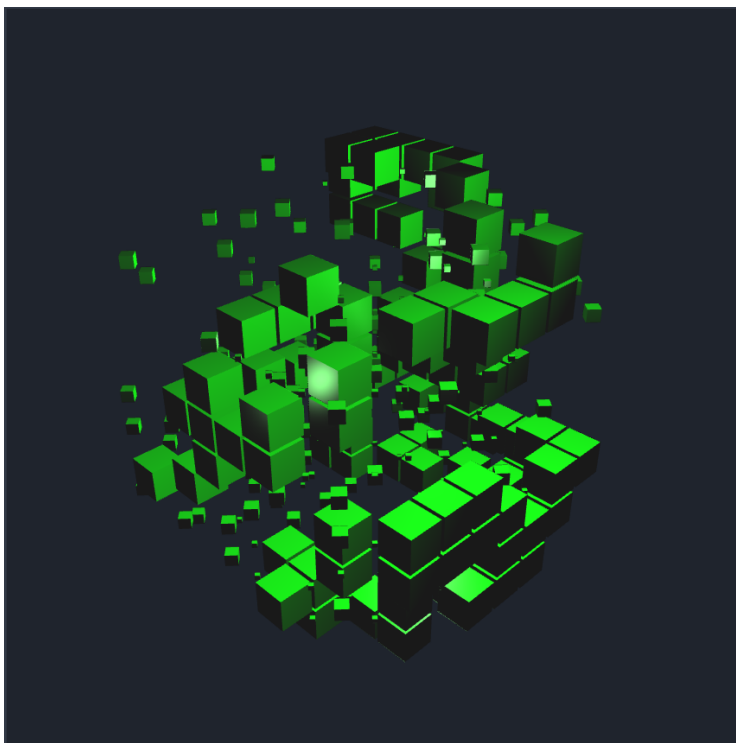


Verkefni 2

TÖL105M

Egill Magnússon

Verkefnið má finna hér: <https://egillm.is/verkefni/tolvugrafik/v2/3d-game-of-life>



Mynd 1: Skjáskot af leiknum í miðri breytingu.

Kóðinn

Allt umhald fyrir leikin er JavaScript megin. Skilgreyni $10 \times 10 \times 10$ sem geymir 1 fyrir lifandi cell og 0 fyrir dauða cell. Upphafstilli grindinda með 30% lifandi og 70% dautt geri það með fallinu `createGrid()`. Upphafstilli einum tening í buffer eins og er gert í sýnidæmi með H í þrívídd, nema í stað þess að vera með color buffer, er ég með normal buffer sem geymir normal fyrir hvert hnút/vertex, normallinn er normall á flötin sem þríhyrningurinn sem hnúturinn er partur af myndar og er notaður í shaderum. Fyrir animation-in geymir ég `prevGrid` sem geymir fyrrum stöðu, upphafstilli það sem tomt $10 \times 10 \times 10$ grind.

Render fallið

Efst í render fallinu er ég með kóða sem breytir sýnisvörpununni úr orthographic í perspective. Fyrir utan það er render fallið frekar einfalt, skilgreini globalTransform fylki, sem heldur utanum snúning á grindinni, ásamt perspective vörpununni. Næst athuga ég hvort komin sé tími á að uppfæra stöðu leiksins, sá tími er breytilegur eftir stöðu á slider ef komin er tími til að uppfæra kalla ég á updateGrid. Næst er ég með íf setningu sem ákveður hvort teikna á teninga með hreyfingu/animation eða ekki. Ef tími frá síðustu uppfærslu á leiknum er minni en animationDuration þá er kallað á renderGrid með animation progress og rotation. Animation progress er $(currentTime - lastUpdatetime) / animationDuration$, og rotation er $animationProgress \cdot 2\pi$.

updateGrid()

updateGrid() sér um að uppfæra stöðu grindarinnar. Það loopar yfir allar cell-ur og kallar á fallið countNeighbours() sem telur fjölda nágranna, svo út frá fjölda nágranna ákveður það hvort cell-a sé lifandi í næstu ítrun eða ekki. Nota sömu reglur og vöru gefnar í verkefnalýsingu.

countNeighbours()

Loopar yfir allar cellur og telur fjölda nágrana cell-a sem eru lifandi.

renderGrid()

Loopar yfir allar cellur og ber saman núverandi stöðu og prev-stöðu. Skiptist í 3 tilfelli:

1. Cell-a er lifandi í núverandi og var lifandi í prev-stöðu: Þá teikna tening með drawCube().
2. Cell-a er lifandi núna en var ekki lifandi í prev-stöðu: Þá animate-a teninginn inn, þ.e. kalla á drawAnimatedCube() með $scale = progress$.
3. Cell-a er dauð núna en var lifandi í prev-stöðu: Þá animate-a teninginn út, þ.e. kalla á drawAnimatedCube() með $scale = 1 - progress$.

drawCube()

Býr til vörpunarfylki, bætir við hliðrun út frá hnitum í grindinni. Bætir svo við globalTransform við vörpunina og teiknar svo tening.

drawAnimatedCube()

Gerið það sama og drawCube() Nema eftir hliðrunina bætir við snúning og skölun, þ.e. skalar um scale og snýr um rotation. Bætir svo globalTransform við vörpunina og teiknar svo tening.

Shading

Ákvað að láta reyna á að útfæra Phong-lýsingu, og gekk ágætlega. Er með uniform breytur: lightPos, viewPos, ambientColor, diffuseColor, specularColor og shininess.