

```
1 https://notendur.hi.is/~dia2/203M/Verkefni/Verkefni3/
2 Þessi .pdf skrá er inná og er mögulega þagilegra að lesa heldur en inná gradescop
3
4 Frog-machine
5
6 Tver.html skrár, frog-machine.html og frog-machine-small.html, 700x700 punktar og 500x500 punktar
7
8 Í leiknum leikur þú frosk (bláa kassann) og tilgangur leiksins er að safna stigum frekar en
9 að komast á leiðarenda
10 Á skjánum fyrir ofan leikmyndina er Score, Lives, Flies og Highscore
11 Score byrjar í 4000 og er sífellt að minnka
12 Til að fá fleiri stig er hægt að gera tvennt. 1. Ná flugun. 2. Komast á leiðarenda
13 Hver fluga gefur froskinum 1 aukalíf og 1000 stig, að komast á leiðarenda gefur 3 aukalíf og
14 stigun sem voru inni þegar þú hófst ofan hughscoreð
15
16 Ef froskurinn stekkur fyrir bíl eða ofan vatnið þá missir hann 1 líf og 1000 stig, og þegar
17 lífin/stigin verða að 0 þá endurstíllist leikurinn
18
19 Til að komast yfir götur þá þarf froskurinn að passa sig á að verða ekki fyrir bíl
20 Til að komast yfir ár þá þarf froskurinn að "verða fyrir" trjábolum með því að stökkva uppá þá
21
22 Bílar og trjábolir færast "smoothly" eftir skjánum en froskurinn stekkur eftir reitum
23 sem eru augljóslega aðgreindir
24
25 Flugan stekkur á milli þessara reita og er alltaf nálægt froskinum og alltaf fyrir framan hann
26
27
28 --Samsetning leiks--
29 Leikurinn inniheldur 11 .js skrár
30 car.js
31 fly.js
32 frog-machine.js
33 frog.js
34 group.js
35 lanes.js
36 logs.js
37 tools.js
38 tree.js
39 walls.js
40 world.js
41
42
43 --tools.js--
44 Toolbox sem ég hef smátt og smátt búið til í gegnum námskeiðið
45 Hefur aðferðirnar (sem ég nota í leiknum)
46
47 absMax/absMin sem finna max/min í fylki
48
49 boxVertices/rectangleVertices sem býr til hnútpunkta fyrir kassa/réttthyrning með miðju (0,0,0)
50 boxIndices/rectangleVertices sem býr til indexa fyrir TRIANGLE_FAN litun og stingur staðsetningu
51 indexanna í fylki sem er notað alltaðar til að teikna, en öll object geyma "hvar" þeirra indexar
52 eru í indexafylkinu, þetta er gert til að spara mér gráu hárin að halda utanum hvar allir indexar
53 eru fyrir hvern einasta hlut sem ég by til
54
55 Eftirfarandi lína lýsir sennilega betur hvað ég er að reyna að gera heldur en orð
56 gl.drawElements(gl, TRIANGLE_FAN, indicetrack[this.indicelocation][2], gl.UNSIGNED_BYTE, indicetrack[this.indicelocation][0] );
57
58 en indicetrack er 'global' vigur sem allir komast í en indicelocation er eitthvað sem hver
59 einastí hlutur hefur fyrir sig
60
61
62 --logs.js og groups.js--
63 logs.js og groups.js halda utanum marga trjádrumba/bíla og sjá um að teikna þá og finna
64 handa þeim rétta línu til að vera í. T.d þá er bara smáðað eitt instance af tree og car
65 hlutum en það eru settir 1-5 á hverja línu í þessu 11x100 grídi. Hver einasti logs eða groups
66 hópur heldur utanum fjölda trjábola/bíla og sér um að setja þá niður á "slambú" línu til að
67 það séu t.d ekki alltaf 5 bílar í línu 4, heldur stekkur hópurinn á milli og finnur aðra línu
68
69 Á visu ákvæð ég að taka þessa aðferð út fyrir logs því að það gerir leikinn algjörlega ómögulegan ef það er ekki hægt að segja til um hversu margir trjábolir koma, og ég festi töluna
70 á 3 á hverja línu í vötnunum. Ennfremur þá er samt aðferðin findLane() ennþá til staðar bæði til að koma trjábolum fyrir í upphafi, og líka ef ég vil breyta þessu seinna meir
71
72
73 --frog.js--
74 Hér snúa ég froskinum ásamt stigafjölda hans, flugufjölda etc
75 Í drawFrog() aðferðinni athuga ég í hverri lítrun hvort froskurinn sé:
76
77 Á götunni, en þá athuga ég hvort að sú lína sé upptekinn af bílahöð, ef svo er þá athuga ég í xPos breytu þeirrar línu hvort til sé xPos[i] sem er innan ákveðinnar fjarlægðar frá froskinum, en ef svo er þá fer froskurinn á byrjunarreit, missir 1 líf og 1000 stig
78
79 Á vatni, en þá athuga ég hvort this.float breyta frosksins sé true, en þá hefur froskurinn 'klesst' á trjádrumb (xPos gildin eins og hjá bílunum) og er ofaná honum, ef ekki þá druknar froskurinn og byrjar á upphafsreit
80
81 Athuga einnig hvort froskurinn nái í fluguna, og gef froskinum auka líf og 1000 stig ef svo er
82
83
84 --car.js og tree.js--
85 Búa til hlíðlút og tréhlut og sjá um að teikna
86 Teikningin fer fram á þann hátt eins og lýst var í logs.js og groups.js
87 Hver einastí log og group hlutur sendir inn í tree/car hlutinn upplýsingar um línuna og xPos staðsetningar hvers bíls sem á að teikna, sem og leggja "hraða" hverrar línu við hvert stak xPos vigursins
88
89 Sjá líka um að athuga hvort að bíll/tré sé komið útfyrir og þá endurstíllist allur xPos vigurinn
90
91 --world.js--
92 Sér um að búa til veginn sem froskurinn stekkur um á
93
94 --walls.js--
95 Sér um að búa til svörtu veggina
96
97 --frog-machine.js--
98 Er skráin sem er í öllu, heldur utanum öll object instance og gerir öll webgl köll og inniheldur
99 render() fallið
```