

De productie van videogames voor en binnen het onderwijs

Wouter Baars

wouter@wouterbaars.net

Wat komt er bij kijken bij het maken van een (educatieve) videogame?

- Techniek
- Organisatie en samenwerking
- Tijd en kosten
- Valkuilen

Techniek

- Imovie, Iphoto, Imac, Iamsterdam, Ipod, Ilearn, ...

I lusion.....

Techniek

'Professionele' VideoGames: Doom III, Unreal Tournament, Heroes of Might and Magic, Simms,

- > 2 jaar ontwikkeltijd
- > 30 man productieteam
- > 1-2 miljoen Euro budget

Productieproces van een high-end videogame



Werken in de gameindustrie?



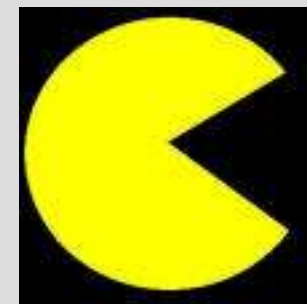
Bewerkingen - Gereedschappen

Programmeren: C / C++

Designers:

- 3D (Maya, 3ds): modellen, levels
- 2D (Photoshop, Illustrator,...) textures, animaties, figuren, knoppen, achtergronden,...
- Audiodesign (Protools, Cubase, Logic)

Igame?



Voorbeeld: 3D gamemaker



Middleware

Algemene Middleware: Macromedia Director, Flash

GameKits: Darkbasic, Torque, Gamemaker, Blender, Quest, Gamestudio,...

zie: www.conitec.net/a4faq.htm

Scripttaal versus “C”

```
typedef struct { double x,y,z;} XYZ;
```

```
/*
```

```
Rotate a point p by angle theta around an arbitrary axis r
```

```
Return the rotated point.
```

```
Positive angles are anticlockwise looking down the axis
```

```
towards the origin.
```

```
Assume right hand coordinate system.
```

```
*/
```

```
XYZ ArbitraryRotate(XYZ p,double theta,XYZ r)
```

```
{
```

```
XYZ q = {0.0,0.0,0.0};
```

```
double costheta,sintheta;
```

```
Normalise(&r);
```

```
costheta = cos(theta);
```

```
sintheta = sin(theta);
```

Scripttaal versus “C”

```
q.x += (costheta + (1 - costheta) * r.x * r.x) * p.x;  
q.x += ((1 - costheta) * r.x * r.y - r.z * sintheta) * p.y;  
q.x += ((1 - costheta) * r.x * r.z + r.y * sintheta) * p.z;  
  
q.y += ((1 - costheta) * r.x * r.y + r.z * sintheta) * p.x;  
q.y += (costheta + (1 - costheta) * r.y * r.y) * p.y;  
q.y += ((1 - costheta) * r.y * r.z - r.x * sintheta) * p.z;  
  
q.z += ((1 - costheta) * r.x * r.z - r.y * sintheta) * p.x;  
q.z += ((1 - costheta) * r.y * r.z + r.x * sintheta) * p.y;  
q.z += (costheta + (1 - costheta) * r.z * r.z) * p.z;  
  
return(q);  
}
```

Scripttaal versus “C”

```
XYZ ArbitraryRotate2(XYZ p,double theta,XYZ p1,XYZ p2)
```

```
{
```

```
    XYZ q = {0.0,0.0,0.0};
```

```
    double costheta,sintheta;
```

```
    XYZ r;
```

```
    r.x = p2.x - p1.x;
```

```
    r.y = p2.y - p1.y;
```

```
    r.z = p2.z - p1.z;
```

```
    p.x -= p1.x;
```

```
    p.y -= p1.y;
```

```
    p.z -= p1.z;
```

```
    Normalise(&r);
```

```
    costheta = cos(theta);
```

```
    sintheta = sin(theta);
```

Scripttaal versus “C”

```
q.x += (costheta + (1 - costheta) * r.x * r.x) * p.x;
```

```
q.x += ((1 - costheta) * r.x * r.y - r.z * sintheta) * p.y;
```

```
q.x += ((1 - costheta) * r.x * r.z + r.y * sintheta) * p.z;
```

```
q.y += ((1 - costheta) * r.x * r.y + r.z * sintheta) * p.x;
```

```
q.y += (costheta + (1 - costheta) * r.y * r.y) * p.y;
```

```
q.y += ((1 - costheta) * r.y * r.z - r.x * sintheta) * p.z;
```

```
q.z += ((1 - costheta) * r.x * r.z - r.y * sintheta) * p.x;
```

```
q.z += ((1 - costheta) * r.y * r.z + r.x * sintheta) * p.y;
```

```
q.z += (costheta + (1 - costheta) * r.z * r.z) * p.z;
```

```
q.x += p1.x;
```

```
q.y += p1.y;
```

```
q.z += p1.z;
```

```
return(q);
```

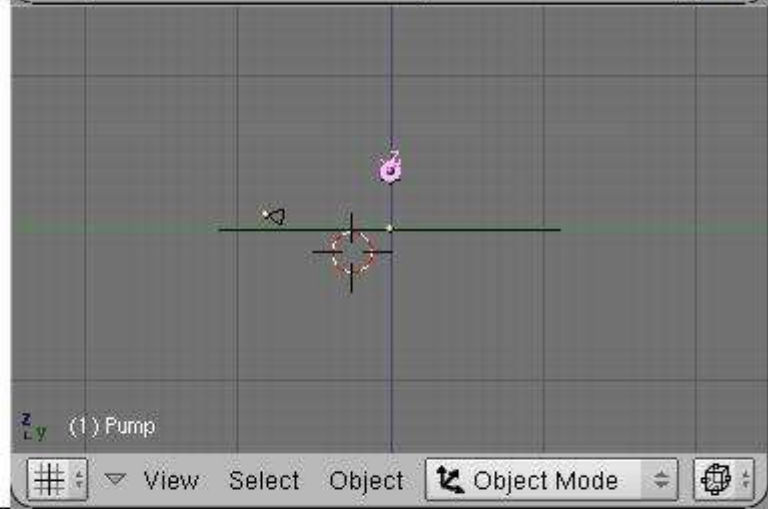
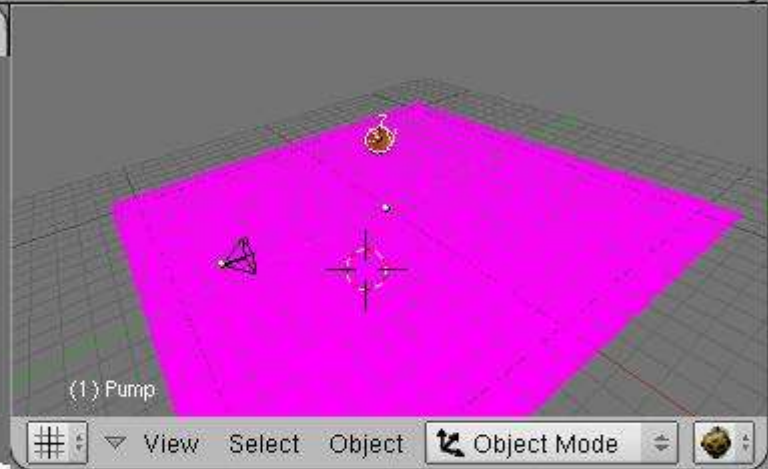
```
}
```

Scripttaal versus “C”

In scripttaal:

Rotate (“kubus”, *axis*, *degrees*)

blend



Blender Logic Editor interface showing a logic brick for the 'Pump' object.

Actor	Ghost	Dynamic	Rigid Body
Do Fh	Rot F	Mass: 1.00 ize: 0.620 Form: 1.04	
Damp: 0.004	RotDamp: 0.900	Anisotropic	

Bounds

ADD property

Sensors: Pump (Add)

- Keyboard naarvoren
- Keyboard sensor

Controllers: Pump (Add)

- AND pass
- AND pass1

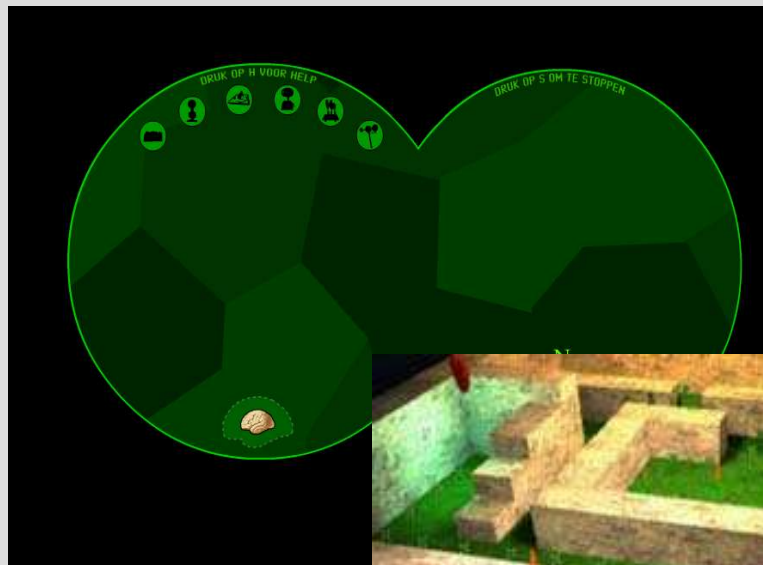
Actuators: Pump (Add)

- Motion naarvoren1
- Motion naar achteren

Middleware - techniek

- Snellere productie
- Veel standaard 'bouwstenen', maar toch redelijk open structuur
- Scripttaal ipv “C”
- Grenzen aan de mogelijkheden

Resultaten



Vaardigheden - techniek

De middleware maakt voornamelijk het programmeren makkelijker.

3d en 2d grafische pakketten en audio creatie zijn nog nodig.

Gamekits worden vaak wel geleverd met uitgebreide modellen en geluidsdata bases.

Techniek - Leertijd

Gamekit (Darkbasic, Torque,...)

Scripttaal (Lingo, actionscript, python,...)

2d grafische software (Photoshop, Fireworks,...)

3d design software (maya, discrete, blender,...)

evt. audiobewerkingssoftware

Reken op minimaal 100 uur per element leertijd voor de basis, > 300 uur voor goede beheersing.

Keuze voor middleware in de klas

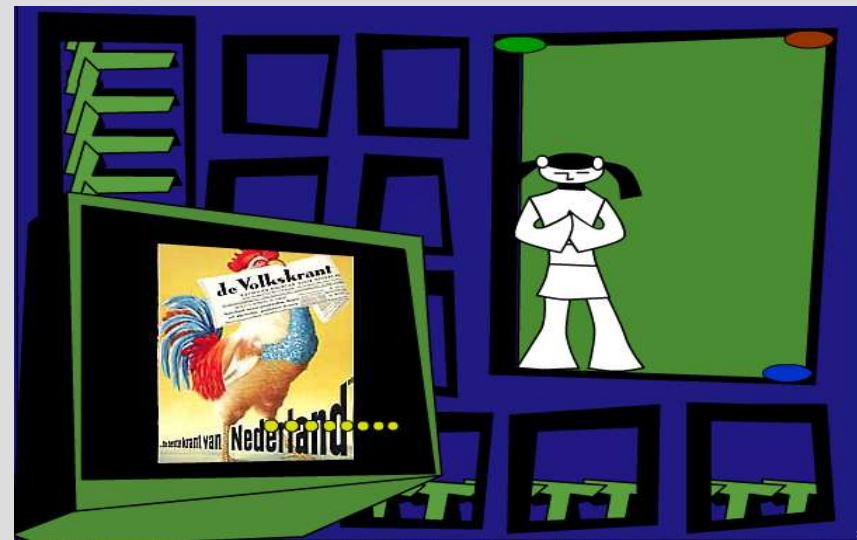
- Degelijkheid van de software
- Moeilijkheidsgraad tov gewenst studieniveau
- Sluit scripttaal aan bij overig onderwijs en/of de arbeidsmarkt vraag
- Toekomstperspectief van leverancier
- Mogelijkheden (2d/3d/webgames/non games)
- Open source/community support

Wat kost dat?



Replica van een hit, weinig nieuw design

Bedenk een concept en
gameomgeving 2d



Wat kost dat?

- Meer levels
- Beter graphics
- Beter AI
- 3d
- Multiplayer
- Mobile game
- Streaming video.....
- Enz. Enz. Enz.

Wat is leuk?

De **spelregels** van een VideoGame bepalen of die game leuk is.

Niet: de vormgeving

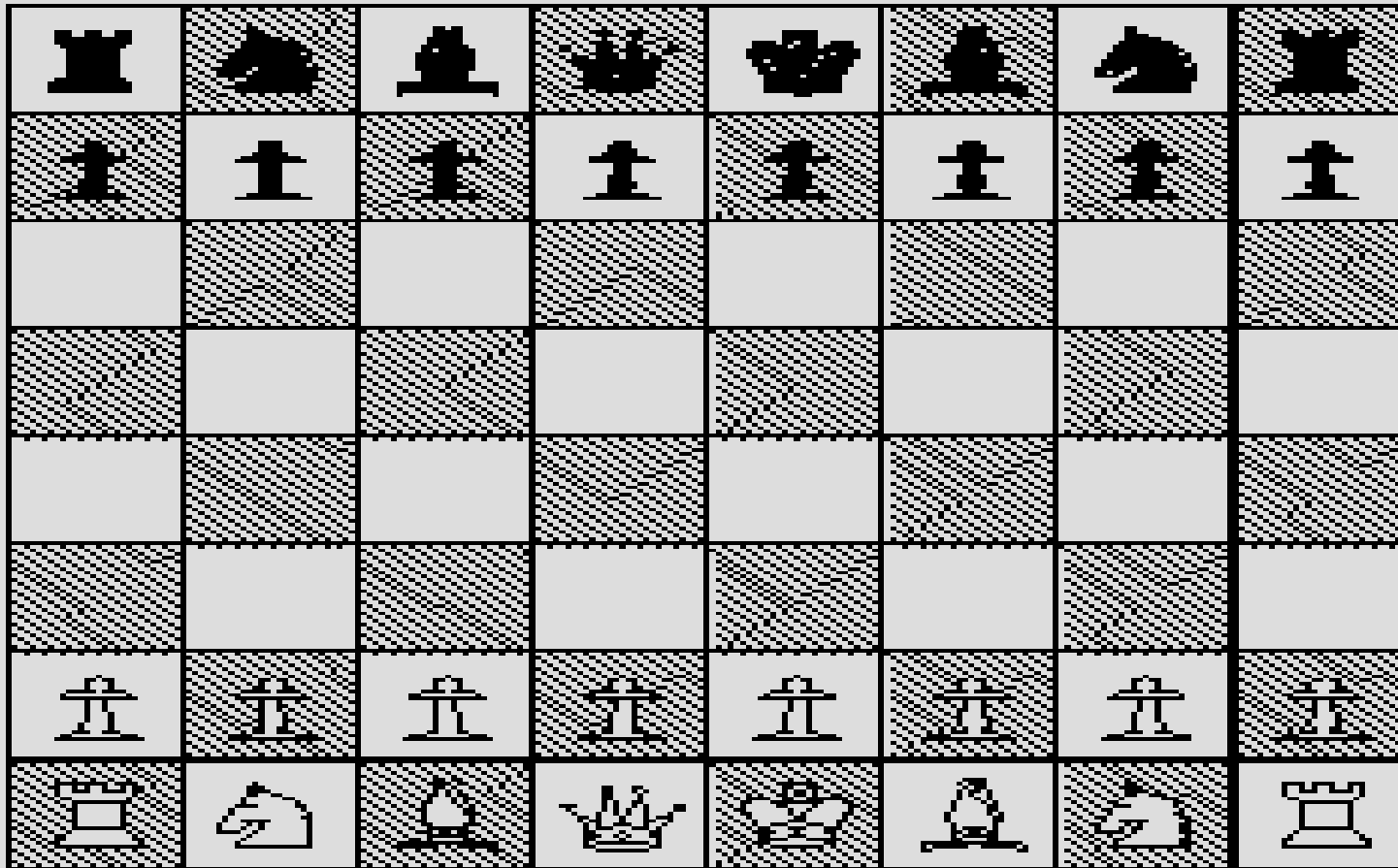
Niet: 2d of 3d

Niet: de verhaalcontext.

Leuk spel



Ook een leuk spel...



lets minder leuk...?

1. e4 e5 2. Nf3 Nc6 3. Bb5 a6 4. Ba4 Nf6 5.
O-O Be7 6. Re1 b5 7. Bb3 O-O 8. h3 Bb7 9.
d3 d6 10. a3 Na5 11. Ba2 c5 12. Nbd2 Nc6
13. c3 Qd7 14. Nf1 d5 15. Bg5 dxe4 16. dxe4
c4 17. Ne3 Rfd8 18. Nf5 Qe6 19. Qe2 Bf8
20. Bb1 h6 1/2-1/2

Samenwerking

Hoe groot acht je de kans dat een professor een pop hit schrijft?

Projectmanagement

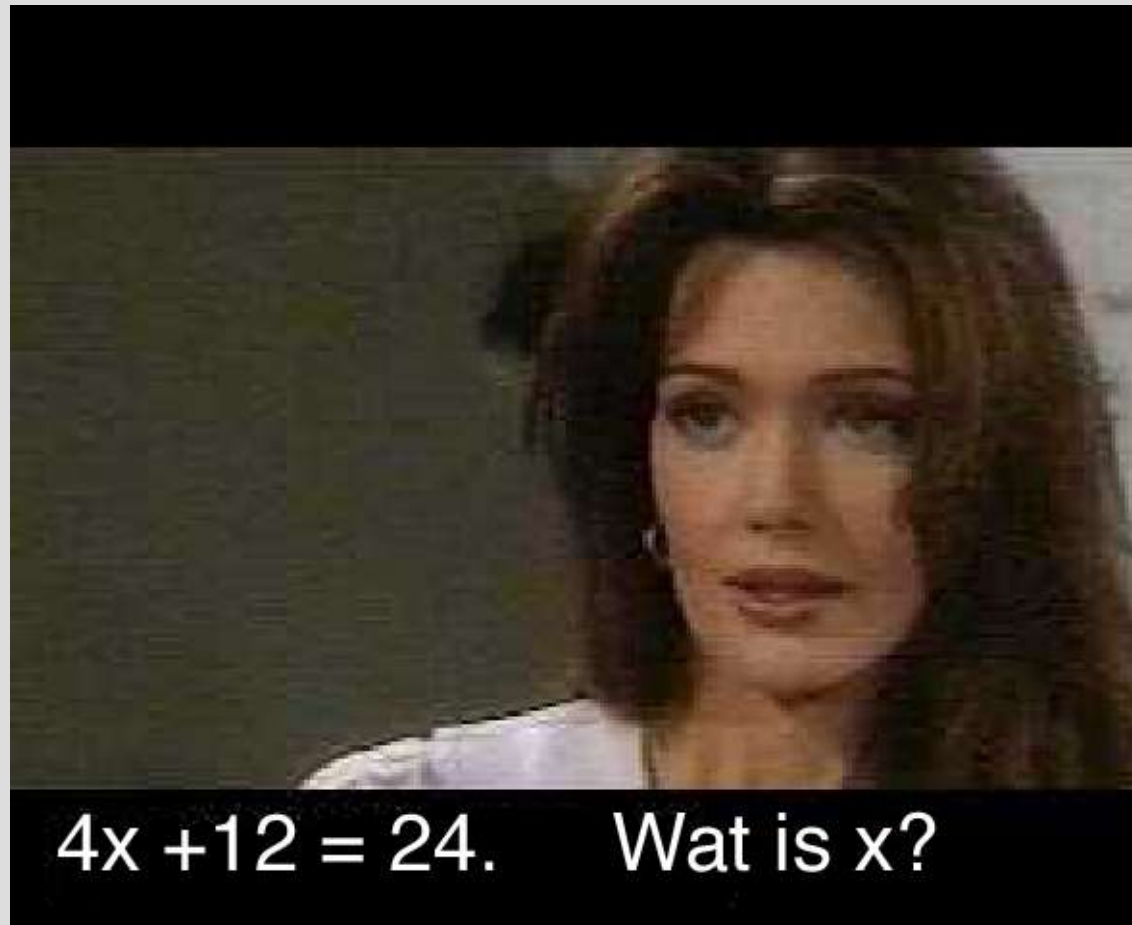
1. Idee
2. Spel inhoud ontwerp (functioneel ontwerp)
3. Productie: Design en Programmeren
4. Testen
5. Nazorg

Subsidie voor Games?

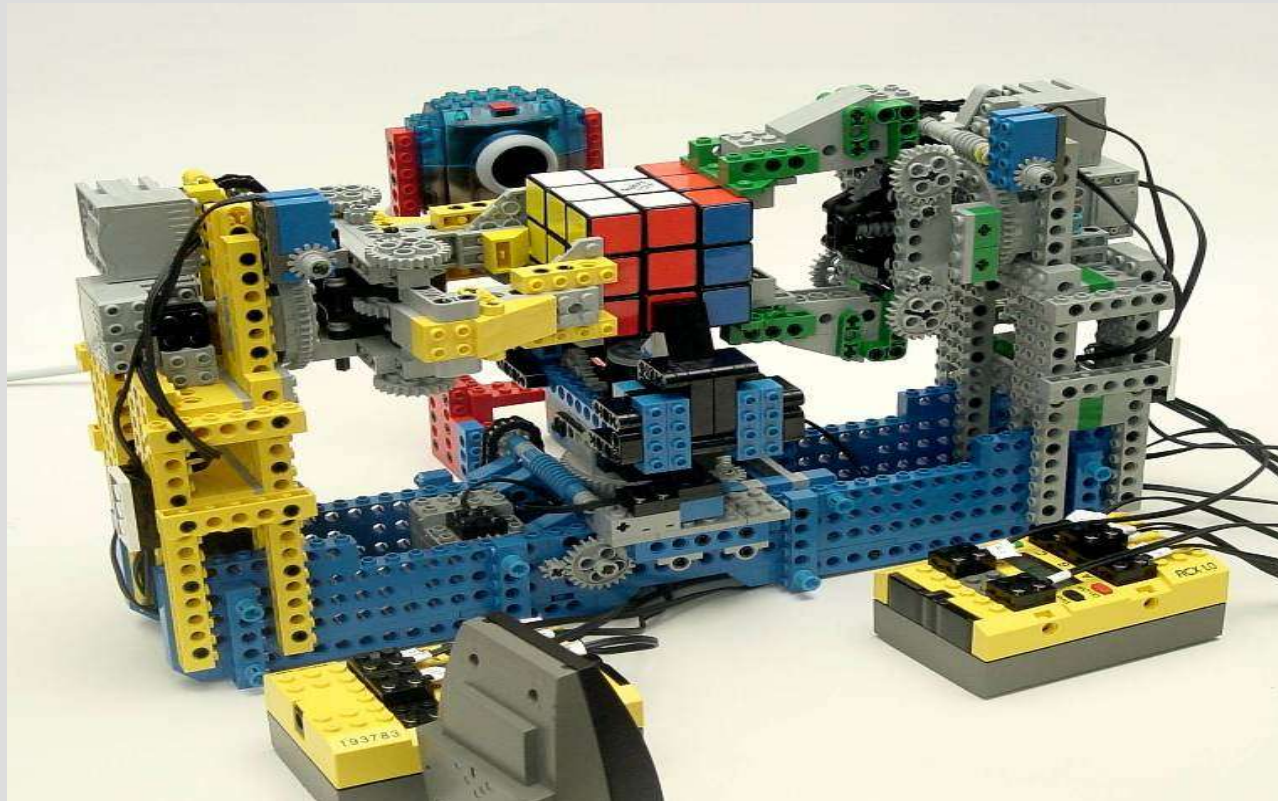
De Creatieve Industrie is de nieuwe motor van de economie



Samenwerking – Educatieve content



Eerst het spel dan de middelen



Conclusies

- De spelregels zijn het belangrijkste, niet de techniek/vormgeving/3d
- Middleware versimpelt het maken van een game, maar het blijft een vak
- Het begrijpen van het productieproces vergroot de kansen op succes
- Een educatieve game maken is een kunst
- Er is altijd een business model, ook als het een niet commerciële game betreft

33