

Einar Guðni Guðjónsson & Viktor Þór Freysson

Egg18@hi.is – vthf1@hi.is

Tölvugrafík

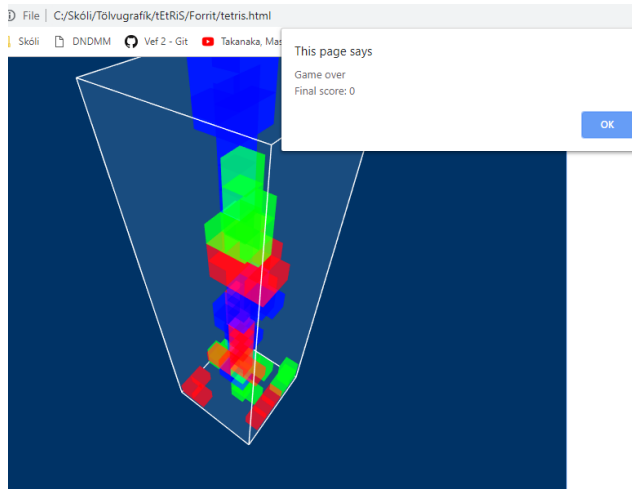
Prítris

Í þessari útfærslu af Tetris höfum við gert Tetris í þrívídd með einungis tveimur gerðum af kubbnum. Þessum langa (3 kassar) og þann sem er eins og L í laginu nema með eina miðju, einn til hægri og einn á toppnum. Þessi útfærslu kallast því Prítris. Útfærslan gekk nokkuð vel og er nóg af kóða til að fara yfir. Til að losna við rúmar 60 línur af kóða fyrir hnútana er sér javascript skjal sem heldur utan um hnútana. Við notumst við *transparent* kassa eða gegnsæjan kassa sem er notaður sem leikvöllurinn. Kubbarirnir koma í handahófskenndri röð niður frá toppnum. Hægt er að notast við örvarnar á lyklaborðinu til að færa kubbana til hægri og vinstri eða í raun um z-ásinn. Enter takkinn byrjar leikinn og bilstöngin lætur kubbana detta hraðar.

Hér hægra megin sjáum við stuttan forritsbút sem gerir fallið `dropDown` sem kallar á fallið `moveBlock` og sendir neikvæða tölu til að hraða fyrir fallið. Lykill 32 er bilstöngin og er restin nokkuð sjálfútskýrandi.

Einnig er lítill fítus sem pásar leikinn með að ýta á “p”. Músin snýr svo kassanum og getur verið mikilvægt að nýta sér það til að betur sjá hvar kubbarirnir detta niður. Mikilvægt er líka að nýta sér takkana **A/Z** til að snúa kubbnum á X-ás. **S/X** til að snúa kubbnum á Y-ás. Svo **D/C** til að snúa kubbnum um Z-ás. Í hvert skipti sem leikmanni tekst að fylla í heila hæð bætist við stigafjöldinn og á sú hæð að hverfa. Það er ekki hægt að vinna leikinn en hinsvegar getur verið gaman að keppast um að ná hæstum stigafjöld. Leikurinn tapast þegar kubbarirnir ná alla leið upp í toppinn.

```
function dropDown(b) {  
  if (keys[32]) {  
    t = moveBlock(b, 'y', -1);  
  }  
  if (dropTimer) {  
    b = moveBlock(b, 'y', -1);  
    dropTimer = false;  
  }  
  return b;  
}
```



Hér má sjá skjáskot af þegar leikur tapast. Við erum með býsna einfalda athugun sem kannar hvort 3 mismunandi breytur séu komnar yfir 19. Það gefur til kynna að hæðin sé kominn upp í 20 og þar með tapast leikurinn eins og má sjá á hér á örfaúm línunum af kóða. Ef svo fer fær leikmaður skorið sitt og leikurinn endurstillir sig.

```
if (curr.center.y > 19 || curr.one.y  
> 19 || curr.two.y > 19) {  
  window.alert('Game over  
' + 'Final score: ' +  
document.getElementById("currScore").  
innerHTML);  
  reset();  
}
```

Skemmtilegra er að fara yfir hvað gerist þegar leikmenn ná í fylla út í heilu línu. Það má sjá í fallinu “deleteRow” hér í texta reitnum. Þetta fall athugar ekki hvort hafi verið fyllt í heila hæð heldur er kallað í fallið í “isFullPlane”. Í hvert skipti sem kubbur hefur náð snertinu ef svo má komast að orði er kallað í fall sem athugar hvort hæðin sé fyllt í alla staði, ef svo er köllum við í deleteRow og má sjá virknina alla hér til hliðar. Fylkið er í raun tæmt eða núllstillt og svo er farið aftur í upprunalega fallið sem kallar á næsta kubb.

Á skjáskotinu á blaðsíðunni á undan má líka sjá hvernig kubbar eru teiknaðir en eru þeir gerðir viljandi hálfgegnisær til að notanda sjái betur hvernig þeir eru uppstíltir. Þrátt fyrir það er þetta mjög erfiður leikur! Miðað við venjulega tetris er þessi útgáfu talsvert erfiðari og ruglingsleg.

Svo er sjálft boxið teiknað með gl.LINE_STRIP þar sem það er langt þægilegasta viðmótið fyrir það sem við höfðum í huga. Svo eru reitirnir “fylltir” með gl.TRIANGLES. Hægt að er að skipta því út fyrir annað og sjá hvað gerist, leikurinn virkar eins en ef þú myndir teikna boxið sjálft með TRIANGLES myndir fá hvíta reiti sem væru ekki gegnsærir sem dæmi.

Hér er svo hlekkur á sjálfa möppuna:

<https://notendur.hi.is/vthf1/Tolvugrafik/Trisis/>

Hlekkur til að komast beint í leikinn:

<https://notendur.hi.is/vthf1/Tolvugrafik/Trisis/Forrit/tetris.html>

```
function deleteRow(levels) {  
  
    var colTime = 200;  
  
    var x = 0;  
  
    var intervalID = setInterval(function()  
{  
  
        levels.forEach(y =>  
  
            grid[y].forEach( (_, x) =>  
                _.forEach( (_, z) => (grid[y][x][z] = x)))  
  
            );  
  
            if (++x === 7) {  
  
                window.clearInterval(intervalID);  
  
            }  
  
        }, colTime);  
  
        setTimeout(() => {  
  
            for (var i = levels.length - 1; i >=  
0; i--) {  
  
                grid.splice(levels[i], 1);  
  
                grid.push(Array.from(Array(6), _ =>  
Array(6).fill(0)));  
  
            }  
  
            }, 7 * colTime);  
  
        }  
    }  
}
```