

# Tehnologija iger in navidezna resničnost / Razvoj iger (skozi uporabo XNI na iOS platformi)

Peter Peer ++

## Uvodno predavanje (motivacijsko, pregledno)

- kaj je to igra
- igričarska industrija, neodvisni razvijalci
- žanri, kategorije
- potek izdelave, sestavni deli igre, področja
- arhitektura igre
  - glavna zanka (game loop)
  - knjižnice, frameworki
- dokumentacija
- oblikovanje igralnosti, elementi igralnosti
- formalne zahteve predmeta, sprotno preverjanje

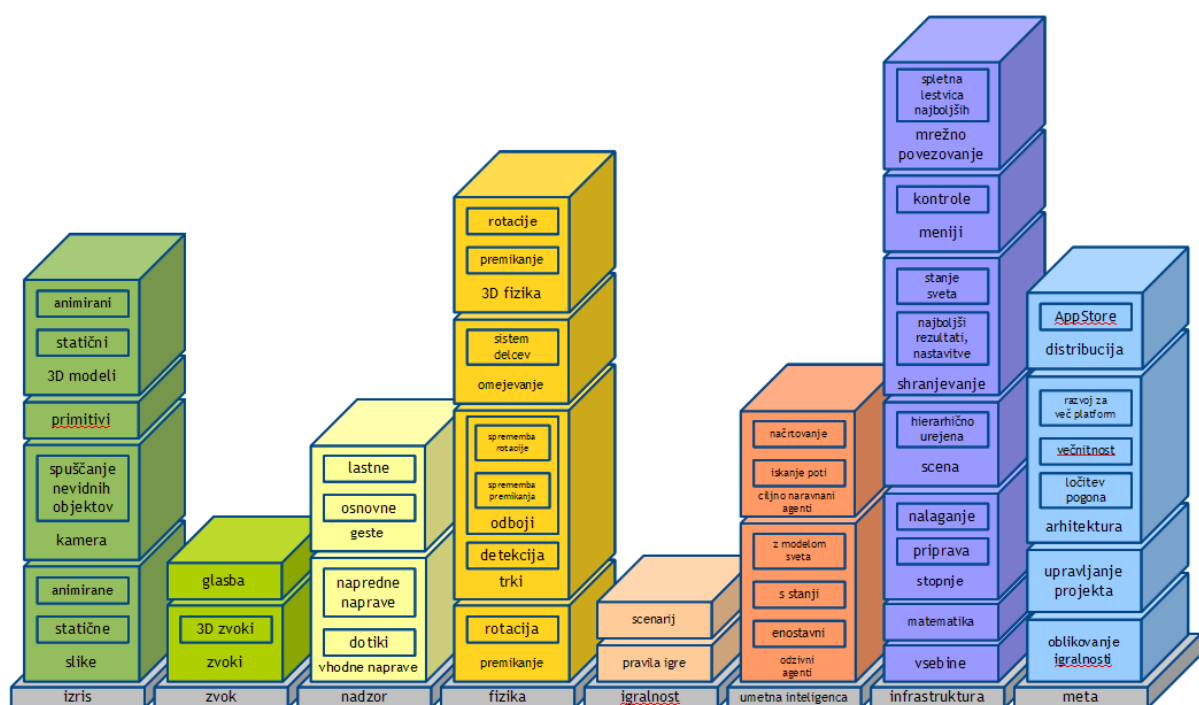
Osnovne informacije, TINR.pptx ...???

UPO je postavljajal temelje, opismenjeval, TINR bo naredil piko na i. Upam, da ste se vmes kaj naučili.  
;-)

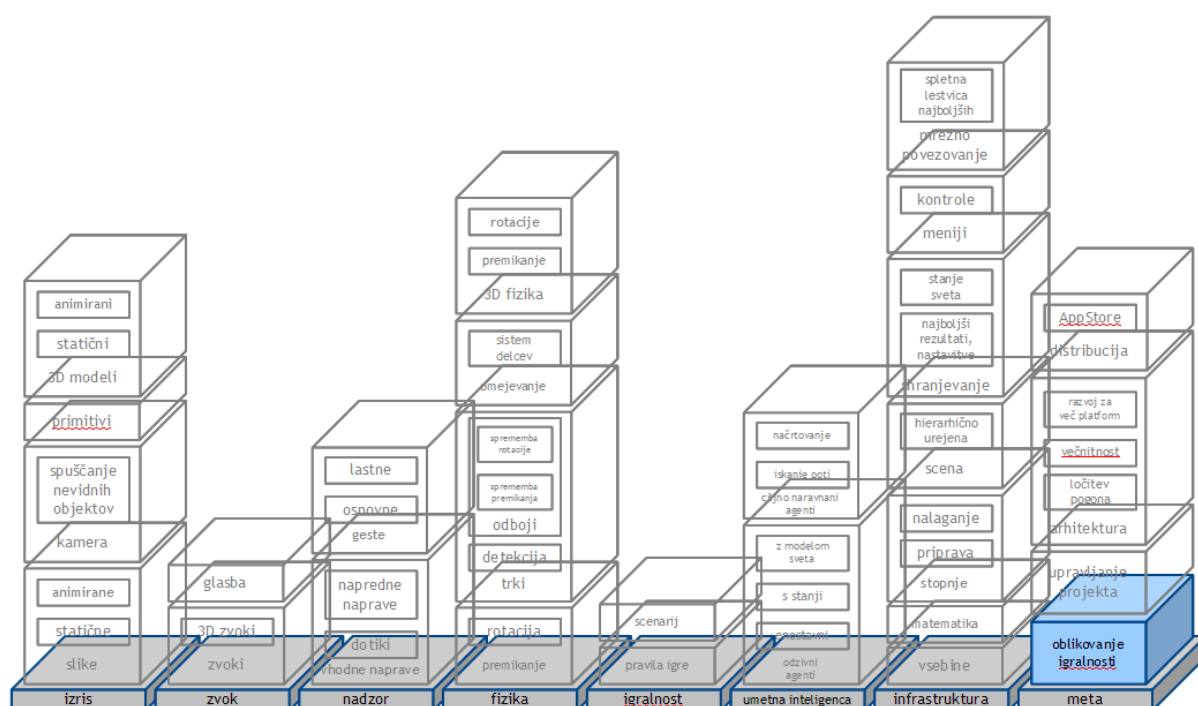
Zakaj ste si izbrali ta predmet?

Pravila predmeta...??? Biblija! + Koncept.pptx ...???

Kateri so sestavni deli igre in kako jih bomo spoznavali?



Aktualne škatlice: + arhitektura, dokumentacija ...???



## 1. Kaj je igra?

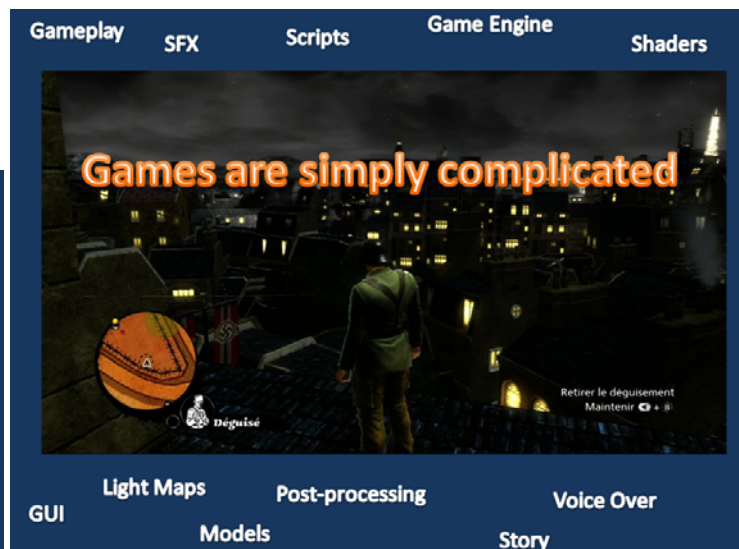
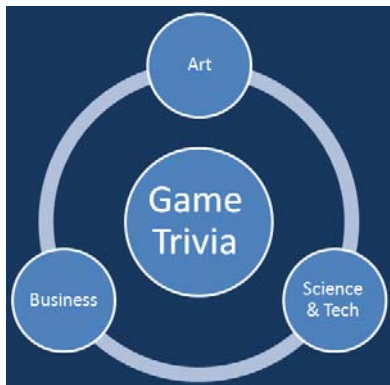
Kakšna je razlika med programsko opremo, igro in filmom? Igra je interaktivni medij: interaktivnost jo razlikuje od filmov in drugih oblik umetnosti; vsebina jo razlikuje od drugih tipov programske opreme; in seveda so veliko bolj zabavne!

Igra = navidezna resničnost + igralnost!

Tri osnovne sestavine igre: ideja, logika, vsebina. Ideja je (predvsem) domena oblikovalcev igralnosti (game designer), ki znajo dobro ovrednostiti vpliv igre na igralca. Logika je vrsta pravil in postopkov, ki idejo osvetlijo v detajle in so osnova za implementacijo (programiranje; namenski programi, kjer skozi vmesnik naklikamo vso potrebno mehaniko; skriptanje). Vsebina povečini zajema 2D slike, 3D modele, glasbo in podatke o stopnjah (grafično oblikovanje; avtomatsko generiranje (retro); prilagajanje/modanje).

Recimo, da je splošno znanje iz srednje šole dovolj dobra osnova za začetek, vendar z resnim projektom (igra:igrica, aplikacija: app) pride zahteva po spopadanju z vedno novimi tehnologijami, ki ti vzame kar  $\frac{1}{4}$  časa, univerzitetno linearno algebro ter integrali iz pdf-jev vodilnih glavnih industrije. Za tiste z željo po izzivu pa je to ravno Meka: tovrstno premagovanje ovir ti predstavlja vsaj toliko zabave kot sprotno igranje nastajajoče igre, nastale kreacije.

Igre imajo vse:



### Namenski programi:

Večino dela opraviš skozi delo z grafičnim urejevalnikom stopenj, likov, dogodkov, zvokov ipd., programiranju se lahko povsem izogneš. Orodja so lahko ali splošna (Game factory 2, Game maker 7) ali pa specifična za določen žanr (FPS creator, Adventure game studio (pokaži in klikni avanture), RPG maker, RPG toolkit).

Lahko se naredi marsikaj, ne nasedajte pa obljubam, da lahko s parimi kliki naredite prvoosebno streljačino (FPS Creator). Tudi z nekaj ducat ne bo ravno šlo. Ah, lažem, saj jo lahko, vendar stvar bo okleščena za igrabilnost, ki pa je v igri ena izmed bistvenih sestavin. Torej, pravzaprav ne lažem! ;-)

Na poti do čistega programiranja ne moremo mimo Torque Game Builderja. Temelji na pogonu Torque 2D, zraven pa dobiš tudi urejevalnik stopenj s par grafičnimi orodji (recimo za nastavitev poligona za detekcijo trkov), potem pa le še v skriptnem jeziku TorqueScript opišeš pravila igre (ni potrebe po inicializaciji recimo DirectX in spuščanju v matematiko trkov).

### Prilagajanje/modanje:

Končane igre so najboljša izhodiščna točka za modeliranja željne oblikovalce, igralce. Ne moreš si ravno izmišljevati, kakšna bo igralnost, imaš pa možnost videti svoje 3D modele in izvirno oblikovane stopnje v končnem izdelku. Nemalekatera igra ima danes priložen uraden urejevalnik za prilagajanje (Half life 2, Unreal tournament, Battlefield, Far cry (FPS); The elder scrolls, World of Warcraft (RPG); Warcraft, Command & conquer, Supreme commander, Age of empires, Defcon, Civilization (strategije); Grand theft auto, The sims, SimCity, The movies).

### Struktura igre:

Igre imajo v primerjavi z ostalo programsko opremo drugačno, specifično strukturo:

- Začetek/inicializacija (start up)
- Predstavitveni filmček/animacija/slika
- Pročelje (front end)
  - Opcije igre, videa, zvoka
- Nalagalni zaslon

## Igra

Uvod

Igranje (gameplay)

Možnost prekinitve, zaustavitve

Odjavni filmček/animacija/slika

Avtorstvo

Zaključek (shut down)

Glavna zanka:

```
// a simple game loop in C++
int main()
{
    Initialize();
    while (true)
    {
        Process();
        Update();
        Render();
    }
    ShutDown();
}
```

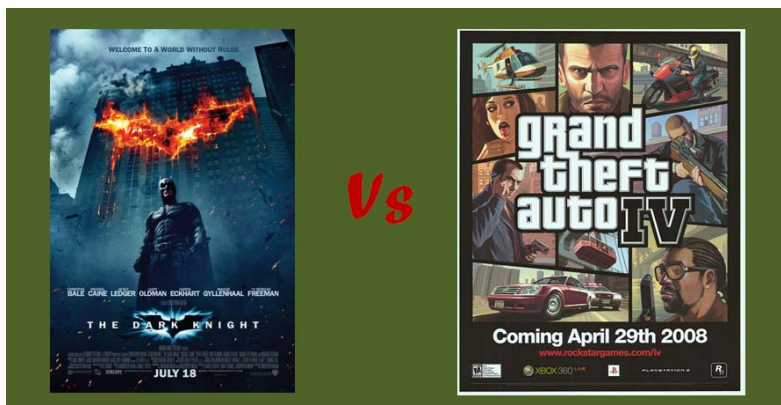
Kaj se dejansko zgodi?

Procesiraj vhod iz kontrolnih enot, na primer, če igralec pritisne presledek, potem mora karakter v igri skočiti.

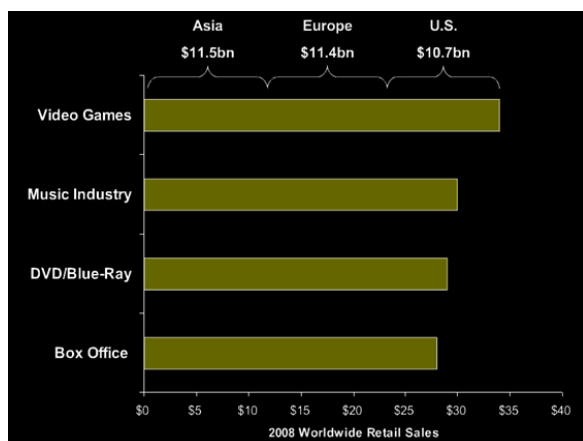
Posodobi entitete: v tem delu igra reagira na spremembe igralca ter ustrezno spremeni stanje igre.

Posodobi grafičen prikaz: izriši ozadje, slike (sprites), posodobi HUD (head-up displays) elemente,...

## 2. Zakaj je industrija iger zanimiva?



1. Jasno, zaradi zabave, tako z vidika uporabnika kot ustvarjalca! ;-)
2. film proti igri  
kontekst primerjave: vidik prodaje/prihodka v prvem tednu  
239 milionov \$ proti 500 milionov \$  
(Vir: Video Games Industry Trends by Ferrara)
3. Velikost trga



Problemi? Pripeljati stvar(i) do konca! Najti založnika! Uspeti!

### Indie scena:

1. Neodvisna scena, očitno! ;-)
2. Ni založnika, mogoče se pojavi kasneje.
3. Fond (budget)? Kaj je to? ;-) Ta je majhen, če sploh obstaja.
4. Individualisti in majhne ekipe.
5. Poenostavljeno rečeno: nasprotje načinu dela velikih igralcev na trgu.
6. Si sam svoj šef!
7. Na preži za inovativnostjo, raziskovanje novih vidikov snovanja (design), pripovedovanja zgodbe in celo umetnostnega izražanja.

Kaj bomo mi, Mainstream ali Indie? Vsaj na začetku ... (Potem bomo po vzoru EA Games prerasli v FRI Games! ;-)

### Zakaj Indie?

- Časovno zelo zahteven proces.
- Garancije za finančno povračilo ni!
- Za vse si praktično sam, posledično se ti lahko malo obrne. ;-)

### Sami minusi, kaj pa plusi?

- + Delal boš nekaj, kar si resnično želiš!
- + Ni šefa, ki bi ti skakal po glavi – ti si šef!
- + Če si dovolj kreativen boš slej ali prej naredil hit, kot je recimo tisti Japonček Pixel naredil Cave Story. In potem si več kot poplačan ... Vztrajnost je vrlina, mar ne?! ;-)
- + TINR: odličen pristop za učenje, spoznavanje celote, the big picture!

Razvoj od vsakega posameznika pričakuje samomotivacijo, bolje rečeno zvrhan koš le-te, in pošteno zaščito pred obupavanjem. Preden imaš kaj za pokazati, preteče namreč (prekleto) veliko časa. Ena izmed dobrih zaščit je ta, da načrt projekta dobro zastaviš na samem začetku in ga potem minimalno spreminjaš tekom razvoja, vse ostale ideje pa daješ na stran za različico 2.0. Tako imaš preverjeno več možnosti, da bo različica 1.0 dejansko tudi ugledala luč sveta.

Kako velik zalogaj je igra iz trgovine za 40€? 20 do 100 ljudi dela 2 do 3 leta! Ljudje z izkušnjami, delavnikom, ki je daljši od 40 ur na teden. In temu primeren je tudi proračun!

Zakaj je segment programiranja kljub vsemu tako pomemben? Da te skupnost sploh začne jemati resno, moraš najprej znati programirati; da ti ne zabrusijo: »Ja, najprej se postavi prt in krožnike, potem greš pa šele mizo delat.« (Vrvy) Torej, 3D modelov ti ne bo nihče naredil, če ne boš demonstriral, da jih znaš uporabiti v pogonu, torej moraš prej imeti in obvladati pogon.

### 3. Proces razvoja igre

Kje se razvija?



Zelo kratka zgodovina:



Sam proces:



(Katera je ta igra? ;-)

To so torej glavne štiri komponente. Načeloma nobena ni manj pomembna, je pa seveda modra tista, ki deluje kot lepilo. Ta vidik nas bo tudi najbolj zanimal.

Snovanje/oblikovanje igralnosti (gameplay design)

Žanri

Dokumenti

2D/3D svet (art)

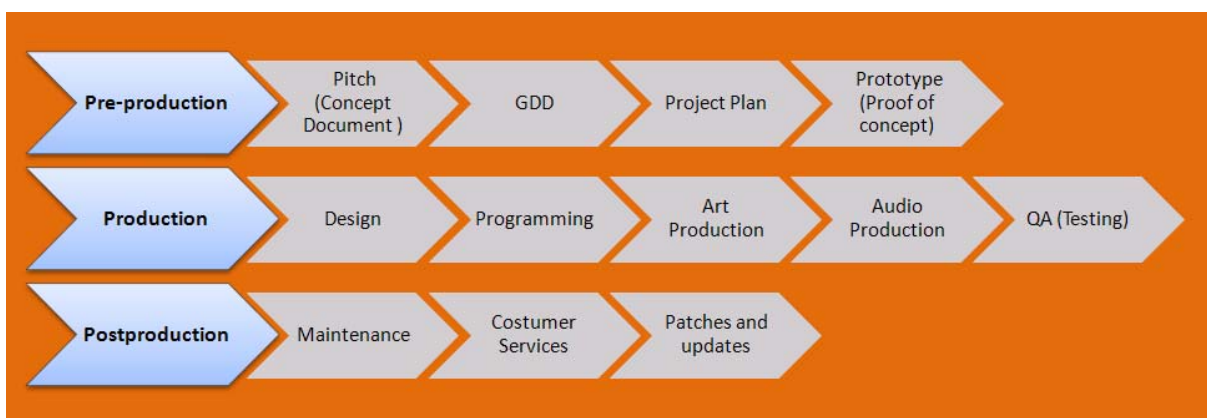
Programiranje

Pogoni (engines)

Poddiscipline

Zvok

Proces detalneje:



(GDD – Game Design Document)

(ProofOfConcept spada bolj v fazo produkcije, saj največkrat pomeni, da smo prišli do prvega

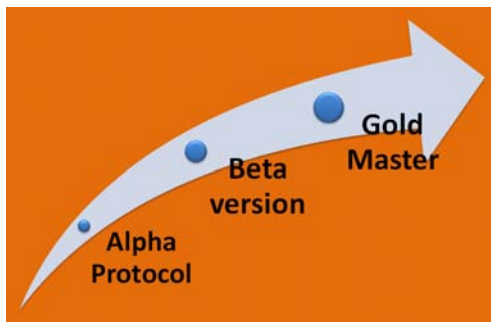


mejnika, kjer lahko igro igramo (first playable milestone), brez dejanskega programiranja pa je to seveda težko.)

Spredaj lahko dodamo Concept Phase, zadaj pa Aftermarket Phase.

Detaljno zaporedje je diskutabilno, saj je vezano na metodo razvoja programske opreme. Prototipiranje je le ena izmed petih osnovnih metod (kar smo se naučili pri predmetu Vodenje projektov). Primarni cilj te metode je potrditi koncept, kar v splošnem z vidika vodenja projekta pomeni, da dobimo potrditev naročnika, da je to dejansko to kar potrebuje. Tako prototipiranje služi tudi za zajemanje in posledično potrditev naročnikovih zahtev. Potem pa imamo pri prototipiranju kretnico: ali začeti sedaj potrjen koncept graditi iz nič skozi načrt (do sedaj tega nismo imeli, saj smo dejansko naročniku pomagali definirati zahteve) ali pa graditi končni izdelek kar iz prototipa. V prvem primeru bosta sedaj nastali funkcionalna in tehnična specifikacija (o vsebini teh dveh dokumentov bomo govorili malce kasneje) kot odgovora na fazi projekta analiza in načrtovanje. V drugem primeru pa je priporočljivo, da ti specifikaciji v grobem nastaneta že prej (da nimamo hišice iz kart, ki je ne bomo mogli dobro nadzirati, da je osnovna zasnova, arhitektura dobra).

Trije glavni mejniki:



Tri glavne kategorije?!



**Snovanje/oblikovanje igralnosti (gameplay design):**

- Definiraš vsebino in pravila igre.
- Opišeš izgled igre in interakcijo.
- Narediš detajlni načrt dela.

Snovanje je kritičen korak, zahteva izkušnje in znanje (kaj pa ne? ;-).



Elementi igralnosti:

To so vsi elementi, ki omogočajo interakcijo z ali igralci, karakterji ali enotami igre. Tipični primeri so: orožja, stavbe, dvigala, pasti, uroki, posebni talenti, stikala, denar, razne stvari, ki recimo odsevajo proces pridobivanja moči.

Žanri/zvrsti:

V osnovi lahko opisujejo način igranja/igralnost (gameplay), zgodbo ali celo perspektivo.



Razvoj koncepta se začne z visokim konceptom (high concept), ki je tipično zgolj en stavek, opis igre, ki pritegne, npr.:

- Ping pong na računalniku. (Pong)
- Ulična dirkačina, kjer voziš avto za pobeg za mafijo. (Driver)
- Povprečen tehnolog se spopada z pošastmi po nesreči v skrivni raziskovalni ustanovi. (Half-Life)

(Everything sounds much better in English. ;-)

Visok koncept lahko pade v kategorijo obsoječih žanrov, lahko jih združuje, mogoče pa odkriva nov žanr.

Avantura:

- Temelji na zgodbi, ki se navadno naslanja na miselno reševanje situacij, ki nato povzročijo akcije.
- Lahko so tekstovne (Zork) ali grafične (King's Quest).
- Zgodbo lahko pripovedujemo z vidika prve (Seventh Guest) ali tretje osebe (Monkey Island).
- Navadno se ne odvija v realnem času – igralec si vzame čas za razmislek med akcijami, v navideznem svetu igre se med tem nič ne zgodi.
- Navadno interakcija temelji na akcijah z miško (point-and-click), kjer igralec pokaže kaj želi s premikom miške.
- Igralec pričakuje kompleksen svet in dobro zgodbo.

Akcija:

- Realno časovna igra, v kateri mora igralec reagirati hitro na dogodke v navideznem svetu.
- V tem žanru dominirajo igre FPS (First Person Shooter) (Quake, Unreal, Halo).
- Lahko je seveda tudi tretje osebna igra (Tomb Rider), kjer igralec vidi svoj lik kako se giblje v navideznem svetu in mora narediti še kaj več kot le streljati in ubiti.

- Nasprotniki so lahko UI (AI) liki ali drugi igralci preko mreže.

#### Igranje vlog - RPG (Role Playing Game):

- Igralec navadno usmerja skupino likov v seriji nalog.
- Igra se odvija okoli postopnega povečanja zmoglosti in moči lika (Ultima, Might and magic, Final fantasy).
- Igralci pričakujejo, da lahko upravljajo posamezen lik na mikro nivoju – katera orožja nosijo, kakšen oklep imajo na telesu ipd.
- Spopad je pomemben element, s katerim liki dobijo moč, izkušnje in denar za nakup nove opreme.
- Kot pri avanturah tudi pri RPGjevkah igralec pričakuje velik svet s postopno se odvijajočo zgodbo.
- MMORPG (Massively Multiplayer Online RPG) je očitno ekstenzija RPGja, kjer v igri sodeluje veliko število omreženih igralcev. (Ultima online)

#### Strategija:

- Od igralcev zahteva, da upravljajo z omejenim naborom sredstev in dosežejo vnaprej podan cilj.
- Upravljanje virov pogostokrat vključuje odločitve, katere enote ustvariti, kam jih postaviti (prerazporediti) in kdaj jih poslati v akcijo. (Command and conquer)
- Navadno deluje na osnovi vrstnega reda (turn based) – igralec si lahko vzame čas za odločitev, računalnik reagira le, ko je igralec pripravljen.
- RTS (Real Time Strategy) igre sprožajo akcije UI nasprotnika brez, da bi igralec imel na voljo neskončno časa za odločitev.
- UI nasprotnik je lahko tudi drug igralec na mreži.

#### Simulacija:

- Simulacije so igre, ki poskušajo posnemati operacije kompleksnih strojev realnega sveta, npr. bojnih letal, helikoptrov, tankov ipd. (Flight simulator X)
- Bolj resna je simulacija, večja je natančnost in kopija dejanskih kontrol. Igralci pričakujejo, da bodo preživeli ure in ure ob učenju delovanja stroja ter uporabniški priročnik, ki bo razkril tudi detajle.
- Manj resne simulacije, znane pod imenom arkadne simulacije, imajo poenostavljene kontrole, zahtevajo manj učenja in nas navadno manj kaznujejo za napake.
- Nekateri pravijo, da so glasbene igre že samostojen žanr, v vsakem primeru pa gre za bolj ali manj realne simulacije. (Guitar hero worldtour)

#### Šport:

- Športne igre omogočajo igralcu, da je udeležen v svojem najljubšem športu kot igralec ali trener.
- Točno reproducirajo vsa pravila in strategije.
- Igra lahko zajema le posamezne igre, serijo iger ali celo sezono.

#### Pretepačina:

- Navadno gre za igro dveh igralcev, kjer vsak igralec nadzoruje svoj lik preko kombinacije gibov za napad in obrambo. (Seveda, lahko je podžanr športa.)
- Igralec pričakuje, da hitro najde osnovni nabor mehanizmov za napad in obrambo ter jih pili skozi čas.

#### Miselno reševanje situacij (puzzle):

- Intelektualni izziv reševanja problemov. (Caste of dr. Brain, Incredible machine)
- Problemi imajo v tem primeru osrednjo vlogo in niso kot pri avanturah integrirani v širšo zgodbo.

#### Vsakdanje igre (casual):

- Gre za kategorijo, ne žanr.
- Gre za igre, ki imajo enostavna pravila in ne zahtevajo velike časovne obveze!
- Sem spadajo tudi tradicionalne igre, kot so šah, igre s kartami, tri/štiri v vrsto, Reverse ipd. ter enostavne igre, kot so Tetris, Mine sweeper, potapljanje ladjic ipd.
- Igralci navadno želijo hiter pričetek in konec teh iger.
- Enostven uporabniški vmesnik.
- Kratka krivulja učenja.

#### Izobraževanje:

- Namen: izobraževanje skozi igro. (Oregon trail, Reader rabbit)
- Ciljna skupina so predvsem mladi, najmlajši.
- Snovalci morajo tesno sodelovati s strokovnjaki tistega področja, ki zagotovijo, da je vsebina primerna za ciljno skupino.

Resne igre (serious): Njihov osnovni namen ni zabava. Njihov osnovni namen je učiti, raziskovati, promovirati. Izobraževanje torej spada bolj pod to streho resnih iger, ki je pravzaprav kategorija, ne žanr.

Omrežene igre (online): Lahko gre za poljubno igro, če se le odvija preko mreže. (Torej gre za nekakšno lastnost žanrov in kategorije.)

#### Dokumenti:

##### Dokument predloga igre (GDD – Game design/proposal document):

- Visoki koncept (high concept)
- Žanr
- Igralnost (gameplay) – povzetek tega kar igralec počne, ko igra
- Značilke – spisek glavnih prodajnih točk igre (napredna grafika, nova tehnologija, izboljšave glede na recimo prejšnji del ipd.)
- Postavitve – opiše svet igre v nekaj odstavkih in razloži, zakaj je igra zanimiva
- Zgodba – oris zgodbe, če jo igra ima

- Ciljni trg – komu je igra namenjena (otrokom, športnim navdušencem, širokemu trgu ipd.)?
- Ciljna strojna platforma – poleg ciljne platforme podaš oceno zahtev po virih (pomnilnik, hitrost procesorja ipd.)
- *Načrt dela: ocena časa, aktivnosti (mejniki), zaporedja izvedbe (schedule)*
- *Ocena stroškov (budget)*
- Analiza konkurence
- Potrebne kompetence članov razvojne skupine in posledično njihove naloge v tem projektu
- Povzetek dokumenta – še enkrat, vendar z drugimi besedami pove, zakaj bo to odlična igra

V tem dokumentu smo dejansko združili dva dokumenta, pri prvorazrednih (AAA) igrah je namreč poleg GDDja prej potreben še dokument koncepta (concept design document). Njegov bistven namen je predstavitev vizije igre vsem vpletenim stranem (GLC – Green Light Committe: direktor(ji) založnika, distributorja, licenciranja, prodaje, trženja, financ, dislociranih lokacij/mednarodnih enot) in najti konsenz, dobiti odobritev. Na kratko opiše zgodbo, karakterje, izpostavi izboljšave, stopnje in uporabniški vmesnik.

Vsi razvijalci se seveda ne držijo istih terminov in vsebine [5,6] (glej tudi sliko treh faz). GDD namalokrat imenujemo tudi **funkcionalna specifikacija**, *načrt dela* in *ocena stroškov* pa sta lahko velikokrat tudi izvzeti točki iz GDDja. Seveda pa sta nujno potrebni točki za zaključek predproduksijske faze. Če na GDD gledamo kot na zapis problema, potrebujemo še zapis rešitve. Ta se imenuje **tehnična specifikacija** (v angleščini se uporablja tudi izraz TDD – Technical design document).

Tehnična specifikacija [6]:

- Identificira bistvene programerske naloge
- Identificira tehnične izzive/probleme
- Poda načrt reševanja teh izzivov/problemov
- Identificira tehnologijo, opremo, programerje in ostale člane skupine
- Razdela načrt dela (ne le mejniki, kot pri GDD)

Lepa razlaga kaj spada v posamezno specifikacijo se glasi [5]: *Kar gre v igro* in *kako se igra*, brez ozira na to, kako bo kaj dejansko implementirano, spada v funkcionalno specifikacijo, ki je največkrat napisana z vidika uporabnika, igralca, spišejo pa jo oblikovalci igralnosti. *Kako je igra implementirana* in *kako zagotavlja posamezne funkcije* pa spada v tehnično specifikacijo, ki je največkrat napisana z vidika sistema, spišejo pa jo programerji. Absolutne resnice, kje je prav, da potegnemo ločnico med posameznimi dokumenti pa seveda ni.

Predproduksijsko fazo zaključimo, ko imamo načrt dela, oceno stroškov, GDD in TDD. (Pri AAA igrah moramo še enkrat dobiti zeleno luč GLCja.) Z vidika vodenje projektov torej ta faza združuje fazi analize in načrtovanja.

...??? docs v knjigi ...??? konkretni primeri

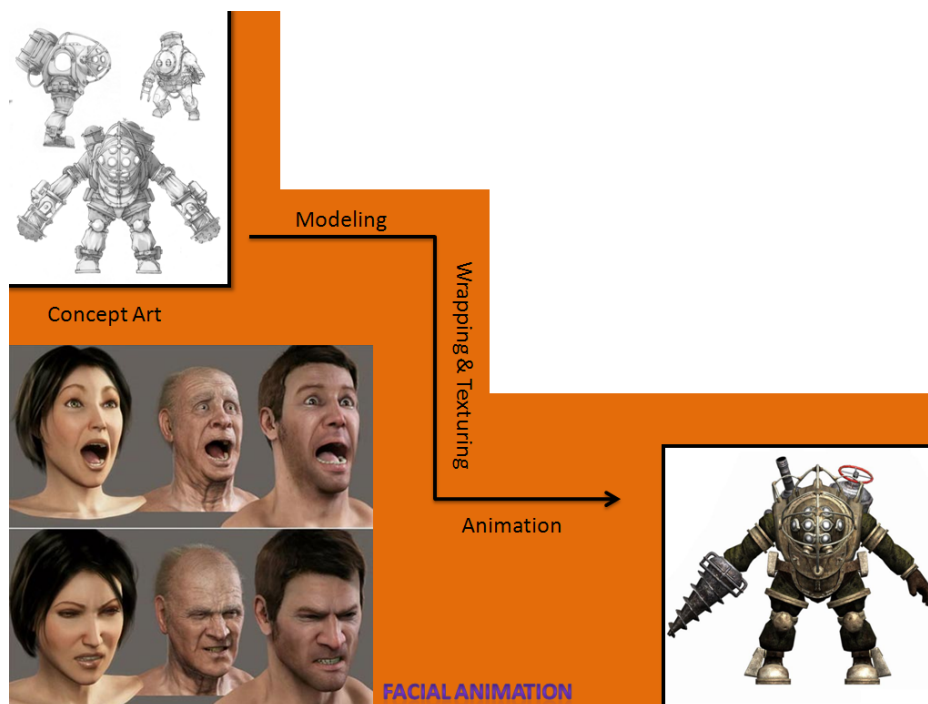
## 2D svet (art):

- Koncepti
- Slike (sprites)
- Teksture
- Uporabniški vmesnik

Orodja: GIMP, Photoshop, GIMPshop

Najbolj gotov format: png (pri pretvorbi/uporabi pazi na transparencu, alpha kanal)

## 3D svet (art):



Formati: x (DirectX), fbx (XNA, 3ds max, Maya), 3ds (3ds max – široko podprt format), md2 (modeli za Quake) – vse to ti nudi zastonjski Blender.

Krivulja učenja vmesnika 3D modelirnih programov je strma kot redko kje, težavnost pa se še kvadrira, če delaš animacijo.

## Poddisciplin programiranja:

- Programiranje načina igranja (gameplay)
- Programiranje UI
- Programiranje orodij
- Programiranje grafike

Razlika med API in SDK:

Application Programming Interface je programski vmesnik do storitev knjižnic, OS: XNA (XNI), DirectX, OpenGL. Software Development Kit je zbirka knjižnic/APIjev, orodij, primerov, dokumentacije/pomoči: iPhone SDK, OpenGL SDK, Java SDK, DirectX SDK ipd.

Pogoni (engines):

- Zaprti pogoni (in-house) (Scimitar/Anvil: Assassin's Creed)
- Celostni pogoni (full tool chain) (UnrealEngine3, Unity)
- Integrirne knjižnice (middleware&integrated libs) (grafika: SDL (Simple direct media layer), OGRE3D (vmesniki (wrapper): pyOgre, Ogre4j, OgreDotNet, MOGRE,...), Irrlicht (vmesniki: Venom, Bindenlicht,...), Allegro (2D), Java3D, JOGL, PyGame (zgrajen na SDL), Tao framework; fizika: Open dynamics engine, Newton game dynamics, Bullet (modelirnik Blender), PhysX; zvok: OpenAL, Fmod, Direct sound)

Integrirano razvojno okolje IDE (Integrated Development Environment): inteligentno povezovanje vseh delov, ki so potrebni za razvoj – barvanje sintakse, sprotno osnovno preverjanje sintakse, razhroščevanje, hitro prevajanje, poganjanje, gradnja uporabniškega vmesnika, dopolnjevanje/predlaganje imen metod, spremenljivk ipd.

- Visual studio
- Kdevelop, Kate
- Xcode
- Eclipse, NetBeans
- FlashDevelop

Kateri programski jezik izbrati? V vseh se da narediti vse. Vsi imajo prav. Poznaš enega, poznaš vse. To je le nekaj sorodnih trditev, ki jim moramo pritrditi! Izbiraš glede na specifične zahteve ali lasten okus, dobro počutje. V nobenem primeru pa že pri hitrostnih 2D akcijah ne bo šlo brez pogovarjanja z grafično kartico! Pri prvorazrednih igrar (AAA) pa brez C/C++ ne bo šlo! Poleg performanc ima tudi največji nabor (zelo) uporabnih knjižnic (glej pogoni zgoraj)!

...???

Orodja za programiranje: Unity, UDK. ...???

**Zvok:** ...???

#### **4. Zaključna misel ali dve, kakšen nasvet**

Imamo razjasnjene osnovne pojme, izbrano okolje, jezik. Kako naprej? Najprej pozabi na svojo mega idejo za igro, za zdaj. Zakaj?

- Najbrž ne veš najbolje, kje začeti.
- Iskal boš eno po eno stvar, kako kaj narediti.
- Verjetno ne boš vsega uspel narediti, kot si si zamislil.
- Iz razočaranja se boš naveličal, še prej pa zamoril vse forumaše.

Zato se najprej le igray! Preglej vsa orodja, knjižnice, primere, pomoč, predelaj (ne le preleteti) tutoriale, najdi forume skupnosti. Za prvi eksperiment naredi največji zamazek igralnosti in uporabi

sličice, ki si jih na hitro naredil recimo kar v Paintu. Ne obremenjuj se z arhitekturo, z ničemer. Potem uporabi knjižnico (lahko v drugem eksperimentu) za enostaven 2D izris, vpleti enostavno fiziko odbojev od robov ekrana/okna. Nato se poigraj s kakšnim 3D pogonom in se pomikaj po ustvarjenem ozemlju z miško in WASD. Potem se počasi vrnemo na našo mega idejo in začnemo snovati vse dele igre, z uporabo omenjenih dokumentov, po principu faz projekta, vse do arhitekture, implementacije, testiranja, odpravljanja napak. Seveda na tem mestu brez objektnega programiranja ne bo več šlo.

Če ti nekaj ne gre, najdi pomoč na to temo in preberi vse okoli problema. Če še vedno ne gre, se pojdi ukvarjat s čem drugim. Stvar prespi, tuš pomaga. ;-) Šele zadnja opcija naj bo forum, no, torej postavljanje vprašanj na forumu.

Dokler nimaš skodirane  $\frac{3}{4}$  igre, uporabiš začasno grafiko, šele potem išči oblikovalce.

### Skupnosti:

[www.igda.org](http://www.igda.org) (International Game Developers Association)

[www.gamasutra.com](http://www.gamasutra.com)

[www.GameCareerGuide.com](http://www.GameCareerGuide.com)

[www.GameDev.com](http://www.GameDev.com)

[www.TigSource.com](http://www.TigSource.com)

[www.SloGameDev.net](http://www.SloGameDev.net)

<http://forums.IndieGamer.com>

### Viri:

[1] <http://www.slideshare.net/AhmedSaker/game-development>

[2] [http://www.cs.miami.edu/~visser/home\\_page/CSC\\_329.html](http://www.cs.miami.edu/~visser/home_page/CSC_329.html)

[3] <http://www.joker.si/article.php?rubrika=17&articleid=3756>

[4] <http://en.wikipedia.org>

[5] [http://www.gamasutra.com/view/feature/3411/the\\_anatomy\\_of\\_a\\_design\\_document\\_.php](http://www.gamasutra.com/view/feature/3411/the_anatomy_of_a_design_document_.php)

[6] Biblija (povsod!)

### Naloga na vajah:

Preberite poglavje "Od ideje do načrta" v skripti vaj in spišite koncept igre za svoj projekt z uporabo predloge:

- zasnovana vsebina igre (game concept)
- kaj od tega bo prototip igralnosti
  - glavna igralnost predstavljena na enem igralnem zaslonu.