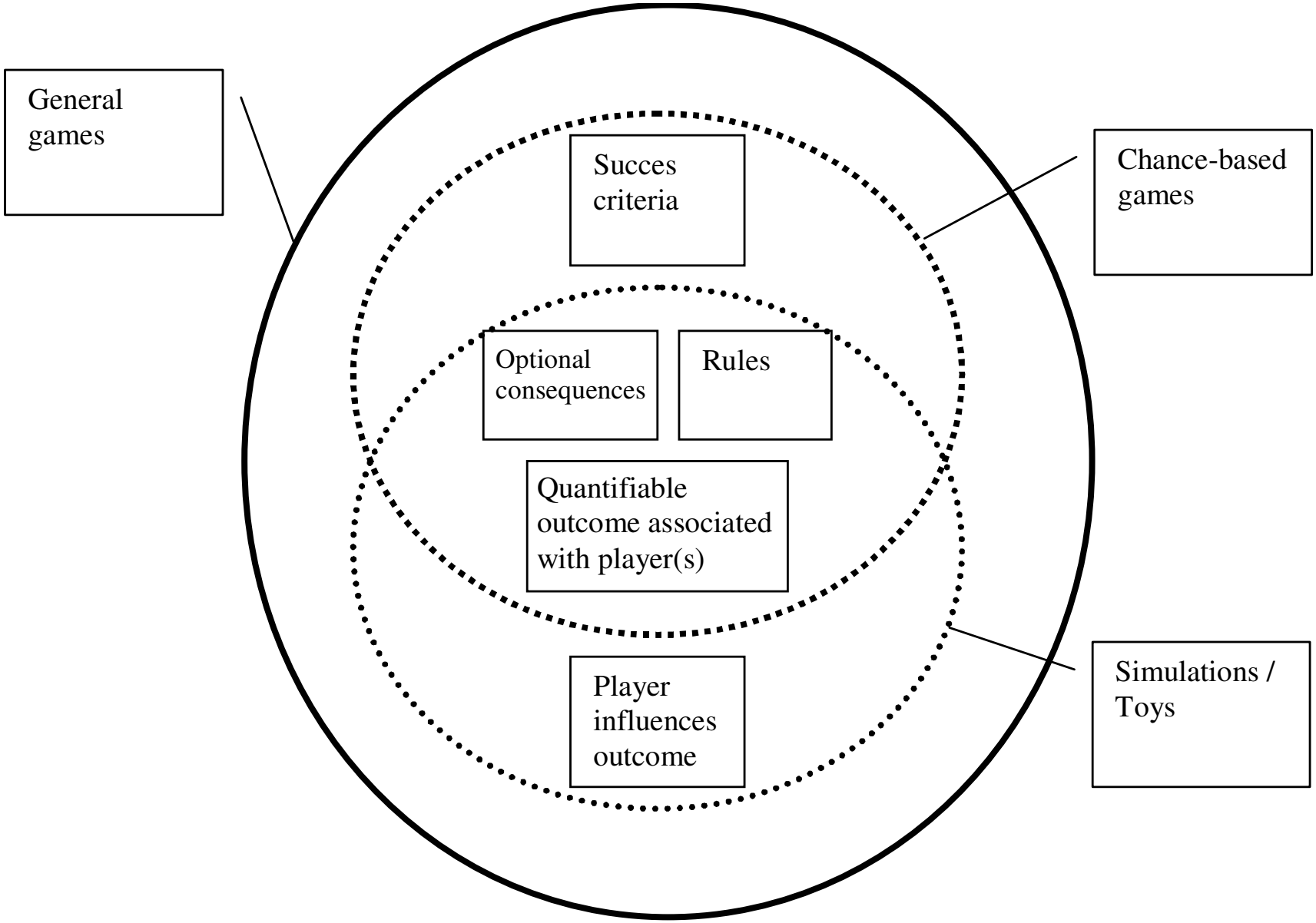
CS 563 Advanced Topics in Computer Graphics
Designing a PC Game

by Linna Ma

Topic

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

Background - Characteristics



Background - Characteristics

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.

Background - Characteristics

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.

Background - History

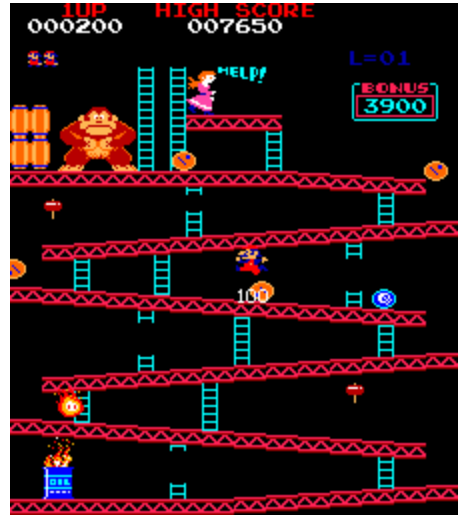
- 1970's and Earlier
  - Space War developed at MIT 1961
    - Used vector graphics on PDP-1
  - Star Trek popular in 70's
    - Mainframes
    - Almost no graphics

Background - History

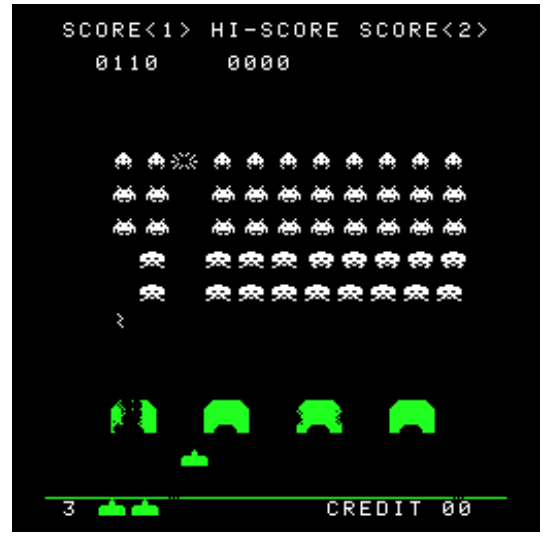
1980's

Adventure

Platform



Action



Driving

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

Background - History

Sim City
(Simulation, but no goal)

Background - History

Sims (Sim city in a social context)

Background - History

Doom - First person shooter
(action + some exploration from adventure + first person perspective)

Background - History

Real time strategy
(board game + real-time)

Background - History

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

Background - History

2000

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

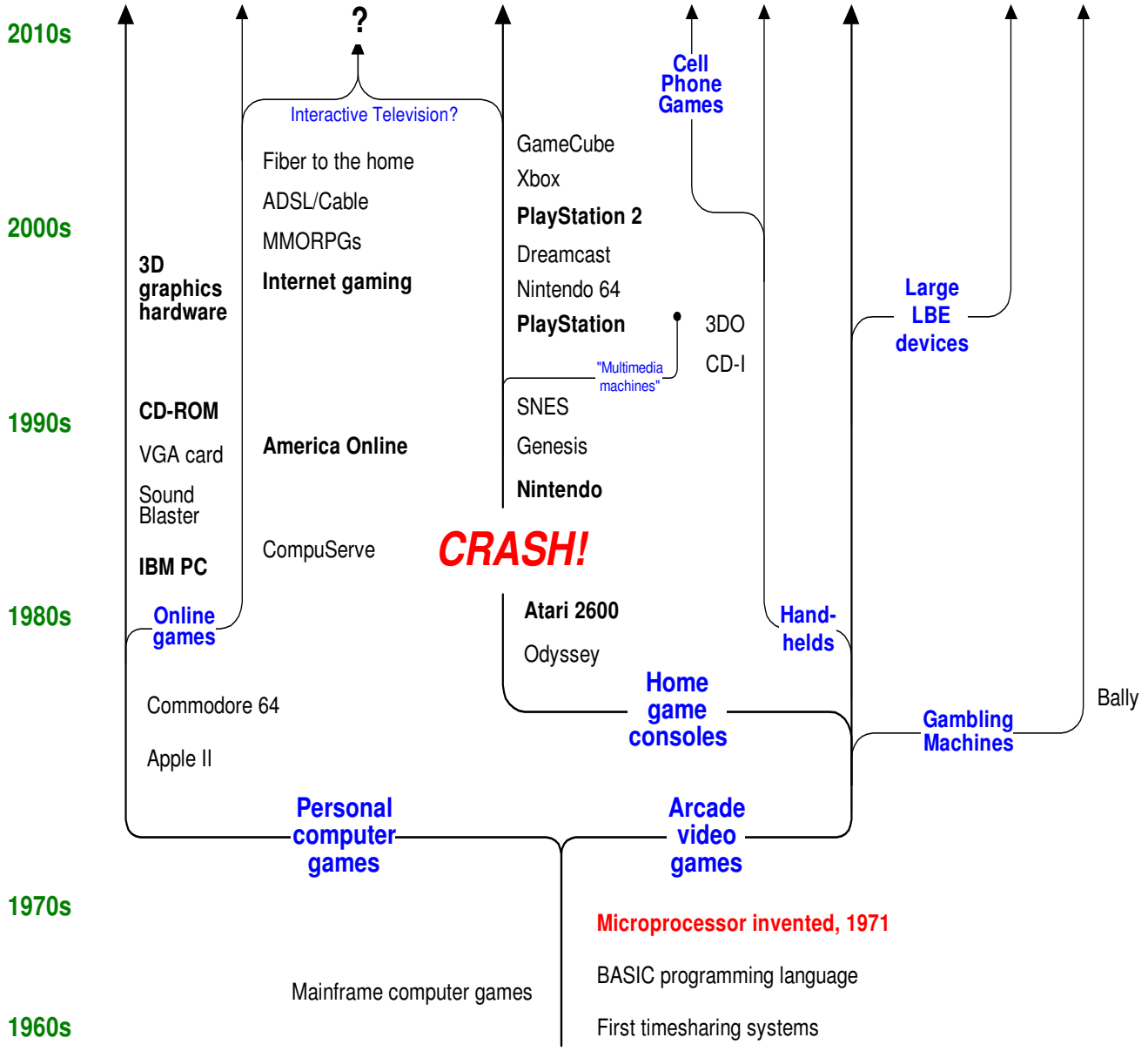
Background - History

2003

Online Game

- SIMS Superstar
- Star Wars Galaxies >275,000 registered users
- Ports from console games
  - Warcraft III
  - Half Life 2
  - UT 2003

Background - History



Background - Genres

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

Design - Platforms

- 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***

Design - Components

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

Design – Game Engines

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

Design – Game Engines

Example Screens

Design – Game Engines

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.

Design – Game Engines

A typical time budget for a game is:

Graphics	86%
Animation	5%
Sound	5%
AI and game play	4%

Design – Game Engines

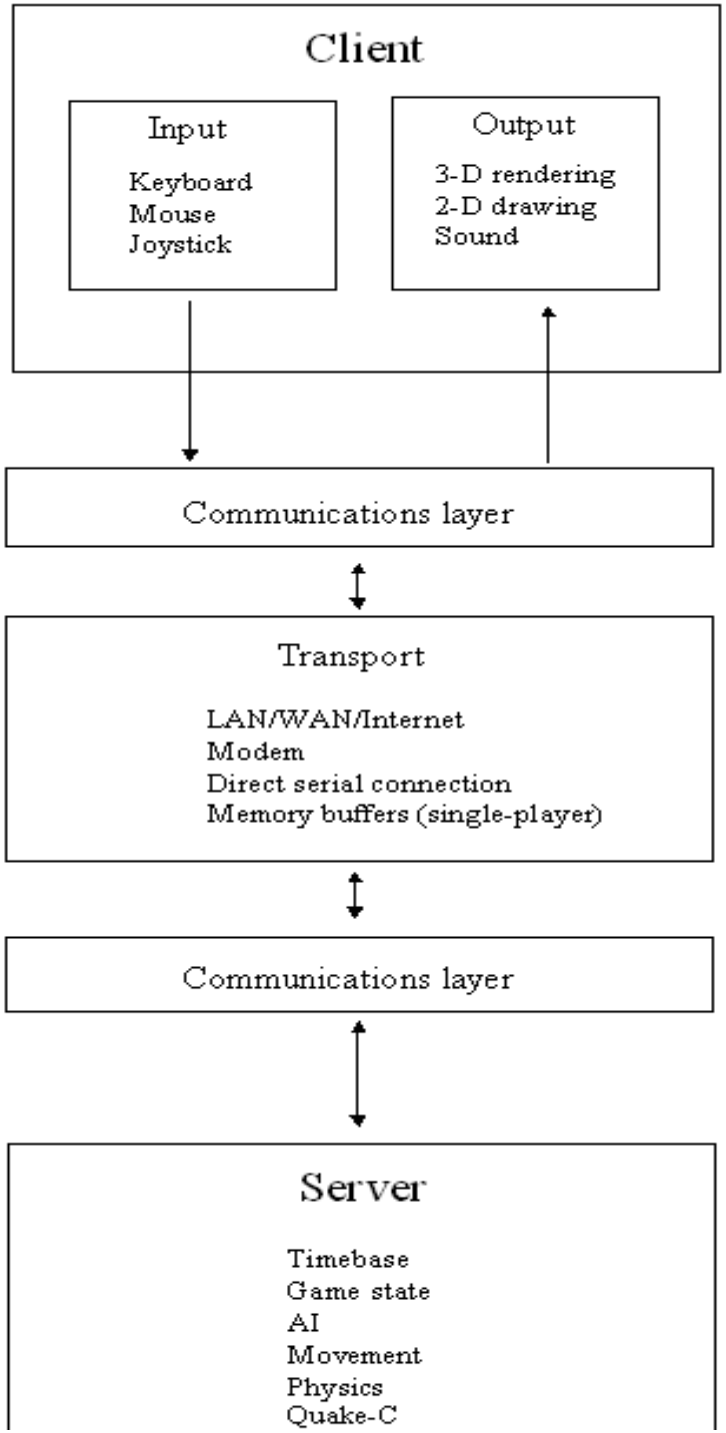
What is the difference between a 3D library and a 3D engine?

- 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

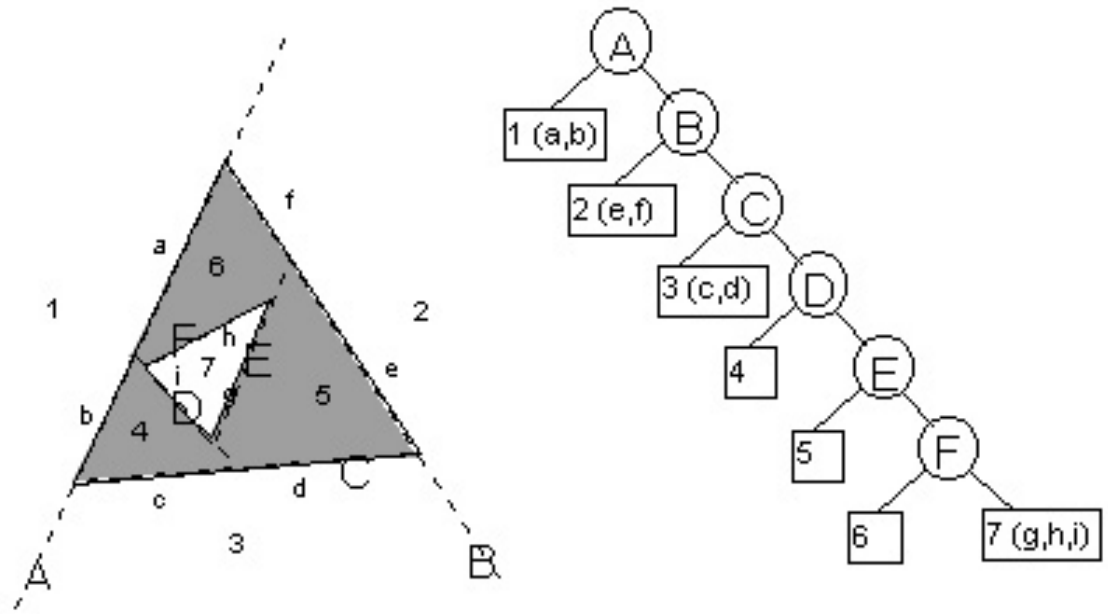
Client-server (v.s.DOOM had a synchronous peer-to-peer networking architecture)

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

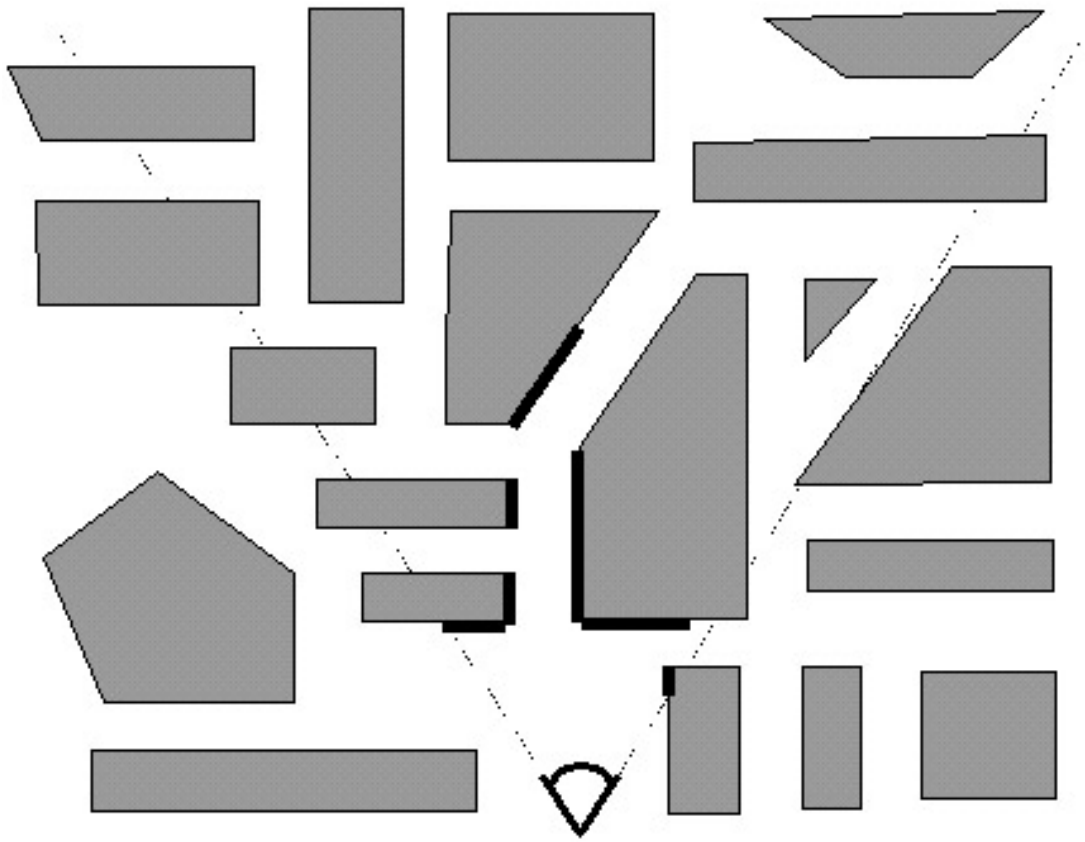
◯ = node

□ = leaf (polys in leaf)

Design – Game Engines Case

Culling and visible surface determination (VSD):

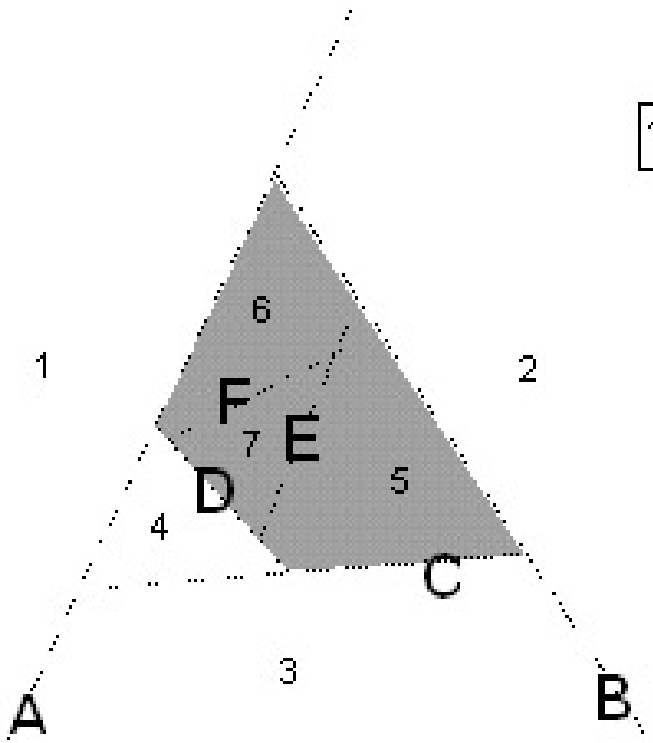
An ideal VSD architecture would draw only visible parts of visible polygons.

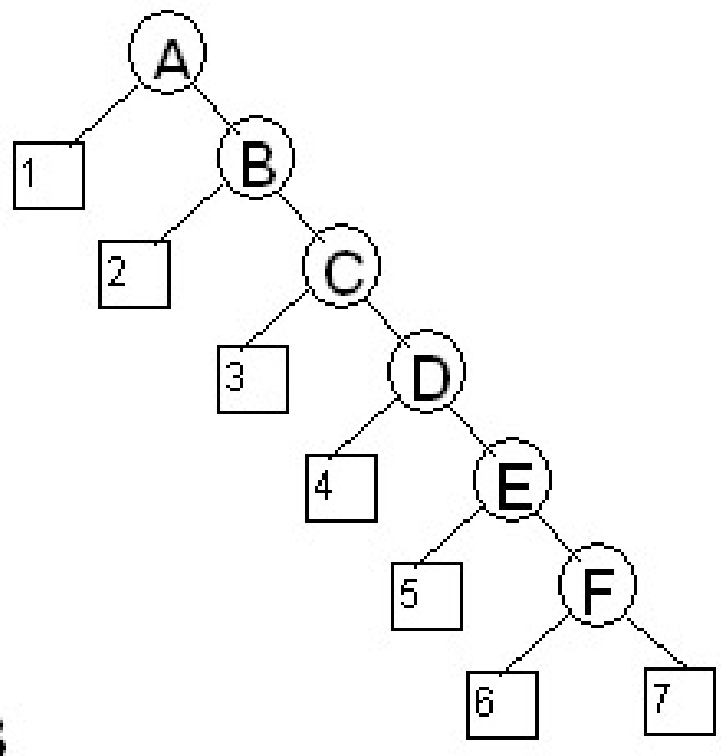


Design – Game Engines Case

Culling and visible surface determination (VSD):

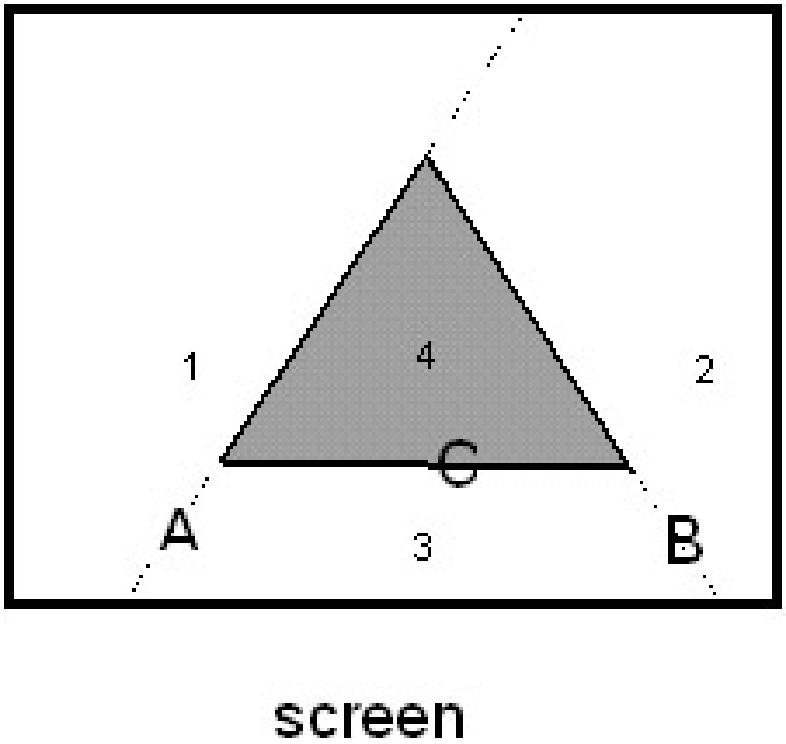
Node E describes the shaded subspace, which contains leaves 5, 6, and 7, and node F.

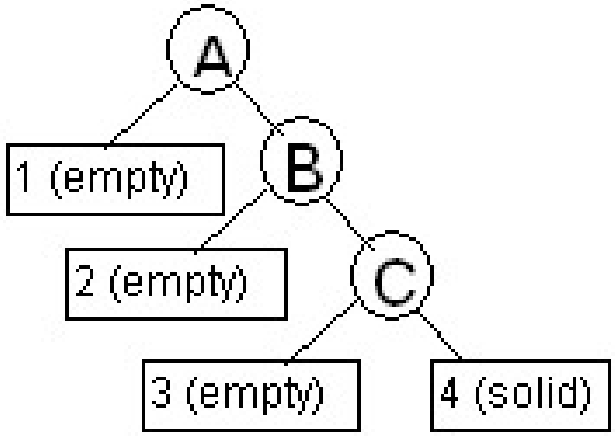




Design – Game Engines Case

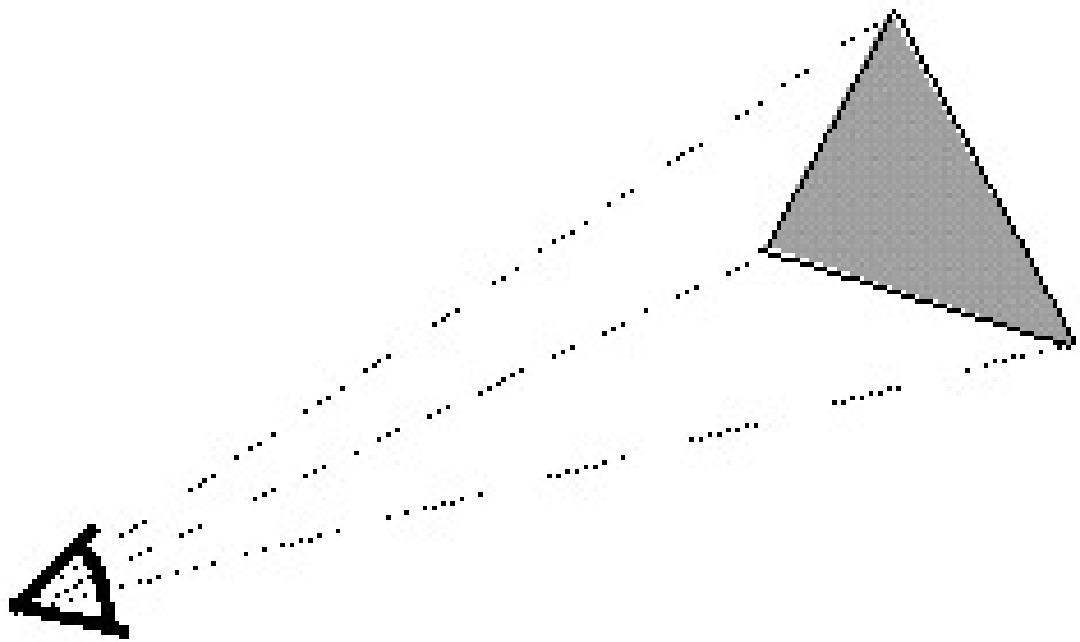
Quake's beam tree effectively partitioned the screen into 2-D regions.





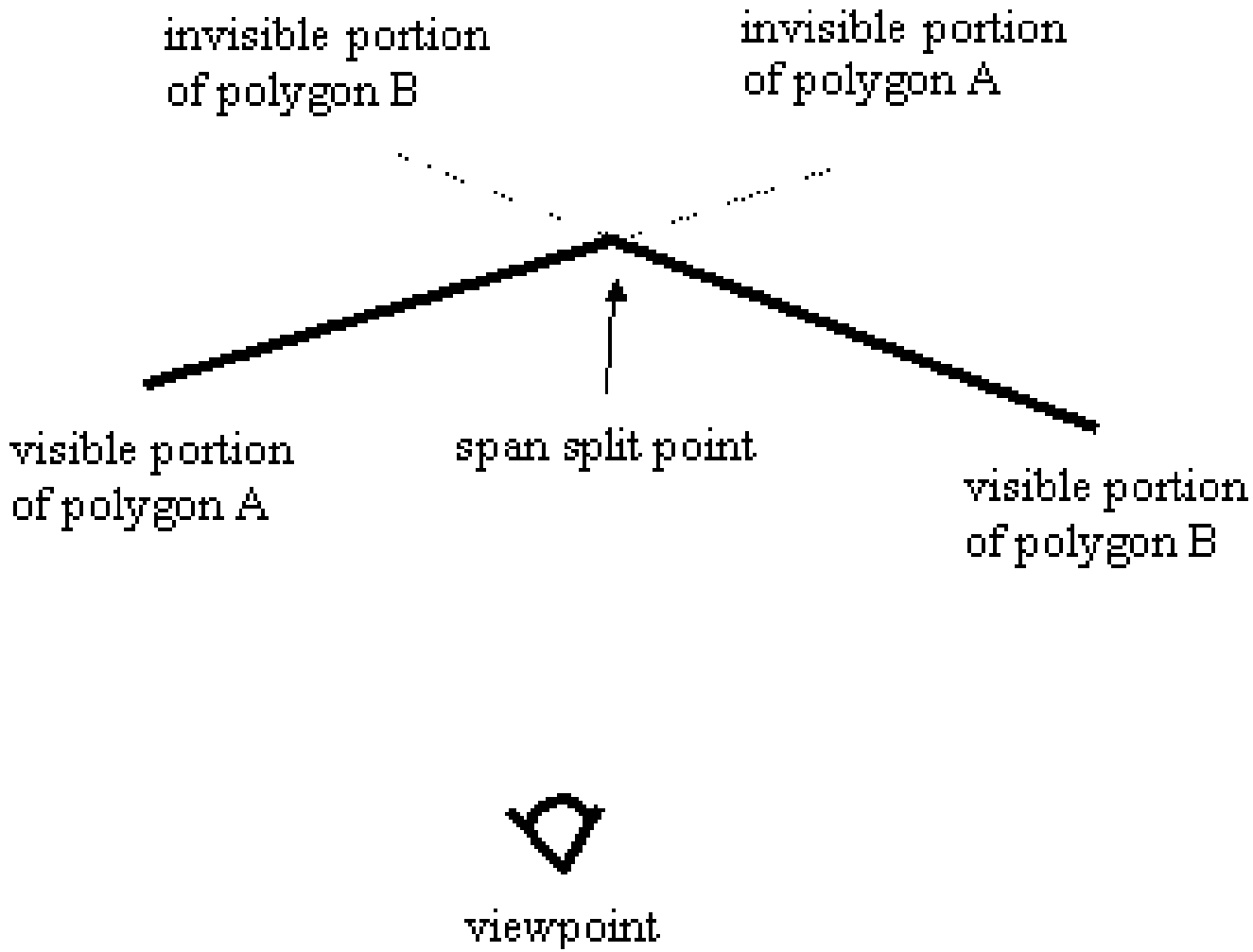
Design – Game Engines Case

Quake's beam tree was composed of 3-D wedges, or beams, projecting out from the viewpoint to polygon edges.



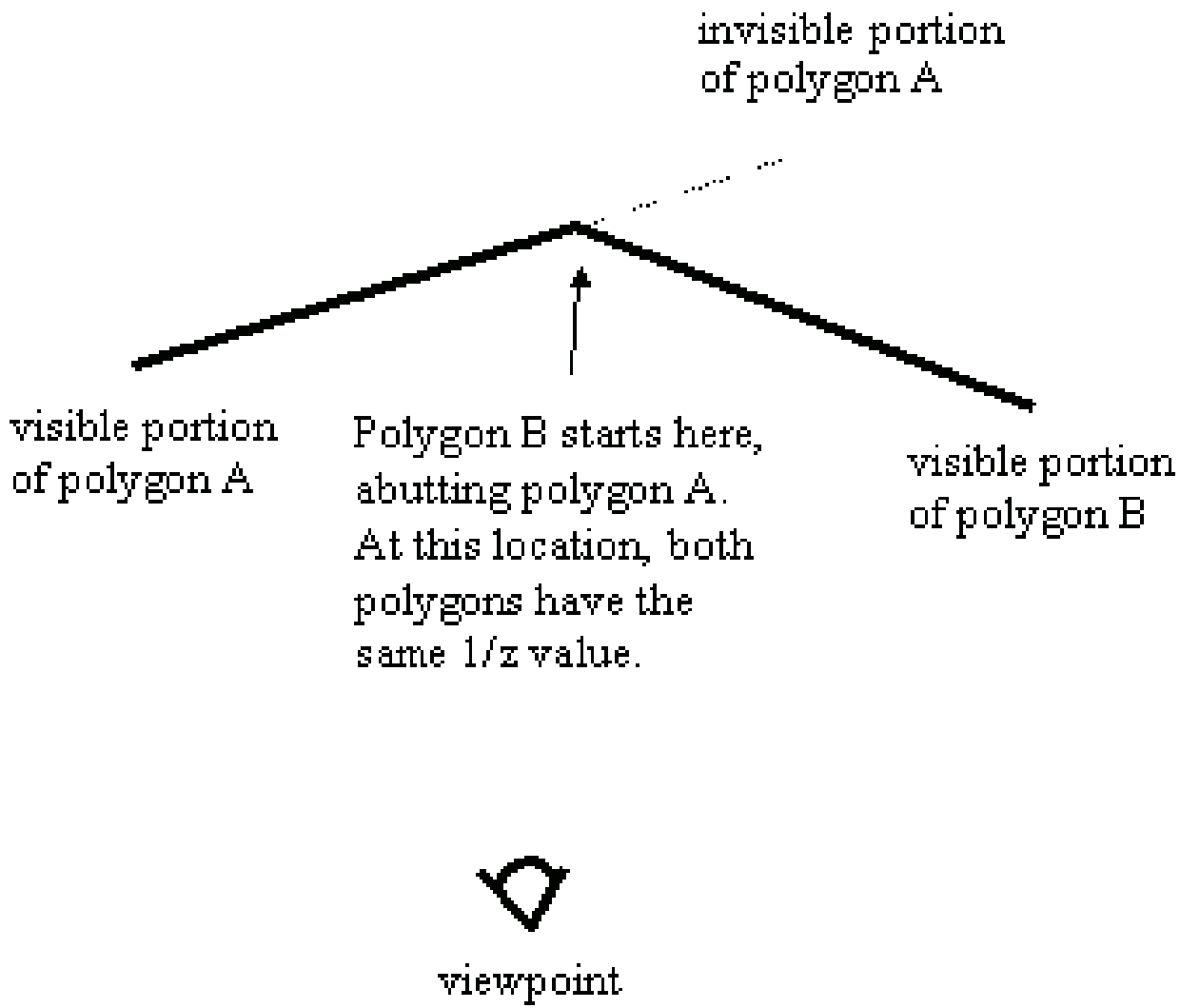
Design – Game Engines Case

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

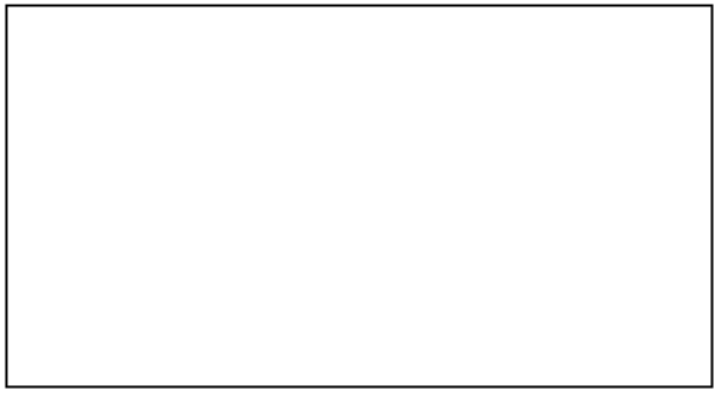


Design – Game Engines Case

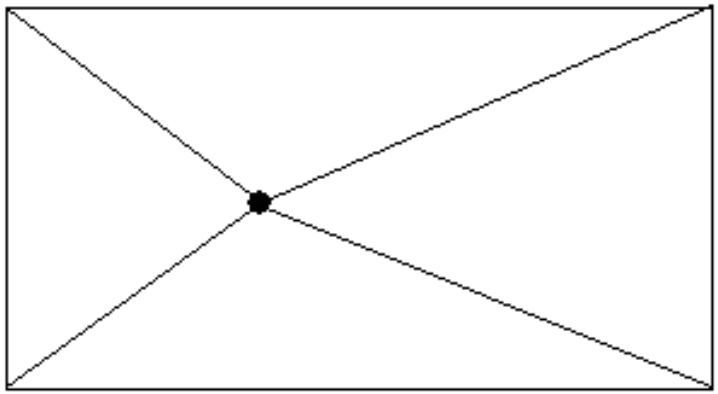
Abutting span sorting. Polygons A and B are viewed from above.



Design – Game Engines Case



Wall is a single polygon before adding a light vertex

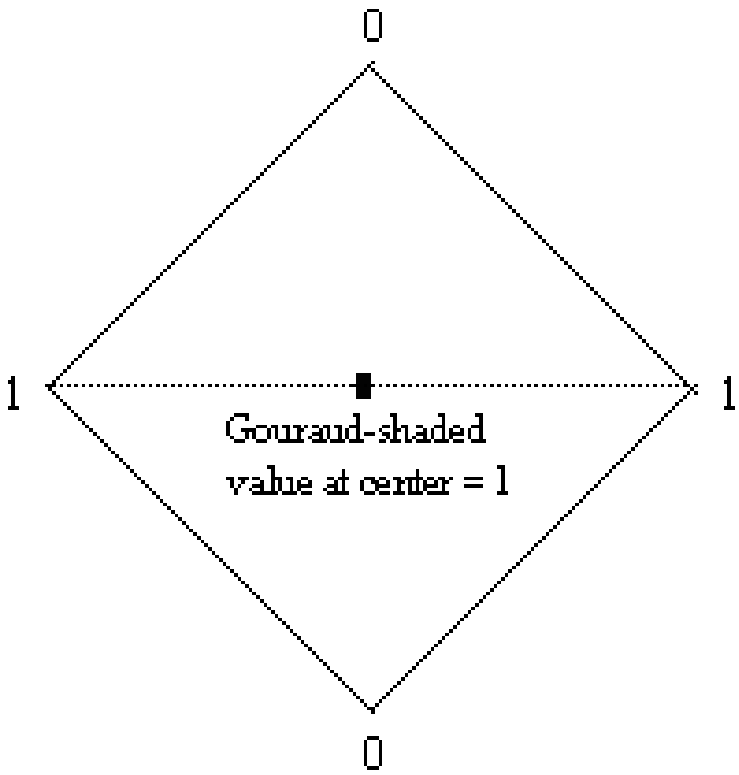


Wall becomes four polygons after adding a light vertex directly beneath a light

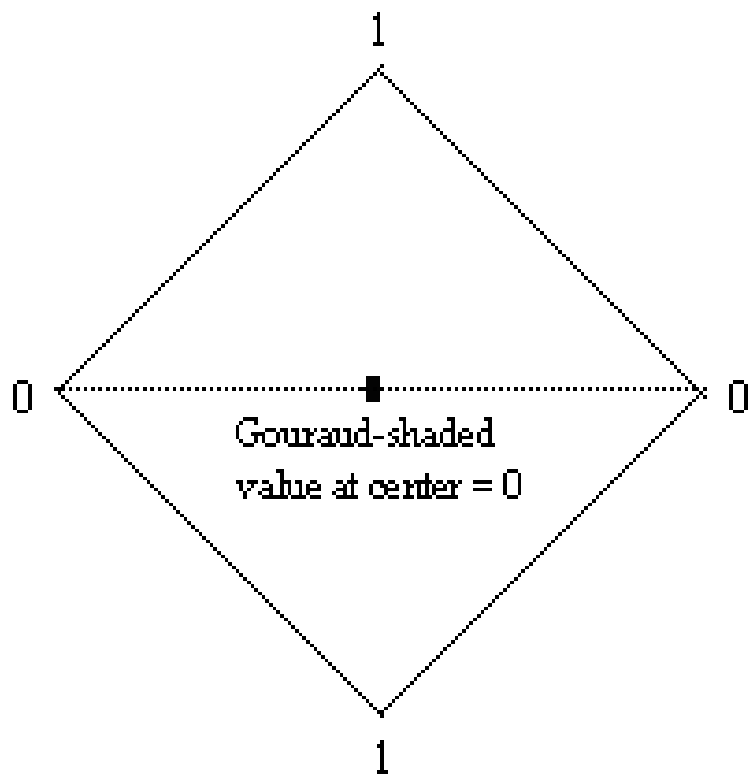
Quake's lighting model:
Adding an extra vertex directly beneath a light.

Design – Game Engines Case

Quake's lighting model: Gouraud shading varies with polygon screen orientation.

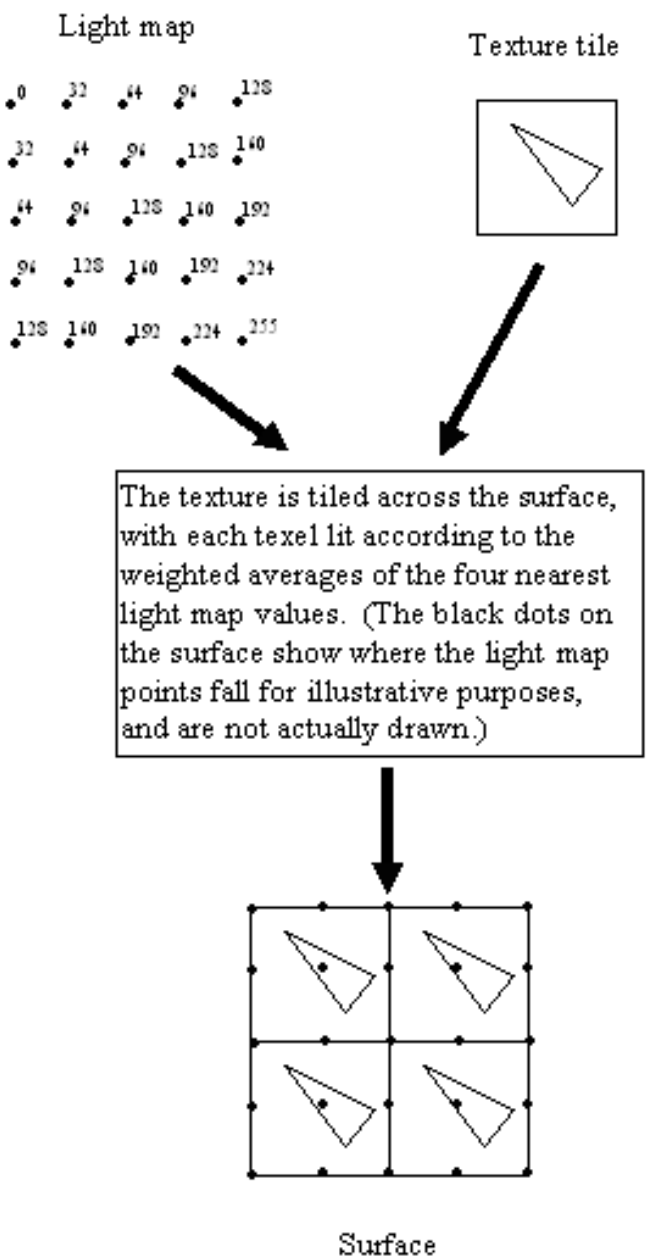


Rotated 0 degrees



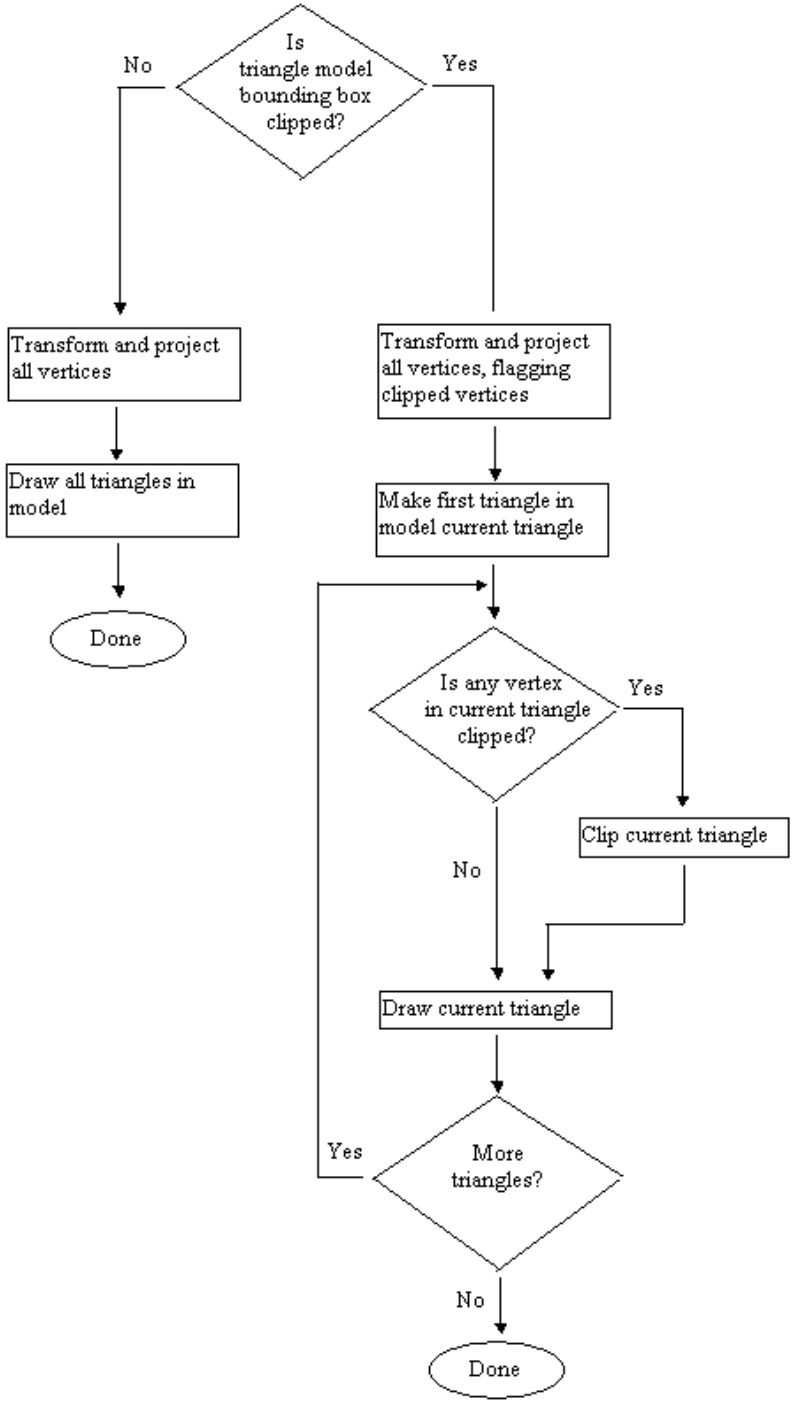
Rotated 90 degrees

Design – Game Engines Case



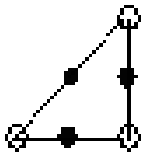
Quake's lighting model: A surface is built by tiling the texture and lighting the texels from the light map.

Design – Game Engines Case

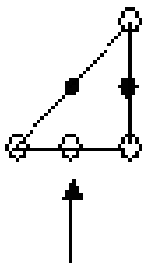


Quake's triangle-model drawing pipeline.

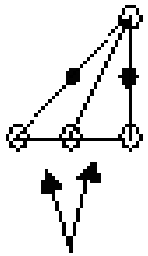
Design – Game Engines Case



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.

Design – Game Engines

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

Design – Game Engines

Trend

- 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

Trend

- 3D technology
  - Motion capture
- Virtual reality
  - Improve interaction devices
- Really good AI
  - Allow games to react to users dynamically
  - Build really good opponents

Trend

- 3D technology
  - Motion capture
- Virtual reality
  - Improve interaction devices
- Really good AI
  - Allow games to react to users dynamically
  - Build really good opponents

References

- David H. Eberly, *"3D Game Engine Design"*, Morgan Kaufmann Publishers, 2001.
- "*Designing a PC Game Engine* ", Lars Bishop, Dave Eberly, Turner Whitted, Mark Finch, and Michael Shantz, IEEE Computer Graphics and Applications, January, 1998
- Book: Andrew Rollings and Ernest Adams "Game Design"
- Book: "The illustrated history of electronic games"
- "Ramblings in Real Time" Michael Abrash 2001
- M Olano, J C Hart, W Heidrich and M McCool, *"Real-Time Shading"* A K Peters Publishers, 2002
- Tomas Akenine-Moller and Eric Haines, *"Real-Time Rendering"*, Second Edition, A K Peters Ltd, 2002
- D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Heubner, *"Level of Detail for 3D Graphics"*, Morgan Kaufmann Publishers, 2002.