Kratki uvod u programiranje grafike

Gamedev radionica

prigovaranje zbog gramatickih pogresaka strogo zabranjeno :)

Overview

- Glavna petlja
- 2D/3D I hardverska akceleracija
- Osnove 2D grafike
- Osnove 3D grafike

Glavna petlja

- Igre se vrte u beskonacnoj petlji
- U svakom prolazu petlje:

  1) **priprema se slika I salje na ekran**

  2) osvjezavaju se elementi igre (glavni lik, neprijatelji, kamera...)

  3) provjerava se stanje tipkovnice/misha/...

  4) procesiraju se zvukovi/glazba

  5) obradjuju se mrezne poruke

Glavna petlja

KRIVO: Osvjezavanje elemenata za fiksan faktor (npr. 1.5 pixela po frame-u)

ISPRAVNO: izracunati koliko je vremena proslo od zadnjeg osvjezavanje (npr. 0.09 sekundi)

ALTERNATIVA: koristiti prvu metodu ali nekako ograniciti broj frame-ova po sekundi (primjer: GTA 2)

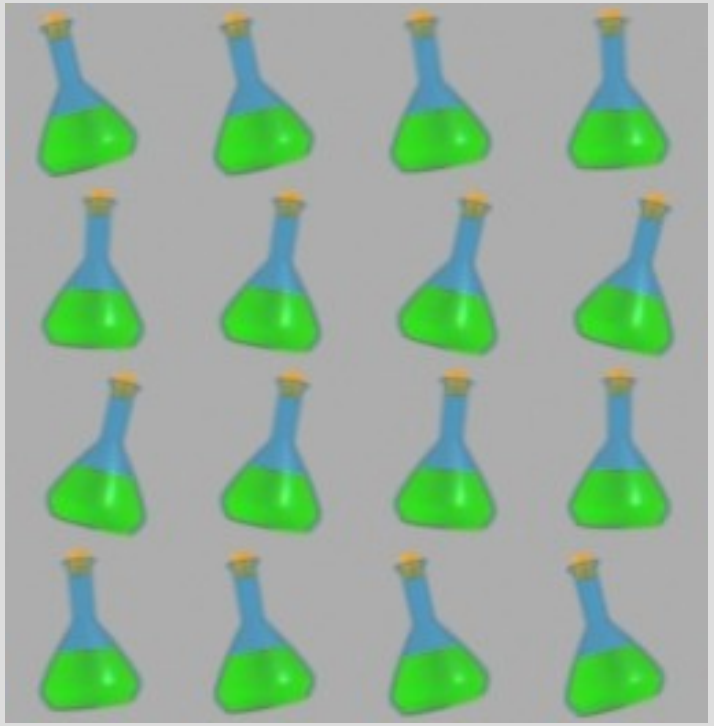
2D/3D u suvremenim igrama

- **2D**
- Koriste se segmenti 2D slika, tocke I linije
- Slikovni elementi se nalaze u 2D bufferima

- **3D**
- Koriste se poligoni, linije I tocke
- Slikovni elementi se nalaze u teksturama
- Dimenzije tekstura moraju biti potencije broja 2 (16,32,64...)
- Moguca simulacija 2D-a sa ortogonalnom projekcijom

Klasicno 2D programiranje

- Svi elementi su 2D slike
- Frame (backbuffer) se brise svaki frame I na njega se kopiraju razni segmenti slika
- Na kraju se backbuffer kopira na
- ekran ili u slucaju dvostrukog
- Buffera, zamijene se pokazivaci
-
- **Color key:** jedna boja slike je
- prozirna



2D sad I nekad

Nekad:

- U doba DOS-a: VGA, interrupt #13
- Windows API (BitBlt) i unix X11 naredbe
- DirectDraw – hardverski akceleriran 2D

Danas:

- Emulacija 2D-a 3D API-em

3D

- Prikaz se postize poligonima
- Poligon se zadaje pomocu tri tocke
- Svaka tocka moze u sebi, osim pozicije, sadrzavati razne informacije:
  * texsturne koordinate
  * normalu
  * boju
  * ...

3D

- Tocke se transformiraju pomocu matrica
- 2 glavne matrice: **world** i **projection**
- Projekcijskom matricom moze se postici:
  * 3D perspektiva
  * izometricna projekcija
  * 2D ortogonalna projekcija
  * razne nebuloze :)

3D

- Za pravilan prikaz u perspektivi, poligoni se moraju sortirati ili se koristi **z-buffer**

Z-Buffer:

- 16/32bitni buffer dimenzija ekrana
- Svaki pixel poligona zapise se u color buffer I u z-buffer