CISC 326
Game Architecture

Module 02:
Challenges In Game Development
Ahmed E. Hassan (with slides from Dr. Scott Grant and Nick Graham)
World of Warcraft

- Released November 23, 2004 after a closed beta
- Over 100 million lifetime players and 500 million characters
- Nearly 1 million PvP instances and 700k PvP instances each day
- Nearly 3 million daily trades in the auction house
Eldre’Thalas is Full
Position in queue: 682
Estimated time: 68 min

Change Realm
Critical Lack of Server Capacity

- Stress testing was done in beta, but was not sufficient
- Long queue times prevented players from playing
- Server migration was unacceptable in many cases
SimCity 2013

- Released March 5, 2013
- Enormous sales success immediately following release
- Traditionally a single-player game
- New version required continuous access to EA’s servers
Unable to load the city at this time. Please try again.
Critical Lack of Server Capacity

- Players unable to gain access to game
- EA responded by removing game features, adding new servers
- Amazon.com bombarded with negative reviews
- EA finally announced free game to all people who purchased SimCity
Making Games Is Easy?

- There is a lot more to games than fancy graphics
- World of Warcraft has 13,250 server computers with 75,000 CPU cores
- Managed by a staff of 68
- 10 data centres in US, Europe, Asia

Source: Rich Miller, WoW’s Back End: 10 Data Centers, 75,000 Cores, Data Centre Knowledge, Nov 25, 2009
Hi everyone and thanks for your attention.

We just wanted to reassure you that we do know it’s an extremely annoying and frustrating time for some of you at the moment and we are working around the clock to sort out those issues that you are having. The first patch has just gone up - it's not trouble-free we know and are fixing with a hotfix, but there will be another next week and every week after that till the problems are gone.

At the moment it may seem that the
Total War: Rome 2

At the moment it may seem that the changes are slight, but they aren’t the only ones we are working on currently and bigger changes are happening now for future patches.

If you are having a problem, it is totally unacceptable and a big deal for us, please know that we are spending all our dev effort on fixing outstanding issues.

Mike Simpson, Creative Director, Creative Assembly, Sept 6, 2013

Total War: Rome 2

- Big lesson

- Simpson claimed that game ran correctly on 98% of systems

- But 2% of several million is still a large number

- Vast plethora of PC hardware (different processors, memory, graphics cards, OS and driver versions, ...)

Ad hoc techniques don’t work. Video games are software too.
Challenges in Game Development

- Size of teams
- Strict timeline
- Range of cooperating roles
- Complex development environment
- Low success rates
Challenges in Game Development

- Size of teams
- Strict timeline
- Range of cooperating roles
- Complex development environment
- Low success rates
Game Development Teams

- For “AAA” games, teams typically 10-100 people
- Multi-disciplinary
  - Producer
  - Designer(s)
  - Programmers
  - Level designers
  - Modellers
  - Animators
  - Sound Engineers
  - Musicians
  - Quest developers
  - Writers
  - Testers
  - Actors
  - …
Red Alert 2

- 3 producers
- 10 designers
- 3 story writers
- 16 artists
- 18 programmers
- 3 audio
- 31 video production
- 21 quality assurance
- 9 marketing
- 41 live actors
- 26 voice actors
Battlefield 1942

- Producers: 2
- Designers: 3
- Programmers: 11
- Artists: 12
- Sound: 2
- Music: 1
- Testers: 51
- Documenters: 2
- Voice actors: 18
Civilization 4

- Producers: 3
- Designers: 2
- Programmers: 18
- Writers: 5
- Artists: 34
- Voice actors: 1
- QA: 26
But not all games are AAA

- Mojang: 3 at first, now 40
- Team Meat: 2
  Programmer, Artist
  (+Composer, QA, etc.)
- Frictional Games: 12
  (+contractors, partners, etc.)
- Subset Games: 2
  Programmer, Artist
  (+Writer, Composer)
Challenges in Game Development

- Size of teams
- **Strict timeline**
- Range of cooperating roles
- Complex development environment
- Low success rates
Strict Timelines

- **Time-sensitive topics:** E.g., sports events, movie tie-ins
- **Need to recoup investment**
- **Changing technology makes game stale:** Duke Nukem Forever, development 1996-2011, changed engine at least twice requiring complete rewrite each time
Challenges in Game Development

- Size of teams
- Strict timeline
- Range of cooperating roles
- Complex development environment
- Low success rates
Range of Co-operating Roles

- Many sub-specialities on development team
  - Artists - modellers, animators, lighting specialists
  - Designers - level designers, quest designers, writers
  - Sound specialists

- Schedule dependencies
- Problems crossing organizational hierarchy
Challenges in Game Development

- Size of teams
- Strict timeline
- Range of cooperating roles
- Complex development environment
- Low success rates
Xbox One

- Released November 22, 2013
- 2 x 1.75 GHz quad core AMD Jaguar processor
- 853 MHz AMD Radeon GCN
- 8 GB DDR3 RAM
- Up to 8 wireless game controllers
Playstation 4

- To released Nov. 15, 2013
- 8 core AMD x86-64 Processor
- 8 GB GDDR5 RAM
- AMD Radeon GPU
Playstation 3

- Released December 2006
- 3.2 GHz Cell Processor
- 7 SPE’s – 128 bit SIMD RISC processor
- connected by 10 GBps bus
- 1 PPE as controller
- 256 MB system memory, 256 MB on graphics card
- Custom “Reality Synthesizer” Graphics Processor (NVIDIA/Sony)
- Up to 7 Bluetooth controllers
Consoles: Programming

- Current consoles introduced significantly different programming architectures
  - 6 cores in Xbox 360; 8 cores in Xbox One; 7 cores in PlayStation 3
  - Extensive vector processing capabilities
  - Difficulty of finding this much parallelism in games
    - Obvious candidate: AI
  - Difficulty of concurrent processing
  - In Xbox 360, GPU and CPU share memory
  - Many years required to learn how to develop for new consoles
Consoles: Development

- Proprietary software development kit available for Windows PCs
  - Develop, debug on PC
  - Cross-compile to console

- Consoles do not necessarily make it easy to port to other consoles
  - “native” vs. “ported” vs “cross-platform” implementation
Consoles: Software Delivery

- Still typically DVD/Blu-ray delivery
  - Digital download gaining traction

- Traditionally consoles had no/small HD, games played directly from DVD
  - Implies no opportunity for patching
  - First generation Xbox 360 came in “arcade” version with no HD!

- Compare to PC games where patches routinely delivered after release, require download by players

- Requires rock-solid software engineering, QA processes
Home Computer

- Enormous variety of hardware and SW platforms (e.g., OS: Windows, MacOS, Linux)

- Windows PC’s have open hardware/software environments (contrast to consoles with uniform env.)
  - Video card, CPU, memory differ in capability by orders of magnitude
  - Different versions of DirectX, video card drivers cause enormous difficulties with QA
Portable Game Devices

- Handheld devices for gaming
- Extension into other entertainment areas – movies, MP3’s, social networking
- E.g. PlayStation Vita, Nintendo 3DS
Mobile Devices

- Cell Phones\Tablets
  - Touch-based input
  - Powerful 3D GPUs
  - iPhone: Objective C
  - Android: Java / C
  - Windows Phone: C#/XNA

- Largest install base of all gaming platforms

- Programming challenge in diversity of devices
Cross-Platform Development

Several game development engines now support wide range of platforms

- **Unreal**: Windows, OS X, Linux, PlayStation, Wii, iOS, Android, HTML 5

- **Unity**: iOS, Android, BBOS, Windows Phone, Windows, Mac, Linux, PlayStation, Xbox

- **Source**: Windows, MacOS, Linux, Xbox, PlayStation

- **Frostbite**: Windows, Playstation, Xbox
Challenges in Game Development

- Size of teams
- Strict timeline
- Range of cooperating roles
- Complex development environment
- Low success rates
Many games fail to make a profit

- Apple App Store
  - ~59% of games don’t break even
  - ~80% of developers can’t sustain a business on apps alone
- Profitable vs. Financial Success

http://arstechnica.com/apple/2012/05/ios-app-success-is-a-lottery-and-60-of-developers-dont-break-even/
http://www.gartner.com/newsroom/id/2648515
A Rapidly Growing Market

The Road To 1M Users

- 9 Years
- 9 Months
- 9 Days
- 3 Days

$16,910.556
(Avg. Daily Revenue Nov 2012)

$3,402.544

51% (All of 2012)

43% (Aug-Nov 2012)

7.7% (All of 2012)

NASDAQ

DISTIMO
Games drive most of the revenue and downloads.

30 – 35% of all downloads are Games
over 60% of revenues from Games.
How many apps before Angry bird?

1\textsuperscript{st} app

10\textsuperscript{th} app

20\textsuperscript{th} app

Over 50 apps
How many apps before Angry bird?

1st app

10th app

20th app

Over 50 apps
How much does it cost?

- **Frogger (1982)**: $5 million
  - (mostly marketing)
- **EverQuest (1999)**: $3 million
- **Lord of the Rings: The Two Towers (2002)**: $20 million
- **Half-Life 2 (2004)**: $40 million
- **World of Warcraft (2004-2008)**: $200 million
- **Rift (2011)**: $60-70 million

http://kotaku.com/how-much-does-it-cost-to-make-a-big-video-game-1501413649
50% of developers make $682/year
35K average cost of developing an app
51 years to break-even!!

Angry Bird is Rovio’s 52nd app!!
The Indie Path

• Solo indie developers earned an average income of $11,812 in 2013, down 49% from 2012’s $23,130 average

• 57% of indie game developers (including both solo indies and members of indie teams across all pay ranges) made under $500 in game sales

Challenges in Game Development

- Size of teams
- Strict timeline
- Range of cooperating roles
- Complex development environment
- Low success rates
Software Architecture

- The blueprint underlying complex software systems
- Deliberate design
- Trade-off analysis
- Investigation of system properties before implementation
- Discovery of potential reuse
Early Architecture

- Architectural design is often the first stage in the software design process
  - Identifies the main structural components in a system and the relationships between them

- Even in agile software process models, the early development process should be concerned with establishing an overall system architecture
  - Refactoring components is easy
  - Refactoring architectures is hard
We wish to build a single player game with an online high score list available through the web.
Two nodes represent the player’s computer and the computer hosting the web server.

The player’s computer has a component running the game.

The web server node has a component representing the web server, and another component representing the high score web application.
A node is a computational platform such as a:
- physical computer
- virtual machine
- process
A component is a modular part of the system such as:

- A web server comprising > 1,000,000 lines of C++ code
- A simple service for generating high scores written in PHP
- A game client of a few thousand lines of code

A component has an interface specifying how other components communicate with it
A link represents some relationship between nodes or components such as:

- An HTTP connection between a web browser or other software component and a web server
- A read/write relationship between a process and a file
- A method call between two processes in a program
Adding High Score Viewer

- A new node represents the computer of someone trying to access the high score via a standard web browser

- Uses web server to access high score
Adding High Score Viewer

- In fact, any number of players, high score viewers can connect to web server

- Web browser and game client could be on same node
Where is the high score file stored?

- **Option 1**: use a file on the web server
- Use XML/JSON format to store file

```xml
<highscorefile>
  <highscore name="Iron Man" score="2850000"/>
  <highscore name="Black Widow" score="2720000"/>
  ...
</highscorefile>
```
Problems

- Scalability
- Availability

```xml
<highscorefile>
  <highscore name="Iron Man"
    score="2850000" />
  <highscore name="Black Widow"
    score="2720000" />
  ... 
</highscorefile>
```
Where is the high score file stored?

- **Option 2**: store the data on a database

- **Pro**:
  - Code reuse

- **Con**:
  - Scalability
  - Code Complexity
  - Cost
Availability, Scalability

- Can have redundant web servers
Final solution

- Load balancer forwards requests to one of a pool of web servers
- **Availability** - if one web server fails, another can take its place
- **Scalability** - arbitrary since always possible to add more web servers to pool
Similar Architectures

- Although each software system is unique, systems in the same application domain often have similar architectures that reflect the fundamental concepts of the domain.
Trade-offs

- Development, deployment time
- Reuse (existing database versus custom file format)
- Scalability
- Availability

...important to know requirements to help make best choice

...architectural thinking can help expose requirements
Trade-offs

- **Performance**
  - Localize critical operations within a small number of components on the same computer vs. distributing them across the network

- **Availability**
  - Include redundant components so that it is possible to replace and update components without stopping the system