

Tölvugrafík Verkefni 1

Ragnar Björn Ingvarsson, rbi3

15. september 2024

<https://skogarbjorn.github.io/h2/test1/index.html>

Hluti I

Verkefnislýsing

Beðið var um að gera leikinn duck hunt í webgl með nokkrum forsendum. Það þarf að hafa þríhyrning sem spilara, sem við getum stjórnað með því að ýta á hann og draga með músinni. Fuglar (ferhyrningar) eiga að fljúga yfir skjáinn á mismunandi hraða, og spilari á að getað skotið skoti (ferhyrningur) með því að ýta á takka, sem flýgur beint upp. Ef skotið klessir á fugl, þá hverfur bæði fuglinn og skotið og við fáum stig sem er strik efst í glugga. Ef við erum komin með 5 stig þá endar leikurinn.

Hluti II

Verkefnið mitt

Ég notaði eitt points fylki fyrir alla punkta sem þarf að geyma og hef nokkrar breytur fyrir hvar fuglarnir, skotin og stigin byrja í fylkinu. Á þennan hátt er þægilegt að halda utan um hvernig skipulag fylkisins er með því að hækka viðeigandi breytur þegar skot eða fugl er bætt við fylkið.

Byrjum á game fallinu. Ég einfaldlega teikna fyrst það sem er í buffernum, svo uppfæri ég hvert skot fyrir sig, svo fuglana, og að lokum reyni ég að búa til nýjan fugl með `Math.random()` og tékka hvort það sé minna en 0.01.

`render` fallið virkar svo að ég teikna fyrst spilarann, sem sagt fyrstu þrjá punktana sem þríhyrning, svo hvert skot fyrir sig, sem ég nota `shot_index` í, sama með fuglana og svo stigin, allt í mismunandi for lykkjum til að halda hlutum aðskildum.

Ég einnig nota spacebar til að skjóta skotum, og hef sett takmörkunina að ekki er hægt að hafa fleiri en þrjú skot á skjánum í einu.

Að uppfæra staðsetningu fuglanna og skotanna er nokkuð einfalt, ég bara fer í gegn um alla punkta sem tilheyra fuglunum og bæti smá gildi við `x` gildið. Eins með skotin. Síðan tékka ég hvort þau séu farin af skjánum og ef svo er fyrir eitthvert skot eða fugl, þá tek ég punktana úr points með `splice` fallinu og uppfæri allar index breytur. Svo nota ég `break` til að vera viss um að ekki ruglast þar sem ég breyti points á meðan ég er í for lykkju sem fer í gegn um points.

Fyrir collision þá nota ég AABB algorithmann, og tékka fyrir hvert skot, fyrir alla fugla, ef skotið er komið á efri helming skjásins (þar sem fuglarnir eru). Ef eitthvert skot klessist á við einhvern fugl, þá eyði ég báðum og breaka þar sem skotið sem ég var að tékka á er nú farið.

Ég lenti í nokkrum örðugleikum við að forrita þetta, en aðallega var vesen að láta fuglana fljúga á mismunandi hraða. Ég leysti það með því að hafa annað fylki sem heldur utan um hraða allra fuglanna. Vesnið kemur þó af því hvernig ég á að tengja hvern fugl þegar

ég uppfæri staðsetningu þeirra við hvaða hraða hann á. Til að koma til móts við þessu notaði ég sérhæfðar for lykkjur sem byrja á 0 og hækka upp í fjölda fugla. Síðan, inni í lykkjunni, þá nota ég alltaf $birds_index + i * 4$ til að færa fuglinn, og er svo með i sem er index fuglsins í hraða-fylkinu.

Loks, þá tékka ég í hverri lykkju af `game`, hvort stigin eru færri en fimm, og ef þau eru fimm eða hærri þá kalla ég á render einu sinni enn til að hafa lokaskjáinn réttan, og svo endurtek ekki aftur `game`.

Hluti III

Viðbætur

Ég semsagt útfærði allar viðbæturnar sem lýst var að væri hægt að bæta við leikinn í verkefnislýsingunni. Forritið getur höndlað nær ótakmarkað marga fugla á skjánum í einu, og þeir fljúga allir í mismunandi hæð og á mismunandi hraða. Einnig er hægt að hafa upp að þremur skotum á skjánum í einu og þau virka öll eins og þau eiga að gera. Og loks birtast strik (mjóir ferhyrningar) efst í hægra horni skjásins þegar skot hittir fugl, og ef fimm strikum er náð þá endar leikurinn.