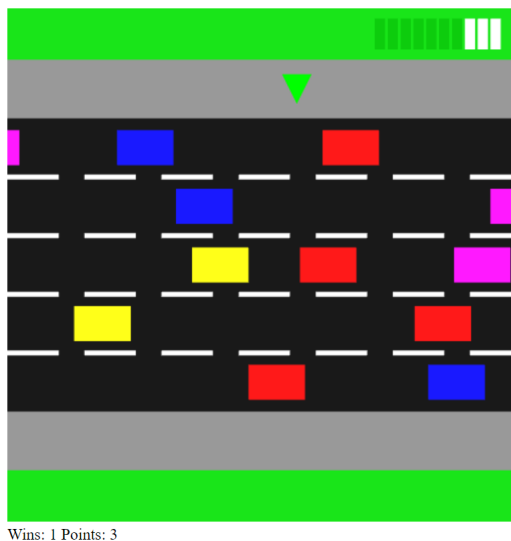


TÖL105M - Verkefni 1 Skýrsla

Inngangur

Ég skráði mig í þennan áfanga vegna áhuga á að læra að búa til litla leiki og herma fyrir kennslu þannig að það var mjög ánægjulegt að fyrsta verkefnið sé einfaldur leikur. Í upphafi verkefnis var ég ekki viss um hvaða stærð og stærðarhlutföll væri best að hafa á frogger leik þannig að ég ákvað að hafa smíði leikhlutanna að miklu leiti stíkaða þannig að auðvelt væri að stilla útlit og lögun leiksins út frá nokkrum stikum. Ég áttaði mig hins vegar ekki strax á þeirri þjögun sem á sér stað þegar hlutföllum canvas elementsins er breytt og tók því stikunin ekki tillit til þess sem takmarkaði notagildi hennar. Þrátt fyrir það er hægt að breyta ýmsu, s.s. heildarstærð götunnar, fjölda akreina og stærð frosksins á einfaldan hátt í kóðanum. Markmiðið var einnig að útfæra allar viðbæturnar og tókst það á endanum.



Lausn

Útlit leiksins er í grunninn þannig að efst og neðst er lítið grassvæði, næst kemur gangstétt og síðast vegurinn. Þetta er útfært með því að gefa `clearColor` grasgrænt gildi og teikna síðan stóran miðjaðan gráan kassa (gangstéttin) sem er aðeins minni en heildarhæð canvas elementsins og ofan á hann er teiknaður miðjaður svartur kassi (gatan) sem er með enn minni hæð. Þessi smíði er stíkuð þannig að stærðin passar við fjölda akreina sem stillt er í breytu. Auðvelt var því að útfæra viðbótina sem jók fjölda akreina í 5. Til þess að aðgreina sýnilega akreinarnar eru ofan á götuna teiknaðar brotalínur milli akreinanna. Ekkert af ofangreindu breytist á meðan leikurinn er í gangi og er einfaldlega teiknað með þremur köllum á `drawArrays` og litað með `uniform4fv` breytum.

Froskurinn sjálfur er einn þríhyrningur og er staðsetning hans reiknuð út frá breytu sem heldur utan um þá akrein sem hann er á og annarri sem geymir í hvaða dálki hann er (færsla til vinstri og hægri). Þegar hann er teiknaður er staðsetning hans reiknuð út frá þessum breytum og sett í `uniform2fv` breytu sem bætt er við `x` og `y` gildi hnúta frosksins í hnútalitara. Stefna frosksins ræðst af `uniform1f` breytu sem hefur gildið 0 þegar froskurinn stefnir upp en π þegar hann stefnir niður og er sú breyta notuð sem horn sem hnútum frosksins er snúið um í hnútalitara.

Bílarnir eru einfaldir ferhyrningar og eru upphafstaðsetningar þeirra skilgreindar í tvívíðu fylki þar sem fyrsta víddin táknar þá akrein sem þeir eru á og seinni víddin geymir hnit fyrir hvern bíl á þeirri akrein (auðvelt er því að fjölga bílum). Í render falli er hreiðruð lykkja þar sem `x`-hnit bílanna eru uppfærð í samræmi við gildi sem geymir hraða hvernar akreinar. Niðurstaðan er síðan send sem `uniform2fv` breyta til hnútalitara. Fylki með nokkrum litum gefur bílunum lit í sömu lykkju í gegnum `uniform4fv` breytu. Til þess að þurfa ekki að fara í gegnum þetta bílafylkið oftar einu sinni eru árekstrar bíla og frosks einnig reiknaðir í þessari lykkju.

Stigafjöldi er táknður bæði með tölu fyrir neðan canvas elementinu og sem strik efst til hægri í leiknum. Strikin 10 eru ávallt öll teiknuð en litur þeirra ræðst af stigafjölda þess sem spilar. Þannig að einungis sá fjöldi stiga er skýrt litaður en hin eru dökk og gefa þá til kynna hversu mörg stig á eftir að safna.

Sigur eða árekstur eru gefin til kynna með alert meldingum og staða leiksins uppfærð samkvæmt því.

Lokaorð

Ýmislegt kom upp á við lausn verkefnisins. Það að reyna að stika útlit leiksins í stað þess að harðkóða leiddi til ýmissa útreikninga sem gat verið erfitt að ná réttum. Þetta ásamt því að flest sem notað er í render fallinu er upphafsstillt í init olli því að mjög mikið er um global breytur sem er ekki mjög snyrtilegt. Hugsanlega væri betra að setja þær í JS hlut og senda hann sem viðfang í render fallinu. Annað sem olli smá vandkvæðum var samverkun `requestAnimationFrame` og `setTimeout` en lengi birtist melding um tap eða sigur áður en skjámyndin uppfærðist. Þetta komst í lag á endanum en ég er ekki viss um að ég skilji alveg virknina.

Það sem ég græddi þó mest á við lausn verkefnisins var að fá betri skilning á WebGL sérstaklega með tilliti til hvernig bufferar virka, hvenær þarf að kalla á, og hlutverki, `bindBuffer` og aðeins betri innsýn í hnútalitara og notkun `uniform` breyta.

Tenglar

<https://bjarnits.github.io/tol105m/verkefni1/verk1.html>

<https://bjarnits.github.io/tol105m/verkefni1/verk1.js>