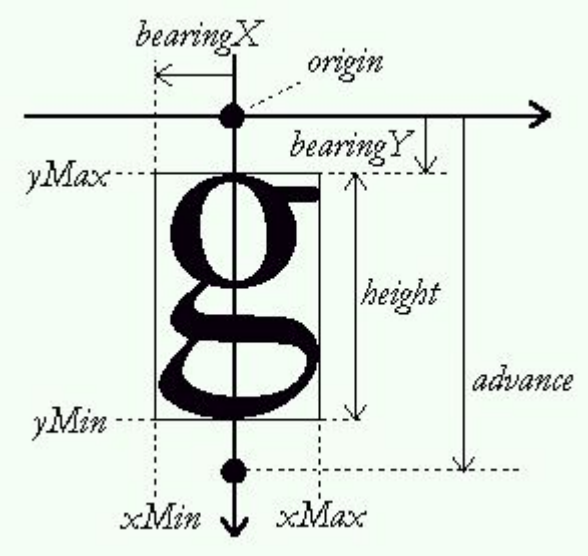


How Freetype draw text

Get char bitmap and draw Parameter by codepoint(Unicode 16)

```
FT_EXPORT( FT_Error ) FT_Load_Char(FT_Face   face, // font file & return glyph
                                   FT_ULong  char_code, // unicode
                                   FT_Int32  load_flags ); // Params
```





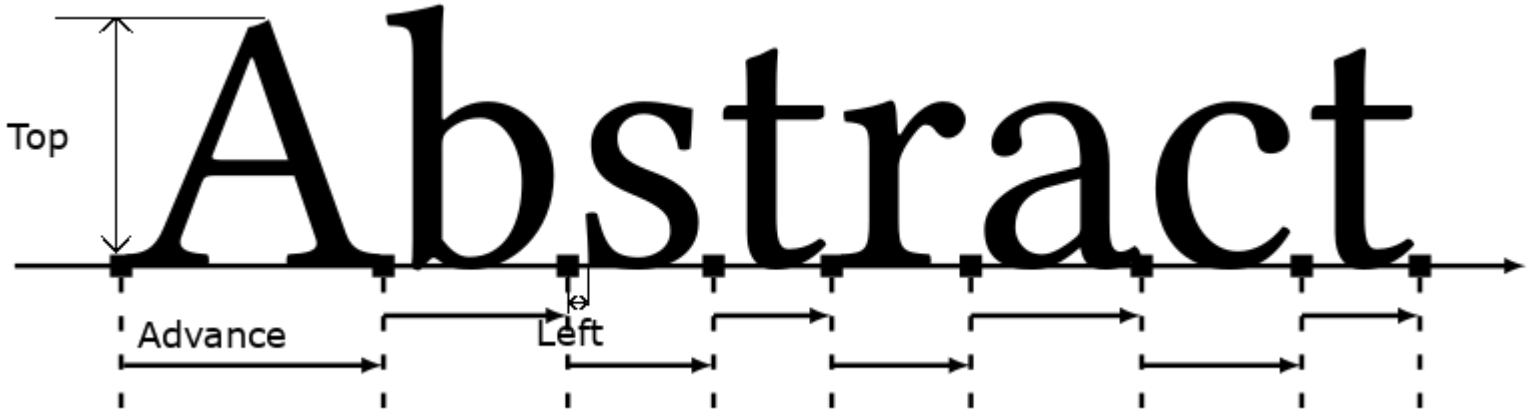
Draw Text With Glyph Metrics

How Mapbox manage glyph

https://github.com/mapbox/glyph-pbf-composite/blob/master/proto/glyphs.proto

```
message glyph {
    required uint32 id = 1;

    // A signed distance field of the glyph with a border of 3 pixels.
    optional bytes bitmap = 2;

    // Glyph metrics.
    required uint32 width = 3;
    required uint32 height = 4;
    required sint32 left = 5;
    required sint32 top = 6;
    required uint32 advance = 7;
}
message fontstack …
message glyphs …
```

Draw Text With Glyph PBF



Shaping

Just use freetype(or glyphpbf) can’t render Khmer correctly.

ស្តី 

ញ្ច 

ស្ត 

ណ្ត 

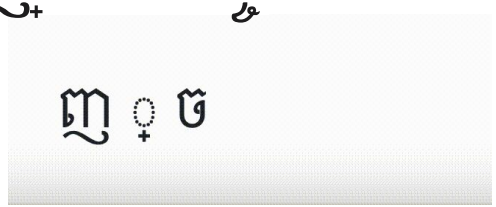
Some scripts need special layout, which is called shaping.

Why Freetype can’t draw Khmer

Khmer has combine and transform, 2 Codepoint can combine into a new glyph.
And can even change the previous glyph.

ស្ + ត + ី = ស្តី

ញ្ + ច = ញ្ច



The mutated glyphs exist in the font file

1. In unicode Khmer has only 114 code point:
   https://en.wikipedia.org/wiki/Khmer_(Unicode_block)

2. The font file “NotoSansKhmer-Regular” has 114 codepoint to load, but it has 362 glyphs

3. Freetype Can load these glyph by index

```
FT_EXPORT( FT_Error )
FT_Load_Glyph(FT_Face   face,          // font and return value
              FT_UInt   glyph_index,   // Glyph index in font file.
              FT_Int32  load_flags );
```

Glyph position can be determined by the previous codepoint

ត position may change from font setting:

1. ស្ ត → ស្ត
2. ណ្ ត → ណ្ត

So ត position is adjusted by ណ and make it looks like align to center.

How HarfBuzz work

```
// Source Code:  https://github.com/tangrams/harfbuzz-example
// Setup harfbuzz
hb_buffer_reset(buffer);

….
// set text
hb_buffer_add_utf8(buffer, text.c_data(), length, 0, length);

// harfbuzz shaping
hb_shape(font, buffer, features.empty() ? NULL : &features[0], features.size());

// Get Harfbuzz adjustion
unsigned int glyphCount;
hb_glyph_info_t *glyphInfo = hb_buffer_get_glyph_infos(buffer, &glyphCount);
hb_glyph_position_t *glyphPos = hb_buffer_get_glyph_positions(buffer, &glyphCount);

for(int i = 0; i < glyphCount; ++i) {
    // Load glyph by freetype via index
    Glyph* glyph = lib->rasterize(face, glyphInfo[i].codepoint);

    // glyphPos[i].x_offset & y_offset is position adjustion
    ...
```

Some example

ជនជាប់សង្ស័យ
32 45 226 46 122(-1.09,-0.703) 59 29 203 132(-0.0938,0.719) 53

ខណ្ឌពោធិ៍សែនជ័យ
26 40 175(-7.28,-0.625) 107 260 44 83(-0.781,-0.703) 108 59 45 32 132(-0.859,0.719) 53

ញញ
34 35 165(0.484,-0.625)

Mapbox issue: font pbf file

As A cross platform map engine. Mapbox use a workaround to remove the dependency to Freetype, and make it’s possible to draw font in JS.

```
        ]
    }
},
"sprite": "                                    styles/mono/sprite",
"glyphs": "                                    fonts/{fontstack}/{range}.pbf",
"layers": [
    {
        "id": "background",
        "type": "background",
        "paint": {
            "background-color": "#f5f5f5"
        }
    }
]
```

Mapbox lack

1. No way to load extra glyph(only unicode code point , no index).
2. No complex char layout.

Mapbox font pbf just the codepoint’s glyph load from font file.

How GrabMap fix this

1. Support download font files
2. Extend GlyphID, Add Font Type , eg. 0: Font pbf, 1: Khmer
3. Use Harfbuzz process text, use the result (Glyph index in font file) as Glyph ID, make offset by the way.
4. Use Freetype →render glyph bitmap (24 pcx), and transform the bitmap to SDF.
5. Khmer skip default shaping(BiDi, Arabic) layout (harfbuzz done it already)
6. Use Glyph index and offset draw text.

Support download font file from tile sever

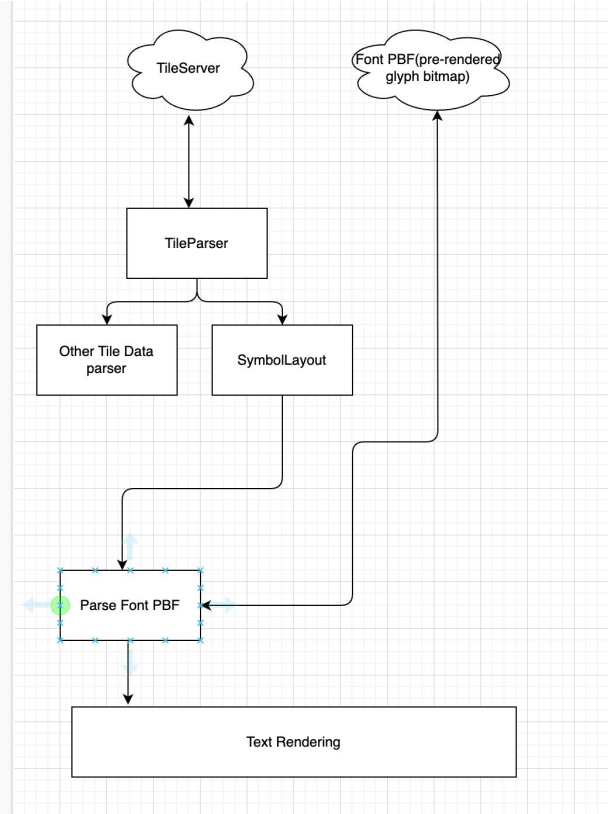
Support download font file from server

```
        ]
    }
},
"sprite": "https://[redacted]/styles/nav-day/sprite",
"glyphs": "https://[redacted]/fonts/{fontstack}/{range}.pbf",
"fonts": "https://[redacted]/fonts/{fontstack}/{language}.ttf",
"layers": [
    {
```

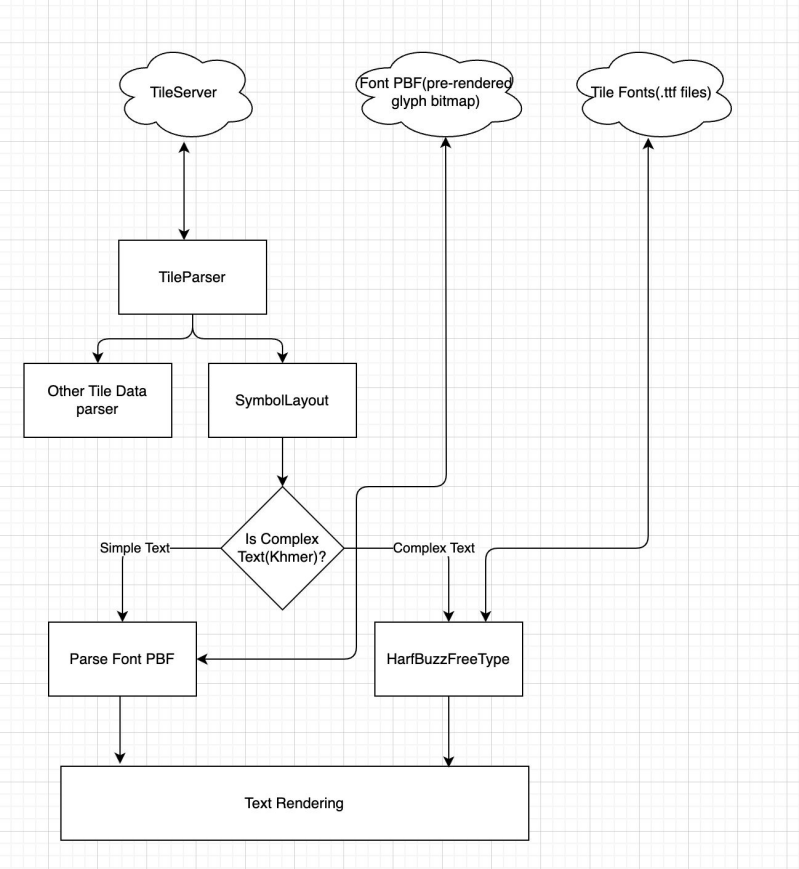
Extend Glyph ID

```
enum GlyphIDType : short {
        FontPBF = 0x00,
        Khmer = 0x01,
        Myanmar = 0x02
    };
// using GlyphID = char16_t; // Mapbox original GlyphID
union GlyphID {
        char32_t hash;
        struct {
            char16_t code;
            GlyphIDType type;
        } complex;
    };
```

Origin Text Layout Struct of Mapbox



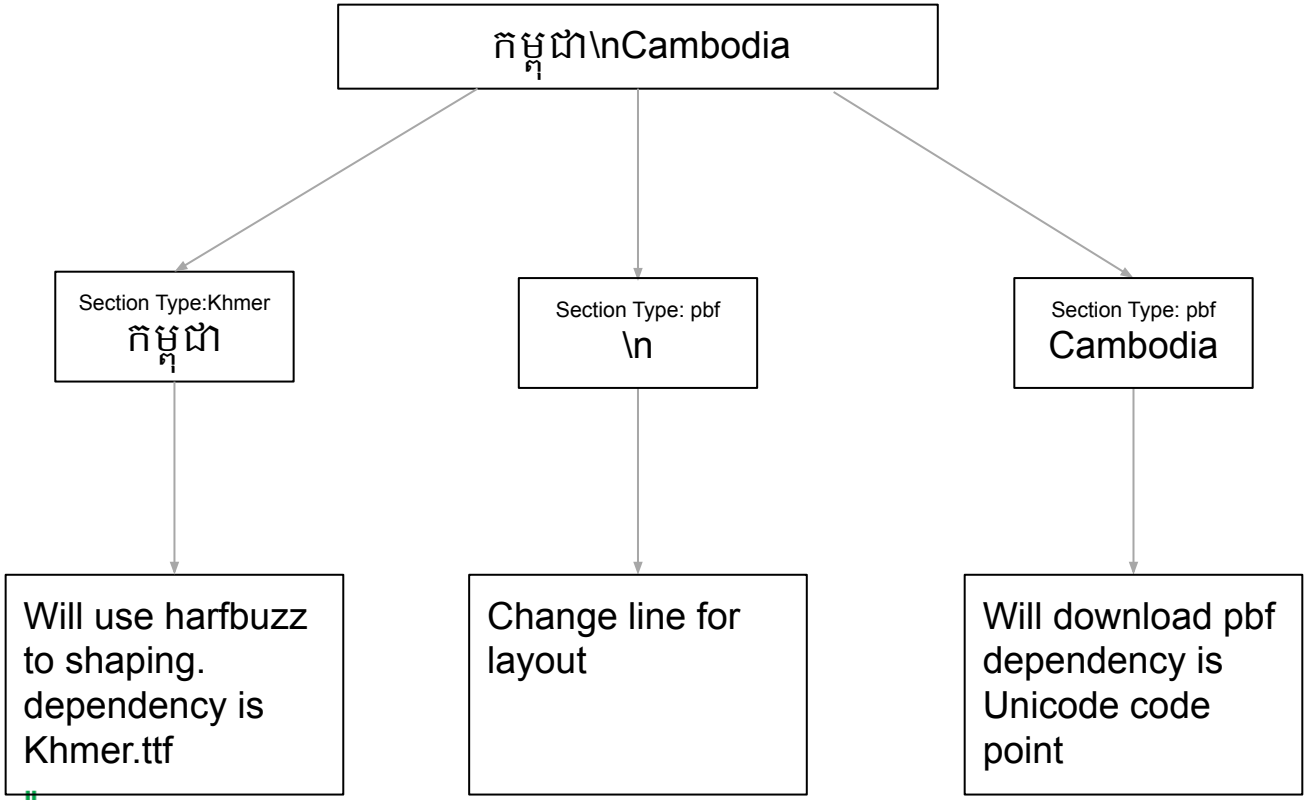
GrabMap Text Layout structure



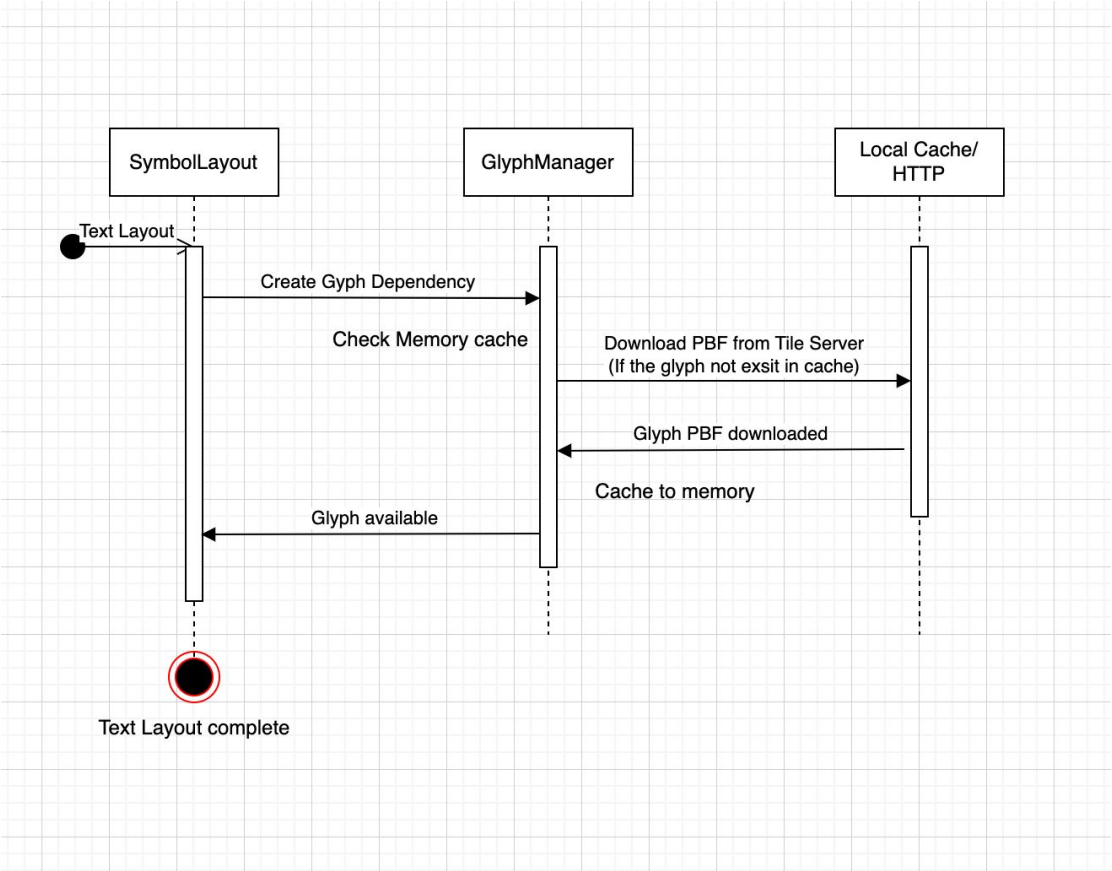
Extend TaggedString

```
Struct SectionOptions
{     …..
      FontStack fontStack;    // we must know the font stack to support Harfbuzz shaping
      ….

      GlyphIDType   type;     // String type
      std::shared_ptr<std::vector<HBShapeAdjust>> adjusts;   // offset by Harfbuzz
      ...
};

struct TaggedString {
     std::vector<SectionOptions> sections;
};
```

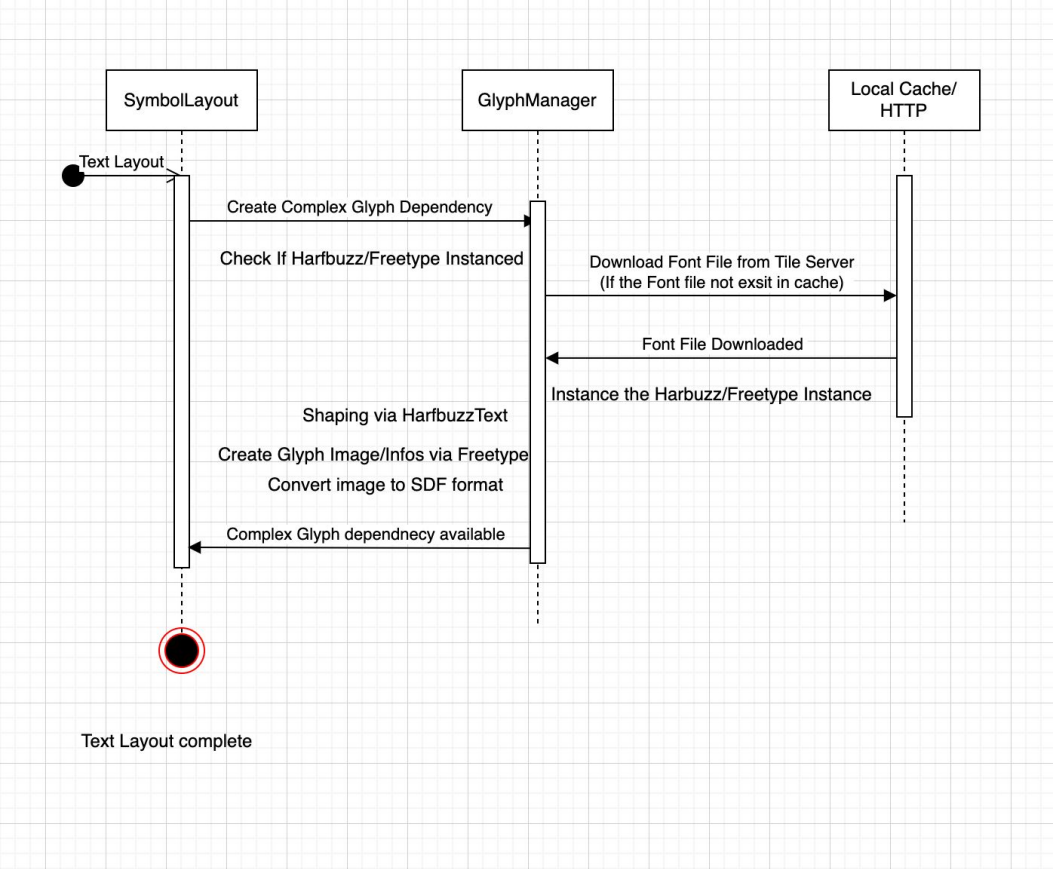
Filter Out The Complex Sub-String as independent section



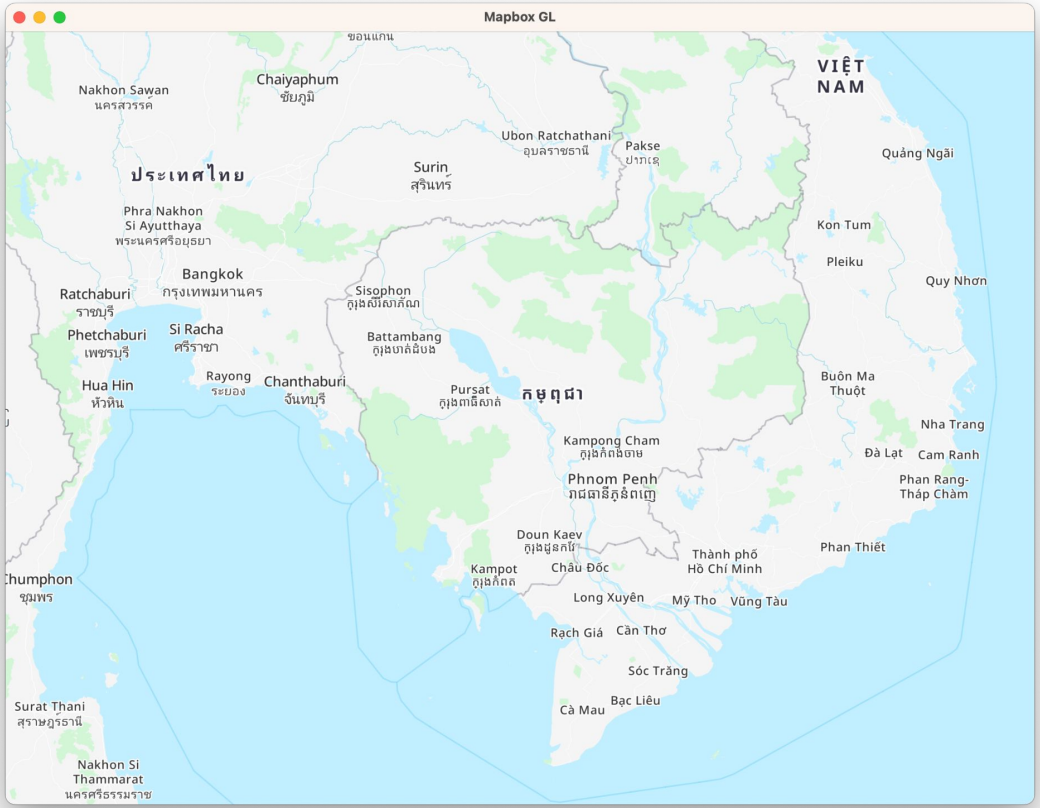
PBF Text Layout Sequence



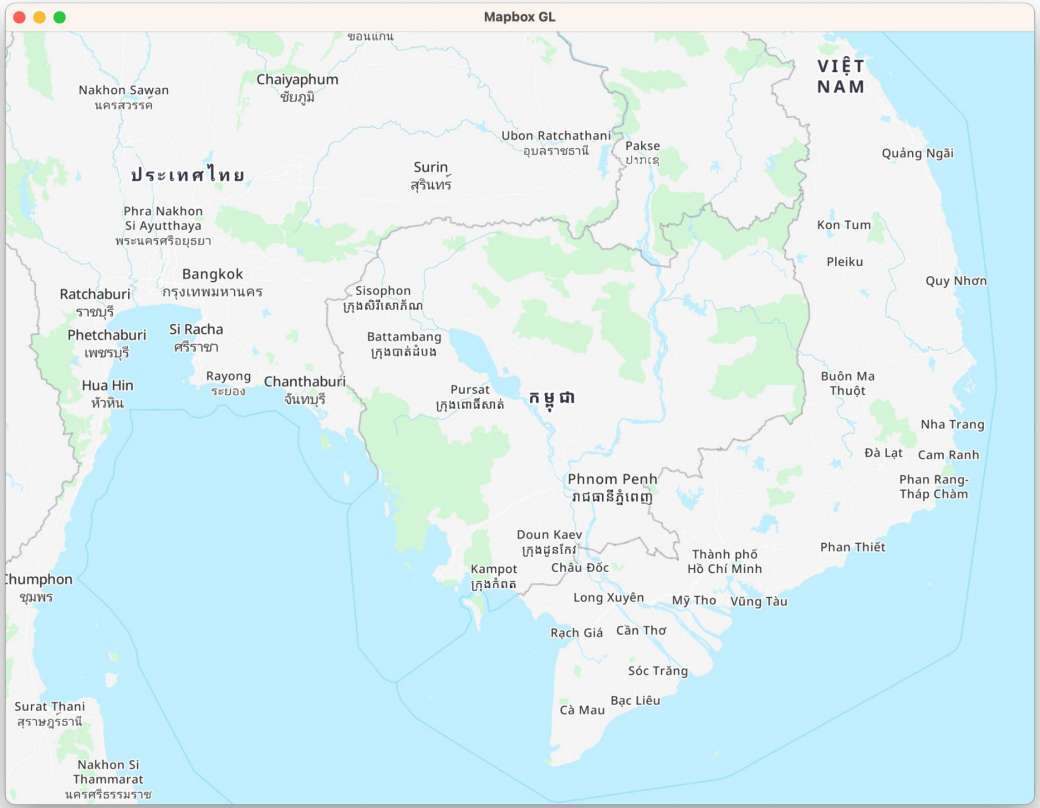
Complex Text Layout Sequence



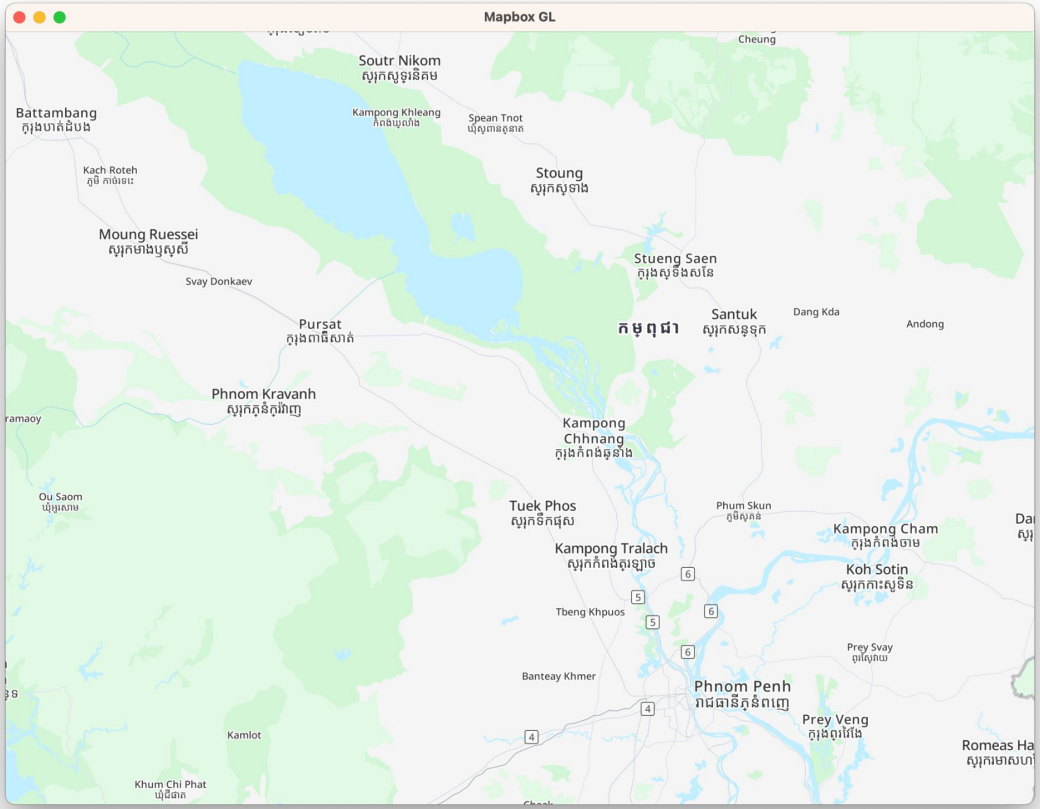
Effect


Mapbox GL
VIỆT NAM
ประเทศไทย
กัมพูชา
Nakhon Sawan
นครสวรรค์
Chaiyaphum
ชัยภูมิ
Ubon Ratchathani
อุบลราชธานี
Pakse
Surin
สุรินทร์
Quảng Ngãi
Phra Nakhon Si Ayutthaya
พระนครศรีอยุธยา
Kon Tum
Pleiku
Quy Nhơn
Bangkok
กรุงเทพมหานคร
Ratchaburi
ราชบุรี
Sisophon
Battambang
กุงบาตดำบง
Phetchaburi
เพชรบุรี
Si Racha
ศรีราชา
Hua Hin
หัวหิน
Rayong
ระยอง
Chanthaburi
จันทบุรี
Pursat
Buôn Ma Thuột
Nha Trang
Kampong Cham
Đà Lạt
Cam Ranh
Phnom Penh
Phan Rang-Tháp Chàm
Doun Kaev
Phan Thiết
Thành phố Hồ Chí Minh
Kampot
Châu Đốc
Long Xuyên
Mỹ Tho
Vũng Tàu
Rạch Giá
Cần Thơ
Sóc Trăng
Bạc Liêu
Cà Mau
Surat Thani
สุราษฎร์ธานี
Nakhon Si Thammarat
นครศรีธรรมราช

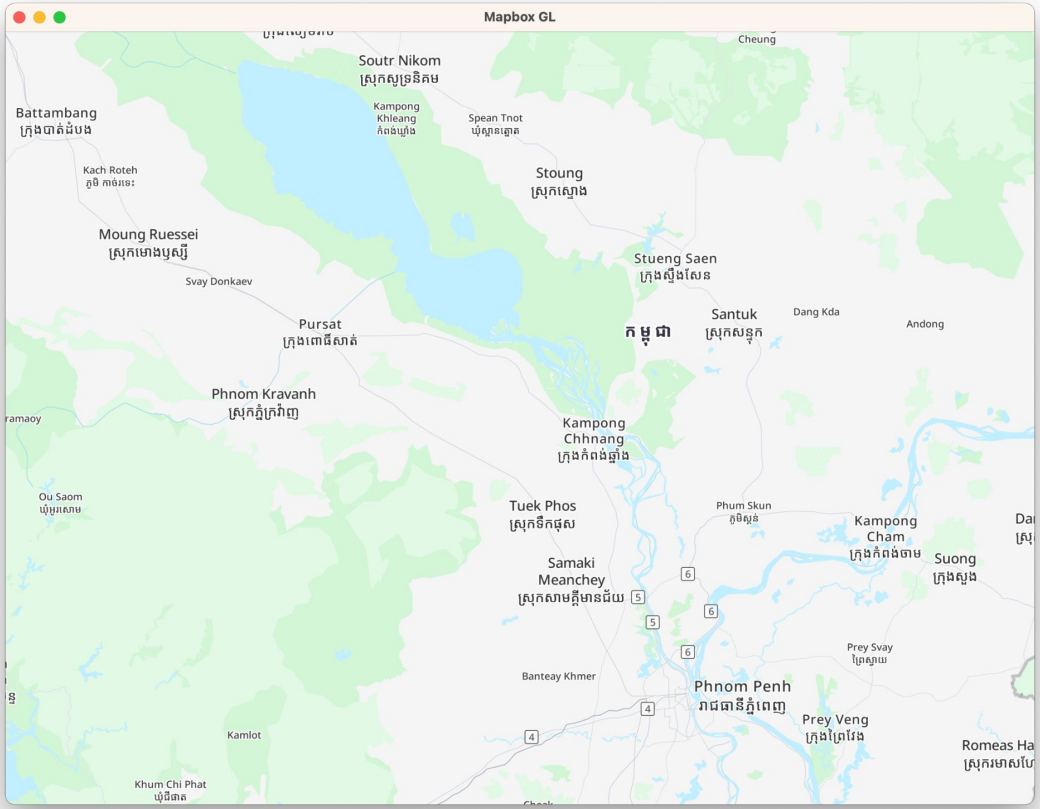
Effect


Mapbox GL
VIỆT NAM
ประเทศไทย
กัมพูชา
Nakhon Sawan
นครสวรรค์
Chaiyaphum
ชัยภูมิ
Ubon Ratchathani
อุบลราชธานี
Pakse
Surin
สุรินทร์
Quảng Ngãi
Phra Nakhon Si Ayutthaya
พระนครศรีอยุธยา
Kon Tum
Pleiku
Bangkok
กรุงเทพมหานคร
Quy Nhơn
Ratchaburi
ราชบุรี
Sisophon
ក្រុងសិរីសោភ័ណ
Phetchaburi
เพชรบุรี
Si Racha
ศรีราชา
Battambang
ក្រុងបាត់ដំបង
Hua Hin
หัวหิน
Rayong
ระยอง
Chanthaburi
จันทบุรี
Pursat
ក្រុងពោធិ៍សាត់
Buôn Ma Thuột
Nha Trang
Đà Lạt
Cam Ranh
Phnom Penh
រាជធានីភ្នំពេញ
Phan Rang-Tháp Chàm
Doun Kaev
ក្រុងដូនកែវ
Phan Thiết
Thành phố Hồ Chí Minh
Kampot
ក្រុងកំពត
Châu Đốc
Long Xuyên
Mỹ Tho
Vũng Tàu
Rạch Giá
Cần Thơ
Sóc Trăng
Bạc Liêu
Cà Mau
Surat Thani
สุราษฎร์ธานี
Nakhon Si Thammarat
นครศรีธรรมราช

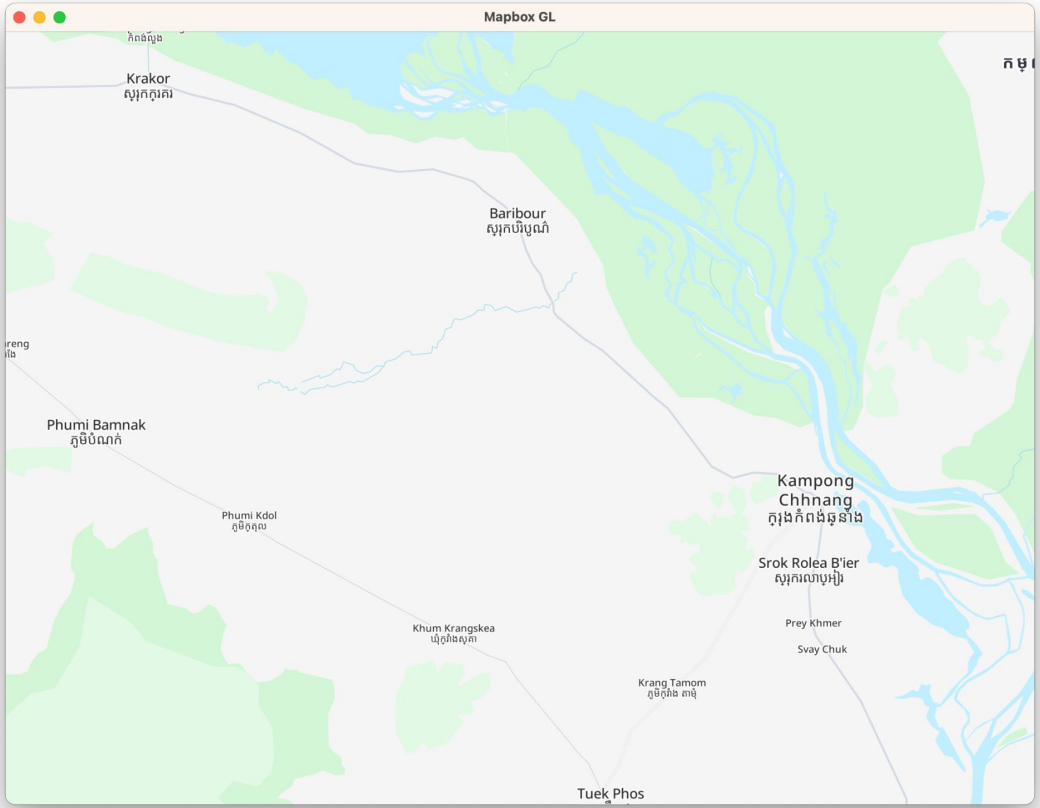
Effect


Mapbox GL
Soutr Nikom
Battambang
Kampong Khleang
Spean Tnot
Kach Roteh
Stoung
Moung Ruessei
Svay Donkaev
Stueng Saen
Pursat
Santuk
Dang Kda
Andong
Phnom Kravanh
Kampong Chhnang
Ou Saom
Tuek Phos
Phum Skun
Kampong Cham
Kampong Tralach
Koh Sotin
Tbeng Khpuos
Prey Svay
Banteay Khmer
Phnom Penh
Prey Veng
Kamlot
Khum Chi Phat

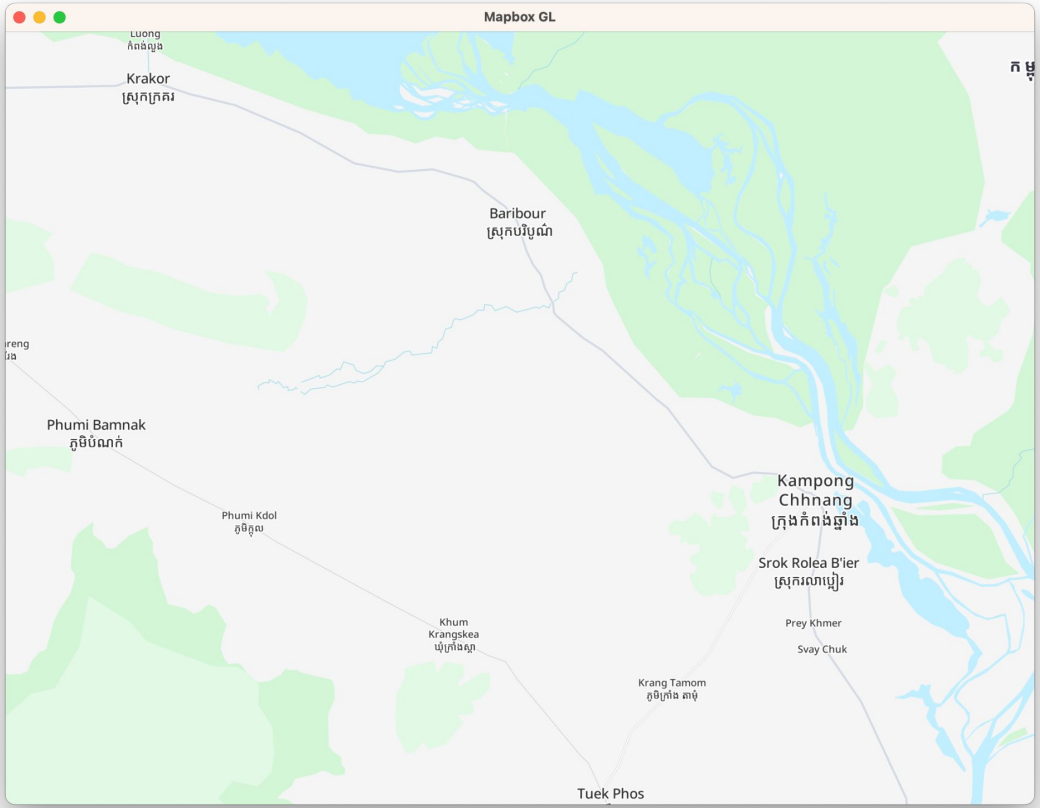
Effect


Mapbox GL
Soutr Nikom
ស្រុកសូទ្រនិកម
Battambang
ក្រុងបាត់ដំបង
Kampong Khleang
កំពង់ឃ្លាំង
Spean Tnot
ឃុំស្ពានត្នោត
Kach Roteh
ភូមិ កាច់រេះ
Stoung
ស្រុកស្ទោង
Moung Ruessei
ស្រុកមោងឬស្សី
Svay Donkaev
Stueng Saen
ក្រុងស្ទឹងសែន
Pursat
ក្រុងពោធិ៍សាត់
Santuk
ស្រុកសន្ទុក
Dang Kda
Andong
ក ម្ពុ ជា
Phnom Kravanh
ស្រុកភ្នំក្រវាញ
Kampong Chhnang
ក្រុងកំពង់ឆ្នាំង
Ou Saom
ឃុំអូរសោម
Tuek Phos
ស្រុកទឹកផុស
Phum Skun
ភូមិស្គន់
Kampong Cham
ក្រុងកំពង់ចាម
Suong
ក្រុងសួង
Samaki Meanchey
ស្រុកសាមគ្គីមានជ័យ
Prey Svay
ព្រៃស្វាយ
Banteay Khmer
Phnom Penh
រាជធានីភ្នំពេញ
Prey Veng
ក្រុងព្រៃវែង
Kamlot
Khum Chi Phat
ឃុំជីផាត

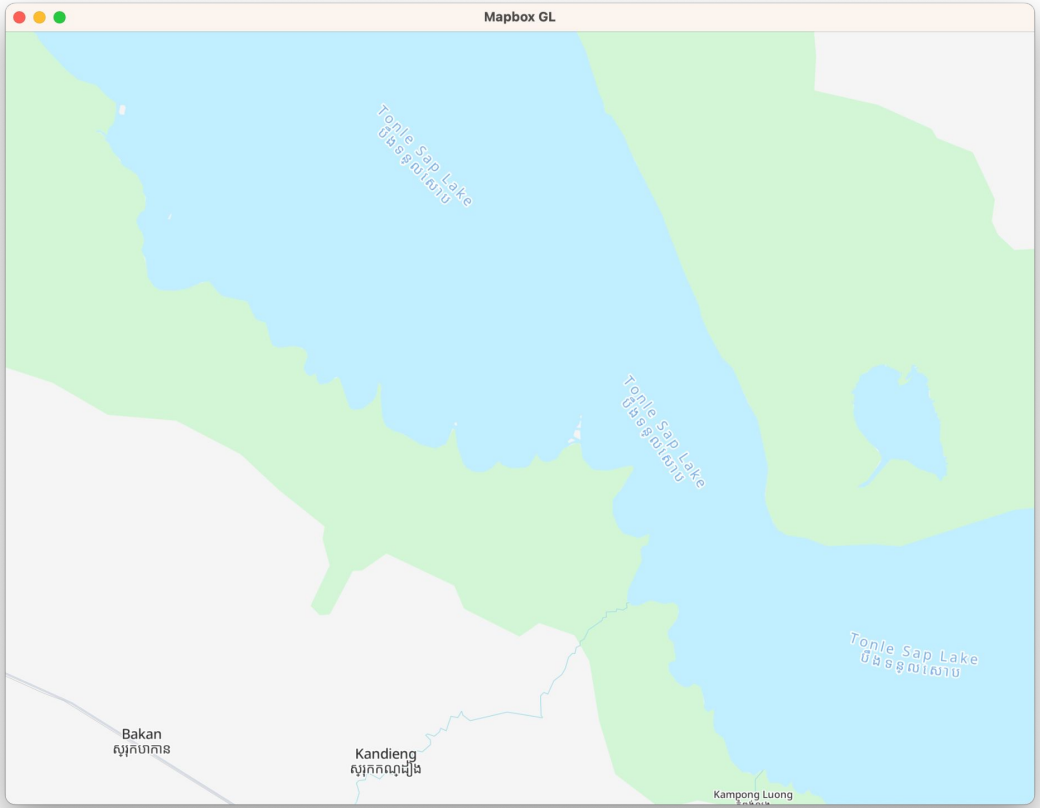
Effect


Mapbox GL
Krakor
Baribour
Phumi Bamnak
Phumi Kdol
Kampong Chhnang
Srok Rolea B'ier
Prey Khmer
Svay Chuk
Khum Krangskea
Krang Tamom
Tuek Phos

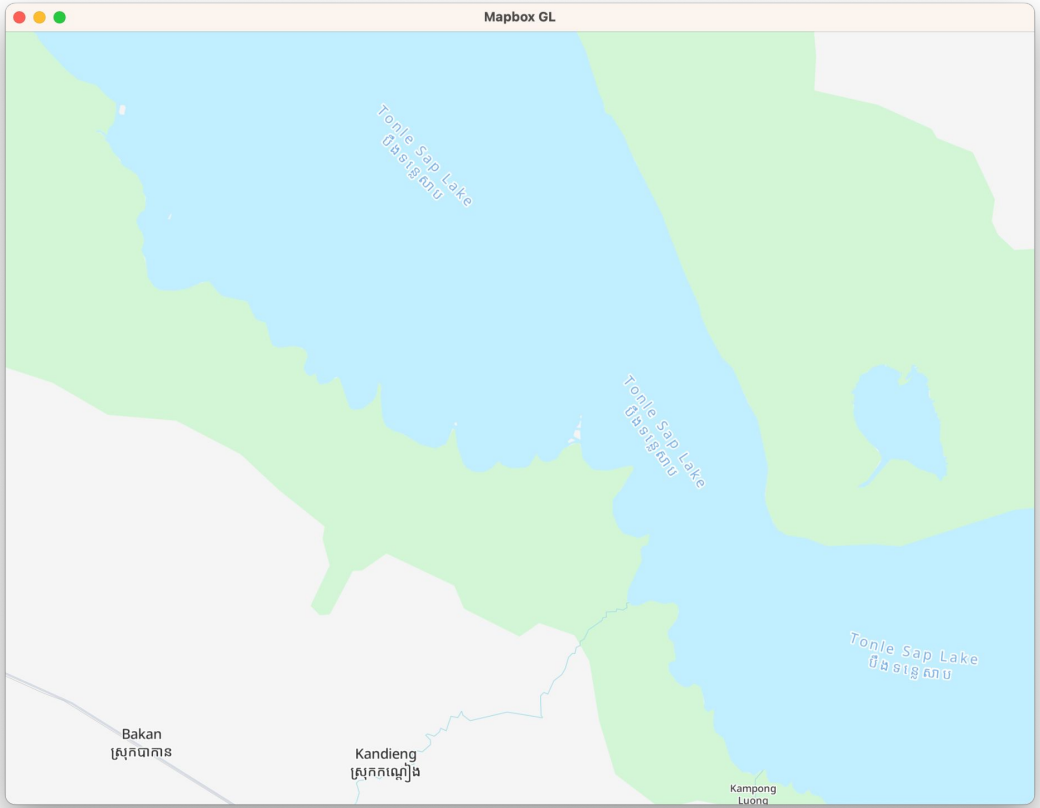
Effect


Mapbox GL
Krakor
ស្រុកក្រគរ
Baribour
ស្រុកបរិបូណ៌
Phumi Bamnak
ភូមិបំណាក់
Phumi Kdol
ភូមិគួល
Kampong
Chhnang
ក្រុងកំពង់ឆ្នាំង
Srok Rolea B'ier
ស្រុករលាប្អៀរ
Prey Khmer
Svay Chuk
Khum
Krangskea
ឃុំក្រាំងស្គរ
Krang Tamom
ភូមិក្រាំង តាមុំ
Tuek Phos

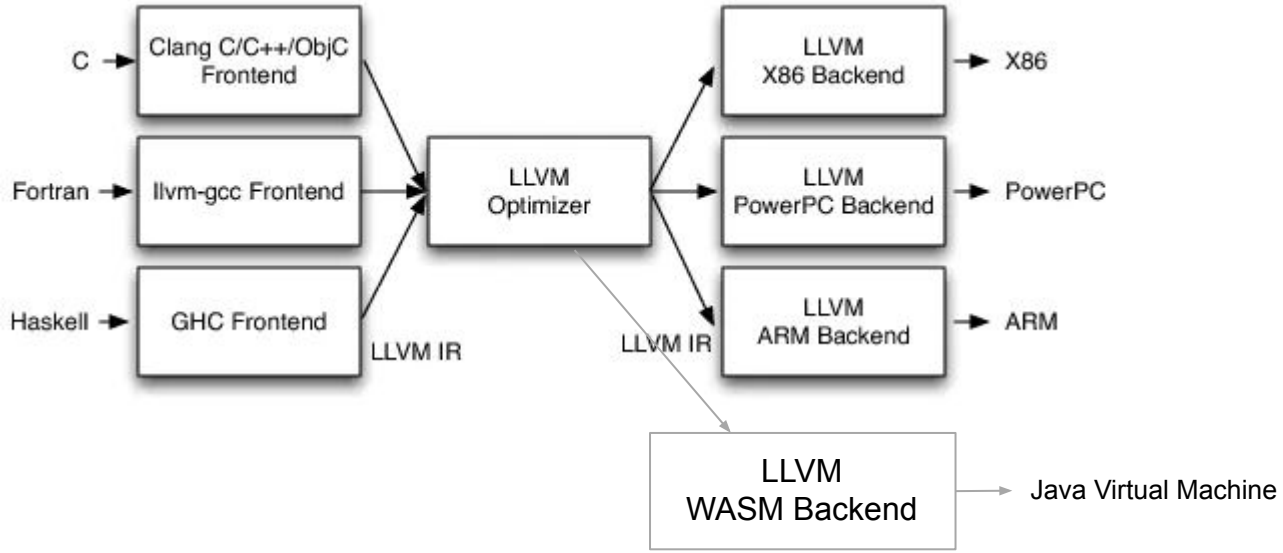
Effect


Mapbox GL
Tonle Sap Lake
បឹងទន្លេសាប
Tonle Sap Lake
បឹងទន្លេសាប
Tonle Sap Lake
បឹងទន្លេសាប
Bakan
ស្រុកបាកាន
Kandieng
ស្រុកកណ្តៀង
Kampong Luong

Effect


Mapbox GL
Tonle Sap Lake
Tonle Sap Lake
Tonle Sap Lake
Bakan
Kandieng
Kampong
Luong

WASM


C
Clang C/C++/ObjC Frontend
Fortran
llvm-gcc Frontend
Haskell
GHC Frontend
LLVM IR
LLVM Optimizer
LLVM IR
LLVM X86 Backend
X86
LLVM PowerPC Backend
PowerPC
LLVM ARM Backend
ARM
LLVM WASM Backend
Java Virtual Machine

HarfBuzzjs key snippet

```
// We could use instantiateStreaming but it's not supported in Safari yet
// https://bugs.webkit.org/show_bug.cgi?id=173105
fetch("hb.wasm").then(function (x) {
 return x.arrayBuffer();
}).then(function (wasm) {
 return WebAssembly.instantiate(wasm);
}).then(function (result) {
 result.instance.exports.memory.grow(400); // each page is 64kb in size
 window.hb = hbjs(result.instance);
 fetch('Noto Sans Medium Khmer.ttf').then(function (res) {
   return res.arrayBuffer();
 }).then(function (blob) { // can be used without our wrapper
   window.fontBlob = new Uint8Array(blob);
   console.log("Font blob loaded.")
   CreateHBGlyphs();
 });
});
```

HarfBuzzjs key snippet

```
// Source code: https://alanchenboy.github.io
var blob = hb.createBlob(fontBlob);
var face = hb.createFace(blob, 0);
var font = hb.createFont(face);
font.setScale(1200, 1200); // Optional, if not given will be in font upem
let scale = 0.02;


var buffer = hb.createBuffer();
buffer.addText(testText);
buffer.guessSegmentProperties();
// buffer.setDirection('ltr'); // optional as can be by guessSegmentProperties also
hb.shape(font, buffer); // features are not supported yet
var result = buffer.json(font);


result.forEach(function (x) {
      var glyphJson = font.glyphToJson(glyph.glyphIndex);
      // glyphJson.values  is SVG data
      // x.ax : advance
      // x.dx : x offset
      // y.dy : y offset
      // y.g  : Glyph Index
}
```

Thanks for Watching