

(TÖL105M) Tölvugrafík

Kristján Leó Guðmundsson

15. september 2023

Uppsetning

Undir möppunni `GameObjects` má finna möppur sem innihalda allt sem er birt á `canvas`-inu, þ.e. `Frog`, `Road`, `Car`, `GrassBush`, og `PointCounter`. Einnig er þar skrá sem heitir `GameObject.js` en þeir hlutir sem eru teiknaðir `extend`-a klasann `GameObject` í henni. `GameObject` útfærir flest allt sem er sameiginlegt með öllum þessum hlutum eins og föll til að teikna, sækja texture og hluta-skrár, reikna árekstra, og mynda buffera.

Í möppunni `Utils` eru skrár sem innihalda ýmis hjálparföll. M.a. skrárnar `MV.js`, `webgl-utils.js` og `initShaders.js` sem voru notaðar í fyrri heimadæmum. Einnig var bætt við skránum `fetchObjects.js` sem hefur föll til að sækja hluta-skrár, `gameControls.js` sem hefur hjálparföll til að hlusta á örvatakkana, og `utils.js` sem hefur ýmis stærðfræði tengd föll til að reikna línur sem skera tvo punkta, fall til að reikna hvort tvö strík skerist o.fl.

Skráin `main.js` bindur þetta síðan saman og sér um að render-a á skjáinn o.fl.

Örvatakkarnir

Froskurinn færir fram, aftur, til hægri og til vinstri þegar viðeigandi örvatökkum er haldið inni. Með því að halda inni annað hvort upp eða niður örinni ásamt annað hvort hægri eða vinstri örinni getur froskurinn einnig farið á ská.

Árekstraskynjun

Til að reikna árekstra milli frosksins og bílanna er haldið utan um þau strík sem liggja utan um hvort tveggja (kallaði strikin `boundingLines` í kóðanum) og athugað hvort að einhver af þeim skerist. Strikin eru geymd sem lína með hallatöluna h og skurðpunktinn k ásamt min og max mörkum í x og y stefnu. Út frá þessum gildum er þægilegt að reikna mögulega skurðpunkta með því að leysa jöfnuhneppið:

$$\begin{aligned}y &= h_1 \cdot x + k_1 \\ y &= h_2 \cdot x + k_2\end{aligned}$$

og athuga hvort punktur falli innan markanna. En auðvitað eru einhver jaðartilvik sem þarf að athuga líka eins og t.d. þegar lína hefur alltaf sama gildið á x ($x = a$ þar sem a er fasti).



Mynd 1: Árekstur 1



Mynd 2: Árekstur 2



Mynd 3: Árekstur 3

Froskur upp og niður

Til þess að hafa útfærsluna á froskinum aðeins þæginlegri var notað tvo auka klasa fyrir bæði frosk á leiðinni upp (**FrogUp**) og frosk á leiðinni niður (**FrogDown**). Þessir klasar eru síðan notaðir í **Frog** klasanum.

Hvernig fást stig

Var ekki alveg viss hvort að stigin eigi að fást fyrir að komast yfir allar göturnar eða fyrir að komast yfir allar göturnar 'og til baka' en útfærslan er þannig núna að stig fást fyrir það eitt að komast yfir. Til að breyta því væri hægt að láta **FrogUp** klasann alltaf skila **false** í **scoreUpdate** fallinu.

Úfærðir aukaliðir

Allir aukaliðir hafa verið útfærðir:

Fjöldi bíla á vegi

Hægt er að stilla fjölda bíla á hverri braut þegar þær eru búnar til. **Road** klasinn tekur inn breytur **y**, **velocity**, **nCars** þar sem seinasta breytan segir til um fjölda bíla á brautinni.

Mismunandi litir á bílum

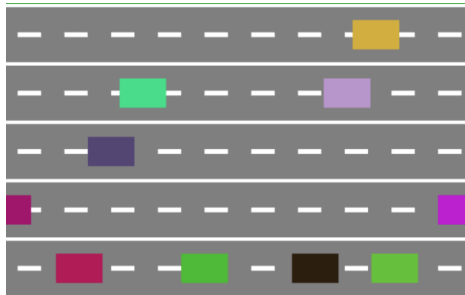
Bílunum eru gefnir slembnir litir þegar þeir eru búnir til.

```
vec4(Math.random(), Math.random(), Math.random(), 1.0)
```

Síðan er myndað texture með einum pixli sem hefur þann lit og breytunni **gl_FragColor** í bútalitaranum gefið það gildi þegar þeir eru teiknaðir.

Fjöldi brauta

Fimm brautir eru teiknaðar í leiknum en auðvelt er að annað hvort fjölga eða fækka þeim með því búa til hluti af klasanum **Road** í **createObjects** fallinu í **main.js**. Passa þarf samt að stilla **y** gildið rétt til þess að veginir skarist ekki.



Talning stiga

Klasinn **PointCounter** heldur utan um stigin og eru þau teiknuð á skjáinn. Þegar 10 stigum er náð er leiknum lokið.

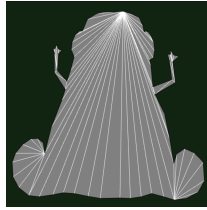


Texture o.fl.

Þar sem ég hafði soldinn auka tíma langaði mig að reyna að teikna flóknari frosk með fleiri punktum. Til þess gerði ég auka forrit sem leyfir manni að skilgreina hlut með endapunkta og skipta þeim upp í þríhyrninga. Forritið leyfir manni einnig að teikna 'texture' á hlutinn. Hnúttunum og fleiru er síðan hægt að hlaða niður sem JSON skjali og texture-inu sem PNG mynd.



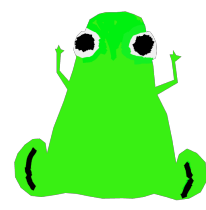
Mynd 4: Útlínur frosksins teiknaðar



Mynd 5: Formi skipt upp í þríhyrninga



Mynd 6: Texture teiknað á froskinn



Mynd 7: Útlínur nýttar

Þar sem við vorum ekki búin að læra hvernig maður setur texture á hluti þá fann ég dæmi hjá Mozilla (Hlekkur á Mozilla tutorial) og myndband á Youtube (Hlekkur á Youtube tutorial) sem var mjög hjálplegt til að skoða og læra af en auðvitað útfærði ég síðan verkefnið sjálfur.

Til þess að geta birt bæði hluti með texture og hluti með einum lit þurfti einhvernvegin að gefa breytunni `gl_FragColor` í bútalitaranum rétt gildi í báðum tilfellum. Ég sá að það var talað um trick þar sem texture-ið og litabreytan eru annað hvort margfölduð eða lögð saman í bútalitaranum hérna (Hlekkur á Stackoverflow spjallþráð). Til að birta litinn er texture-ið sett sem hvítur litur og til að birta texture-ið er litabreytan sett sem hvítur litur. Þegar margfaldað er saman virkar hvíti liturinn sem hlutleysa.

Núna þegar ég var að skrifa þetta datt mér í hug að það væri sniðugara að setja bara texture-ið sem einn pixil með litnum á hlutnum til þess að sleppa við margföldunina og það virðist virka svo að útfærslan er þannig núna, finnst þínu skýtið að ég hafi ekki fattað þetta fyrr hehe.

Vefsíða

Hérna er hlekkur á vefsíðuna: <https://tolvugrafik-test.netlify.app/v1/>