CS 563 Advanced Topics in Computer Graphics

*Designing a PC Game*

by Linna Ma
- Background
  - Six game characteristics
  - History
  - Kinds of Games
- Design
  - Platforms
  - Languages
  - Components
  - Tools
    - *Game engines*
- Trends
Rules

Player influences outcome

Optional consequences

Quantifiable outcome associated with player(s)

Success criteria

Chance-based games

Simulations / Toys

General games
Sid Meier: “A game is a series of interesting choices.”

- Pre-defined rules (a dynamic system)
- Goals
- Variable outcome associated with the player(s)
- Optional real-world consequences. (You can place a bet on the outcome of a game, but you can also choose not to.)
- Non-gambling: The player influences the outcome.
- In a game with a theme, a game is a representation of a fictive world.
What do players want?

- Players want: a challenge, to socialize, a dynamic solitaire experience, bragging rights, emotional experience, fantasize.
- Players expect: A consistent world, to understand the game-world’s bounds, reasonable solutions to work, direction, to accomplish a task incrementally, to be immersed, to fail, a fair chance, to not need to repeat themselves, to not get hopelessly stuck, to do - not to watch.
- 1970’s and Earlier
  - Space War developed at MIT 1961
    - Used vector graphics on PDP-1
  - Star Trek popular in 70’s
    - Mainframes
    - Almost no graphics
1980’s

Adventure

Platform

Action

- Action-adventure: Gauntlet, Jet Set Willy.
- Various: Pac-man (maze), Pengo, Qix, Frogger, Star Wars
Background - History

Sim City
(Simulation, but no goal)
Sims (Sim city in a social context)
Background - History

Doom - First person shooter
(action + some exploration from adventure + first person perspective)
Real time strategy
(board game + real-time)
1994 DOOM released by Id; Myst becomes the all time biggest game

1995 Full-motion video capture becomes part a game

1996 Multi-player gaming becomes serious

1998 Lots of very good PC games Play station is console king
2000

- Development moves from PC’s to consoles
- Playstation II
- Diablo II
- Sims
2003

Online Game

- SIMS Superstar
- Star Wars Galaxies >275,000 registered users
- Ports from console games
  - Warcraft III
  - Half Life 2
  - UT 2003
1960s
- First timesharing systems
- BASIC programming language
- Microprocessor invented, 1971

1970s
- Mainframe computer games
- Mainframe computer games

1980s
- Personal computer games
- Commodore 64
- Apple II
- Arcade video games
- Atari 2600
- Odyssey
- "Multimedia machines"

1990s
- Arcade game consoles
- PlayStation
- Dreamcast
- Nintendo 64
- "Multimedia machines"

2000s
- Online games
- CD-ROM
- VGA card
- Sound Blaster
- IBM PC
- America Online

2010s
- Interactive Television?
- Fiber to the home
- ADSL/Cable
- MMORPGs
- Internet gaming

Crash!

Background - History

Cell Phone Games
- GameCube
- Xbox
- PlayStation 2
- Dreamcast
- Nintendo 64
- PlayStation
- SNES
- Genesis
- Nintendo

Large LBE devices
- 3DO
- CD-I
- Gambling Machines
- PlayStation
- PlayStation 2
- Dreamcast
- PlayStation
- Dreamcast
- GameCube
- PlayStation

Hand-helds
- CRASH!
- Bally
- 3D graphics hardware
- 1980s
- 2000s
- 2010s
- Interactive Television?
Different genres - different pleasures action

- Action Games
- Role-Playing Games (RPG)
- Adventure Games
- Strategy Games
- Simulations (Vehicle, Construction and Management)
- Sports Games
- Artificial Life, Puzzle Games, and Other Genres
- Online Games
- Computer – Win95/98/2000/XP Direct-X (Mac or PC)
- Video games – console games
- Multi-player games
  - Peer to peer – modem
  - PC network
  - Client server
Design - Languages

- Assembly
- C
- C++
- VB
- Java
- Flash
Design - Tools

- Sound digitizer
- Music editor or sequencer
- Graphics tools like Photoshop
- Video digitizer
- Video camera
- Graphics library
- Game engines
It’s often hard to break up a game into distinct parts, because there is usually too much overlap to separate them. But, here are four broad components:

- **Game Engines**
- Rules and Mechanics
- User Interface
- Content and Challenges
Sometimes when a developer or player uses the term “engine” they really mean “graphics engine”. But a game engine encompasses much more. Game engines:

- Power the graphics and sound
- Power the AI
- Power the physics and interactions in the game
- Describe the nature of the game space
- Define the parameters of game objects
- Define the space of possibilities in the game world
Characteristics

- Is broad, adaptable, and extensible.
- Firmly encodes all non-mutable design decisions.
- Allows parameters for all mutable design decisions.
- Should outline the gameplay and challenge possibilities.
- Determines the overall game architecture.
- Is coded so that new design decisions leave it unchanged.
A typical time budget for a game is:

- Graphics: 86%
- Animation: 5%
- Sound: 5%
- AI and game play: 4%
What is the difference between a 3D library and a 3D engine?

- 3D libraries, e.g. OpenGL and Direct3D, only provide an interface for rendering 3D graphics. Supports triangles, materials, textures, etc.

- 3D Engines, e.g. OGRE, Torque etc., provide a higher-level interface for organizing 3D environments. Supports objects, terrains, environments, particle effects, etc.
Design – Game Engines Case

Inside Quake

Client-server (v.s. DOOM had a synchronous peer-to-peer networking architecture)

Run on non-multitasking DOS

Help to have identical code for single-player and multiplayer modes

Communication: reliable packet
The structure of Quake levels

--- = polygonal wall (from above)
---- = splitting surface (from above)
A-F = node
1-7 = leaf
a-i = polygonal surface

○ = node
□ = leaf (polys in leaf)
Culling and visible surface determination (VSD):
An ideal VSD architecture would draw only visible parts of visible polygons.
Culling and visible surface determination (VSD):
Node E describes the shaded subspace, which contains leaves 5, 6, and 7, and node F.
Quake's beam tree effectively partitioned the screen into 2-D regions.
Quake's beam tree was composed of 3-D wedges, or beams, projecting out from the viewpoint to polygon edges.
Intersecting span sorting. Polygons A and B are viewed from above.
Design – Game Engines Case

Abutting span sorting. Polygons A and B are viewed from above.

Viewpoint

visible portion of polygon A

Polygon B starts here, abutting polygon A. At this location, both polygons have the same 1/z value.

invisible portion of polygon A

visible portion of polygon B
Quake’s lighting model:
Adding an extra vertex directly beneath a light.

Wall is a single polygon before adding a light vertex

Wall becomes four polygons after adding a light vertex directly beneath a light
Quake’s lighting model: Gouraud shading varies with polygon screen orientation.

Design – Game Engines Case
Quake’s lighting model: A surface is built by tiling the texture and lighting the texels from the light map.
Quake's triangle-model drawing pipeline.
Design – Game Engines Case

Original triangle
(vertices have already been drawn)

Split vertex
(drawn as soon as it’s identified)

Two new triangles, each of which is recursively processed the same way

One recursive subdivision triangle-drawing step.
Summarize

Develop 3D games by

- Creating 3D geometry
- Applying textures
- Embedding sources of light
- Embedding various object into the game e.g. weapons, vehicles, opponents, etc.
Design – Game Engines

Example: UT2004
Design – Game Engines
Design – Game Engines
- More games using pre-made engines
  - Quake, Unreal, etc
- Still have to tweak and modify a lot
  - Flexibility and good documentation
- Completely generic game engine not possible
  - Too many genre specific optimizations needed
- 3D technology
  - Motion capture
- Virtual reality
  - Improve interaction devices
- Really good AI
  - Allow games to react to users dynamically
  - Build really good opponents
- 3D technology
  - Motion capture
- Virtual reality
  - Improve interaction devices
- Really good AI
  - Allow games to react to users dynamically
  - Build really good opponents
References

- Book: Andrew Rollings and Ernest Adams “Game Design”
- Book: “The illustrated history of electronic games”
- “Ramblings in Real Time” Michael Abrash 2001