VÝVOJ POČÍTAČOVÝCH HER

Historie

- 60. léta
  - Hry si psala obsluha sálových počítačů pro sebe.
- 70. léta
  - Hry vyráběli nadšenci po garážích.
- 80. léta
  - Mnoho firem vyrábí hry, drtivá většina z nich krachuje (drahé hry, ne každý měl počítač, ….).
- 90. léta
  - Propracovanější a dražší hry s GUI na CD -> rozmach (každý má PC, za dobré hry jsou lidé ochotni platit).

Současnost

- krabicové hry s obrovskou reklamou
- většina her mívá demoverze
- online prodej her (které se pak hrají offline)
- online hry (které se hrají jen online)
  - Připojení k internetu má každý, je levné a rychlé.
- mnoho free her zdarma (či možnost donate)
- hry, které žijí z reklam umístěných v nich
- navzdory „pirátství“ trh s hrami vzkvétá

Náklady AAA her

- Minulost:
  - Práce na půl roku pro několik lidí.
- Současnost:
  - Vývoj trvá několik let.
  - Na hře pracují stovky lidí.
    - Zejména programátoři, testeři a grafici, ale také manažeři, muzikanti, dabéři, obchodníci, ….
  - Rekordy:
    - Vývoj StarCraft II trval asi 6 let.
    - Náklady GTA4 převýšily v přepočtu 2,5 miliardy korun.

Rozložení nákladů AAA her

- Platy vývojářů
  - programátoři cca 7k USD měsíčně
  - grafici a designéři cca 6k USD měsíčně
  - 100 programátorů a grafiků cca 8M USD ročně
- Platy ostatních pracovníků
  - manažeři, ale také dabéři, orchestr, ….
- Marketing
  - Velký dopad na prodejnost, průměrně 5M USD.
- Další náklady
  - hardware (vývoj i testování), licence na software, ….

Nejistý byznys

- Vývoj je velmi drahý a trvá opravdu dlouho.
- Úspěch může pokazit mnoho faktorů:
  - konkurenční hra
  - změna zájmu cílové skupiny
  - špatné kritiky
- Firmy mají naprosto odlišné zisky kvartál od kvartálu.
  - Někdy jsou v mínusu, jindy ve velkém plusu.
  - U velkých firem jsou rozdíly stovky milionů USD.

Zisky z prodeje

- Náklady jsou téměř konstantní, téměř vůbec nerostou s počtem prodaných kopií.
- Zisky z krabicových her jsou nejvyšší první dny, u onlinovek je situace jiná (spíše opačná).
- Rekordy:
  - GTA4 - v přepočtu 12 miliard korun první víkend
  - WoW Cataclysm - 3,3 milionů kopií za 24 hodin
  - Call of Duty: Modern Warfare 2 - 20 milionů kopií
  - WoW – nový mount za 25USD, prodáno 100k kusů za 4 hodiny, zisk v přepočtu 50M korun (náklady na den práce grafika mohou činit v přepočtu pár tisíc korun).

Online hry

- Pointou není, aby si miliony hráčů naráz něco koupily, nýbrž aby hru hráli co nejdéle (roky).
- Hru je třeba stále vyvíjet, jinak se hráči nudí.
- Herní chyby nejsou tak zásadním problémem, protože je lze kdykoliv opravit.
- Testovat mohou od jistého bodu i samotní hráči (a reálně jsou ochotni to dělat, i když u AAA her to není běžná praxe).
- Lze snáze vyvíjet i malé hry pro málo hráčů.

Některé AAA online hry

- World of Warcraft – online hra s největším počtem hráčů (asi přes 12 milionů).
- League of Legends – multiplayer online battle arena (několik let starý žánr).
  - oficiálně 32M účtů, pravděpodobně asi 2-3M hráčů
  - turnaje o miliony, profesionální hráči, streamy
- Counter-strike – střílečka.
  - původně ne přímo pouze online hra
  - hraje se většinou online, profesionální hráči, ….

Technické limity online her

- Velké počty hráčů přináší technické problémy.
  - snaha o co největší paralelismus (ne vždy možné)
  - hardware je levný, ale paralelismus má své meze
- Řešení: rozdělit hru na více serverů (světů).
  - Každá velká hra k tomu nakonec dojde.
    - Neexistuje tak výkonný hardware, konektivita; požadavky řádově překračují technické možnosti.
  - Například i LoL, kde je paralelismus snadný (hraje se tisíce nezávislých zápasů naráz), musela být rozdělena na více světů, protože nelze efektivně paralelizovat operace jako přihlášení do hry.

Taxonomie online her

- Online hry lze (jako všechny hry) dělit podle žánru, což není příliš zajímavé.
- Technické provedení:
  - Prohlížečové se hrají v internetových prohlížečích.
    - HTML – server s klienty komunikuje pomocí HTTP a posílá jim HTML stránky.
    - Flash – v prohlížeči běží flash player, ve kterém běží hra, komunikace může probíhat jakkoliv.
  - Klientové hry mají na uživatelské straně program.
    - Typicky klasickou binárku komunikující se serverem.

Vývoj menších her

- Pro malé firmy je lepší tvořit online hry než krabicové, protože je lze postupně rozvíjet a pomalu budovat hráčskou základnu.
- Engine online her je optimalizován na výkon, takže jej lze použít také pro další informační systémy.
  - Zejména engine různých strategických a tahových her lze použít pro téměř libovolný databázový IS.
- Malou skupinu vývojářů může uživit i hráčská základna čítající několik tisíc hráčů.

Fáze vývoje malé online hry

- návrh herní logiky a principů
- implementace engine (lze použít pro více her)
- implementace základních datových struktur
- implementace herní logiky
- hrubé testování programátory
  - probíhá zároveň s implementací herní logiky
- veřejné testování hráči
- ostré spuštění hry
- další vývoj hry, aby hráče nepřestala bavit

Návrh herní logiky

- Každá hra je matematický model, který musí být konzistentní.
- Důležité aspekty hry:
  - Hráč by měl mít pocit, že postupuje vpřed.
    - Hra buď nesmí umožňovat vzájemné poškozování hráčů, nebo musí mít nějakou nepoškoditelnou část, která se vyvíjí v čase.
  - Hra musí být (nějak) hratelná, i když do ní hráč vloží relativně málo času.
  - Měla by být možnost vložit do hry libovolně mnoho času.
  - Náhodné prvky: čím je hra náhodnější, tím úspěšnější jsou nezkušení hráči; na druhou stranu se hůře stane zkušený hráč skutečně dobrým.
  - Hra by měla mít nějaké žebříčky, statistiky, aby se hráči mohli porovnávat (nemusí být kompletní, stačí „top X“).

Implementace engine

- Engine zahrnuje mimo jiné následující:
  - síťová komunikace server-klient (a protokol)
  - komunikace serveru s databází
  - ideálně možnost provozovat více světů (kontexty)
  - paralelismus (ale ne vše se vyplatí paralelizovat)
- Engine by měl být rozumně optimalizován na rychlost i v případě velkého počtu hráčů.
- Je třeba provést benchmarky a testy pro zjištění, co vlastně optimalizovat a jak moc.
  - většinou to NENÍ herní logika, ta bývá nenáročná

Základní datové struktury

- Před samotnou implementací herní logiky je dobré zprovoznit základní datové struktury, které umožní se do hry alespoň přihlásit.
  - Jde o struktury typu „hráč“, „session“ a podobně.
  - Nemusí být ani nutně možné se do hry korektně zaregistrovat, hráče lze vyrobit ručně.
- Následná implementace pak není „na slepo“, ale programátor se může podívat, co vlastně napsal (a vidět to v nějakém prostředí hry).

Implementace herní logiky

- Když už existuje kostra hry, ve které se jde například přihlásit za nějakého hráče, je na čase postupně budovat herní funkcionalitu.
- Postup při implementaci je rozumné volit tak, aby bylo možné nově naimplementované části alespoň částečně vyzkoušet.
- Objeví-li se při implementaci nějaký problém, lze se zamyslet nad menšími změnami návrhu herní logiky, které by mohly problém řešit.

Testování programátory

- Engine je třeba otestovat velmi pečlivě (unit testy, zátěžové simulace, napsat si vlastní testovací programy).
- Herní logiku stačí testovat jen hrubě; zajistit, aby server nepadal, zaznamenávat logy, ladit nejhrubší chyby, které by způsobovaly velké nekonzistence.
- Čas od času provést zátěžové testy.
  - Špatná implementace herní logiky může být zoufale neefektivní; typicky lze snadno řešit.

Veřejné testování hráči

- Hráči jsou ochotni hrát neotestovanou hru plnou chyb a aktivně se účastnit testování.
  - Typickou motivací je, že si mohou jednak zahrát zadarmo a jednak pak v komunitě mohou být ti, kteří se podíleli na vývoji už od začátku.
- Hráči jsou schopni bugy i rozumně reportovat a popsat, mají-li k tomu nějaký informační systém a motivaci.
  - Motivací může být například soutěž o nejlepšího testera, zveřejnění nicků testerů, možnost hrát pak hru zdarma, ….

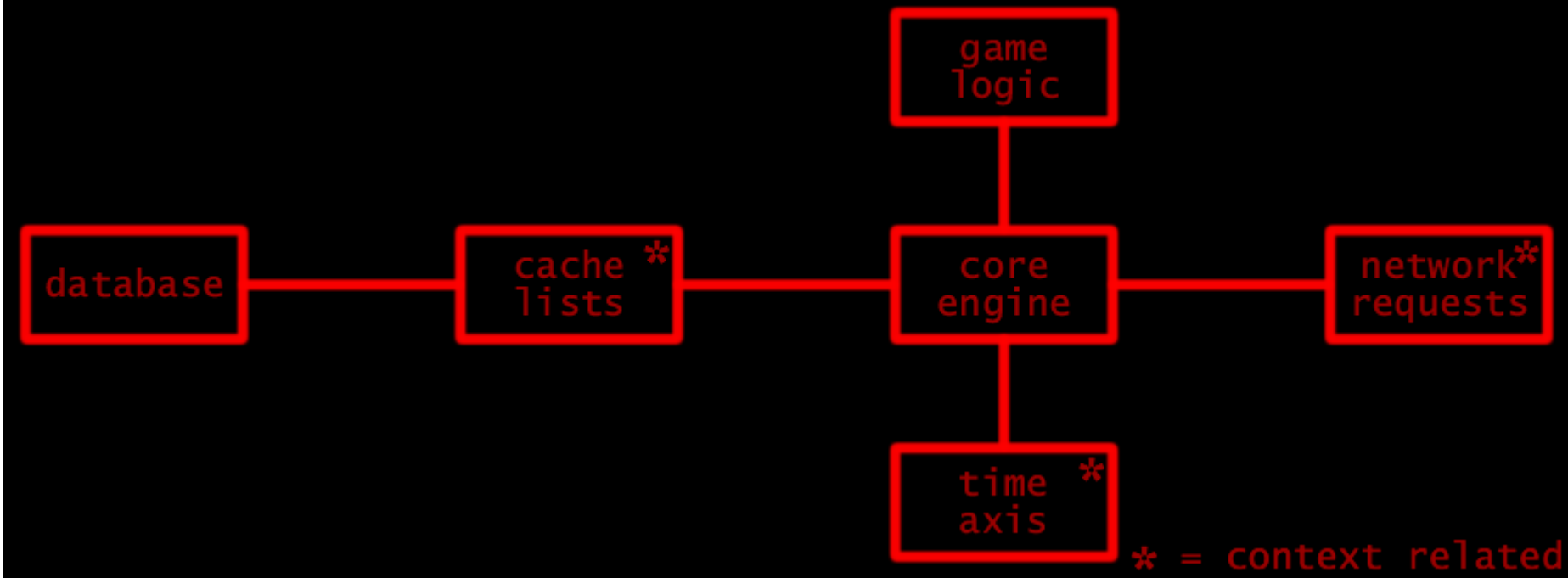
Ostré spuštění hry

- Běží-li hra nějakou dobu v testovacím režimu a nevyskytují-li se už další chyby, následuje „ostrý start".
- Datum a čas „ostrého startu" je dobré sdělit komunitě dlouho předem, nechat někde běžet třeba odpočítávání a podobně.
- Objeví-li se ve hře chyba v ostrém provozu, většinou to hráčům moc nevadí, stává se to i u AAA her.
  - Vidí-li hráčská komunita aktivní práci a rychlé reakce tvůrců hry při opravování chyb, toleruje celkem cokoli.

Další vývoj

- Online hra nemůže zůstat statická, protože by ji hráči po čase opustili.
  - Stejně jako krabicové, tam to ale nevadí.
- Změny je dobré avizovat dlouho dopředu, hráči se pak mají na co těšit.
  - Je důležité dodržet termín, komunita typicky reaguje lépe na vydání zabugovaného patche, než na opoždění vydání něčeho, co jim bylo slíbeno.
  - Změny je dobré avizovat předem; každá změna má pak efekt jednou při oznámení, podruhé při zavedení (v mezičase o tom komunita diskutuje).

Architektura HIAX engine



- Cache listy zefektivňují a zapouzdřují správu dat nad libovolným uložištěm.
- Na časovou osu lze uložit requesty, které pak nemusí aktivovat klient, ale aktivují se samy.

Modálnost architektury

- Databáze: pro použití jakékoli databáze či uložiště stačí jen implementovat interface cache listů, nic jiného se vůbec nemusí měnit.
- Časová osa: některé hry (či IS) ji nevyužívají.
- Pro změnu jedné hry na jinou (či jiný IS) je třeba:
  - Nahradit modul „game logic", např. vyměnit hru za rezervační systém sítě hotelů.
  - Vytvořit nové cache listy pro nové entity (nicméně je stačí jen podědit od abstraktní třídy cache list), např. entitu „hvězdná loď" nahradit entitou „hotelový pokoj".
  - Vytvořit novou databázi (SQL dotazy do konkrétní databáze jsou součástí cache listů, díky cache listům nejsou třeba příliš složité dotazy).
  - Nahradit sadu dotazů v komunikačním protokolu, např. místo requestu „vystřel na nepřítele" bude „rezervuj pokoj s vířivkou".

DĚKUJI ZA POZORNOST