Spurningar

* **WebGL**  
  OpenGL fyrir vefinn. Leið til þess að sýna grafík, 2d eða 3d, í vafranum þar sem skjákortið sér um mesta vinnsluna.
* **GPU**  
  Skjákort, tegund af örgjörva sem er sérhæfður fyrir grafíska vinnslu.
* **Rasterizing**  
  Að teikna upp einhverskonar hlut eftir vektorum. Vinna grafík úr vektorum.
* **GLSL**  
  OpenGL Shading Language, mál svipað C sem er notað til að gera shaders.
* **Vertex**Punktur þar sem tveir eða fleiri vigrar eða línur hittast.
* **Primative**  
  Primitive type: Einföld breyta eins og til dæmis int, bool, char, double. Ekki objectar.  
  Geometric primitive: Línur, kassar, kúlur, polygon mesh. Einföldustu hlutirnir sem er hægt að gera með grafík.
* **Fragment**Gögnin sem þarf til að teikna einn pixla af einhverjum hlut. Ekki það sama og pixel því það á eftir að skoða hvort þetta fragment sé sýnilegt á skjánum.
* **Pixel**Einn punktur á skjánum. Minnsta einingin á skjánum.
* **Clip space**Hlutinn af þrívíddarumhverfinu sem er sýnilegur á skjánum.
* **View frustum**  
  Formið á clip svæðinu. Það er í laginu eins og pýramídi sem búið er að skera toppinn af.
* **Z-buffering**Gömul aðferð til þess að teikna hlutinn á skjáinn á þá vegu að það sem er næst myndavélinni sé sýnilegt og hylji það sem er fjær myndavélinni (en ekki öfugt) er að nota hinn svokallaða „painter´s algorithm“. Þessi algórithmi sorterar öll fragment eftir því hversu langt þau eru frá myndavélinni, og teiknar þau síðan öll í röð, fjærst fyrst og færir sig svo nær. Gallinn er samt sá að þá ertu að teikna yfir fullt af pixlum sem þú ert búinn að teikna núþegar. Til að spara vinnslu er villtu bara teikna það sem hægt er að sjá. Ef það er eitthvað annað á bakvið hlutinn, þá er okkur alveg sama. Þá þurfum við líka ekkert að flokka fragmentin í réttri röð. Það er Z-buffering.
* **Right-hand coordinate system (RHS)**Regla sem er notuð til að sjá hvort búið sé að spegla þrívítt rúm. Ef þú getur stillt up þumalputtanum sem Z-ásinn, vísifingri sem Y-ásinn og löngutöng sem X-ásinn, allt með hægri hendi, þá er rúmið ekki speglað. Ef það virkar bara að gera þetta með vinstri hendinni, þá er búið að spegla það.

*Afhverju eru 3D objectar búnir til útfrá samsettum þríhyrningum í 3D grafík?*

Það er hægt að setja saman öll möguleg form ef notaðir eru nægilega margir, nægilega litlir þríhyrningar. Til að spara vinnslu eru þeir ekki hafðir og margir, og líta þá bara svona nokkurnveginn út eins og það sem formið er að líkja eftir.

*Útskýrðu ítarlegaog tæknilega (en án kóða) með eigin orðum ásamt skýringamyndum hvernig rendering pipeline virkar í WebGL.*

Það er hægt að lýsa þessu svona í stuttu máli:

****Set up**: Öll nauðsynleg gögn s.s. geometry, normals sótt   
**Vertex shader**: Breytir 3d vertex gögnum í 2d gögn fyrir skjáinn, ásamt dýpt.  
 Til að gera þetta eru 3d gögnin látin gegnum ýmis matrix sem færa allt á réttan  
 stað, með rétt rotation, og rétta dýpt.   
**Rasterization**: Tekur gögnin, sem eru enn í formum, og breytir þeim í fragments  
**Fragment shader**: Gefur hverju fragment lit, þau fragment sem eru nær skyggja  
 þau sem eru fjær. Ef eitthvað er gegnsætt er litum blandað saman.  
**Rendering**: Breytir þessum fragmentum beint í pixla á skjánum.

*Transform (translation,rotation og scale). Komdu með sýnidæmi með vector/vigri)í cartesian hnitakerfi (x,y,z) fyrir:*

**Translation, rotation & scale**

*Fylki (e. matrix). Sýndu með sýnidæmi hvernig 4x4 fylki vigri/vector er reiknaður með:*

**Translation matrix, Rotation matrix, Scale matrix**

*(Sjá myndir)*



