linkur á verkefnið

## Fríða

Ég byrjaði á því að útfæra einfaldan *"player controller"* fyrir Fríðu. Þessi útfærsla nýtti offset breytu í punktalitaranum til þess að hreyfa Fríðu og virkaði vel. Það tók mig þó ekki langan tíma að endurhugsa þessa útfærslu þegar ég fór að snúa Fríðu.

```
attribute vec4 vPosition;
uniform vec4 offset;
uniform float rotation;

void
main() {
  gl_PointSize = 20.0;
  gl_Position = vPosition + offset;
}
```

```
// frog
gl.bindBuffer(gl.ARRAY_BUFFER, frogBuffer);
gl.vertexAttribPointer(vPosition, 2,
gl.FLOAT, false, 0, 0);

gl.uniform4f(colorLoc, 0.1, 0.6, 0.0, 1.0);
gl.uniform4f(offsetLoc, offset[0],
offset[1], 0, 0);
gl.drawArrays(gl.TRIANGLES, 0, 3);
```

Útfærsla tvö reiðir meira á javascript og nýtir það að hægt sé að yfirskrifa buffer ef búið er að skilgreina á honum stærðina. Fyrir verkefni 2.4 hafði ég búið til aðferð til að smíða þríhyrning útfrá miðjuhnitum og radíus, nú bætti ég við breytu sem samsvaraði til snúningi þríhyrningsins. Þá fyrir hvert kall á render fallið gat ég yfirskrifað núverandi Fríðu með nýrri staðsetningu og snúningi.

```
// frog
gl.bindBuffer(gl.ARRAY_BUFFER, frogBuffer);
gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
gl.bufferSubData(
  gl.ARRAY_BUFFER,
  0,
  flatten(tri_from_coords(offset[0], offset[1], 0.04, rotation))
);

gl.uniform4f(colorLoc, 0.1, 0.6, 0.0, 1.0);
gl.uniform2f(offsetLoc, 0.0, 0.0);
gl.drawArrays(gl.TRIANGLES, 0, 3);
```

## Umverfi og umferð

Næst á dagskrá var umhverfið, ég vildi að Fríða hefði öruggan stað til að vera á *(brimborg?)* á miðri götu. Ég setti inn þrjá kassa sem breytast aldrei, þeir eru teiknaðir á undan *næstum öllu* og eru því alltaf á bakvið.

En öruggir staðir geta ekki verið öruggir án þess að hafa hættur handan við hornið þessvegna voru bílarnir næst á dagskrá. Upphaflega reyndi ég að nota `Math.sin()` og ehv sniðugar brellur til þess að fá bílanna til að fara fram og til baka. Ég gafst fljótlega upp á því og byrjaði að nota móduló reikning sem virkaði eins og í sögu. Bílarnir kalla á aðferð til að fá staðsetninguna sína útfrá því á hvaða vegi þeir eru, hvar þeir eru í röðinni á veginu og tíma. Þessir útreikningar eru svo líka notaðir seinna til þess að athuga hvort bíll hafi keyrt á Fríðu.

```
const offsets = (sect: number, road: number, car: number, time: number) => {...}
```

# Aukastig

## Fleiri en einn bíll á sömu braut

Upphaflega hafði ég lesið kröfurnar um bíla vitlaust og gerði ráð fyrir því að það þyrfti fleiri en einn bíl á hverjum vegi. Þetta gerði brekkuna aðeins brattari í byrjun en borgaði sig fljótt þar sem það eina sem ég þurfti að gera til að auka bíla var að hækka global breytuna `num_cars` og passa að offset á milli þeirra héldist jafnt.

## Bílarnir eru af mismunandi lit

Kóðinn fyrir þetta var frekar einfaldur, ég hafði áður bætt við litabreytunni `uniform vec4 color` inn í bútalitarann svo að Fríða og mýrin væru ekki eins á litinn. Vegna þess að bílarnir mínir voru allir smíðaðir inn í nokkrum *for* lykkjum, þá gat ég einfaldlega breytt `color` útfrá því á hvaða vegi bíllin var.

```
...
const traffic = (time: number, gl: WebGLRenderingContext) => {
    for (let section = 0; section < 2; ++section) {
      for (let road = 0; road < num_roads / 2; ++road) {
        gl.uniform4f(colorLoc, 0.6, 0.2 * road, 0.0 + 0.5 * section, 0.8);
        for (let car = 0; car < num_cars; ++car) {
          // sækja upplýsingar um offset í sér fall
          const [x_off, y_off] = offsets(section, road, car, time);
          gl.uniform4f(offsetLoc, x_off, y_off, 0, 0);
          gl.drawArrays(gl.TRIANGLE_FAN, 0, 4);
          ...
```

## Fjölga brautum upp í fimm

Hér eins og með *Fleiri en einn bíll á sömu braut* var lausnin eiginlega komin áður en ég áttaði mig á vandamálinu. Hér notast ég við global breytuna `num_roads`, ég endaði á að setja hana sem sex frekar en fimm einfaldlega til að hafa symmetríu á milli svæðana.

## Halda utan um stig með strikum

Þetta var síðasta viðbótin við leikin, ég notaði sömu aðferð og ég notaði til að teikna bílanam til þess að teikna stigakassa inn í buffer.

```
gl.bindBuffer(gl.ARRAY_BUFFER, pointBuffer);
gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
gl.uniform4f(colorLoc, 0.0, 0.0, 0.0, 1.0);
for (let i = 0; i < counter; ++i) {
  gl.uniform4f(offsetLoc, i * 0.03, 0, 0, 0);
  gl.drawArrays(gl.TRIANGLE_FAN, 0, 4);
}
```

Ég held síðan utan um þennan eina buffer með einu striki, þetta strik er síðan teiknað einu sinni fyrir hvert stig. Þetta gerir það hentugt að nota for lykkju og setja hámark teikninga jafna og fjölda stiga.

# *Aukahlutir, mér til gamans*

Ég skemmti mér mikið við að gera þetta verkefni og bætti þessvegna við nokkrum aukahlutum sem ekki voru í kröfunum. Helstu hlutirnir eru:

- Blóðpollur sem birtist þegar bíll keyrir á Fríðu, þetta er hálf ógeðslegt en ég lít á þetta sem hvatningu til spilarans til þess að gera betur.
- Texti sem hvetur spilarann áfram því lengra sem hann kemst inn í leikinn
- Dauðateljari sem birtist þegar spilarinn deyr
- Auk erfiðleikastig eftir að spilari *"vinnur"* leikinn, þá hækkar hraðinn á bílunum