

CS529

Fundamentals of Game Development

Lecture 2

Antoine Abi Chakra
Karim Fikani

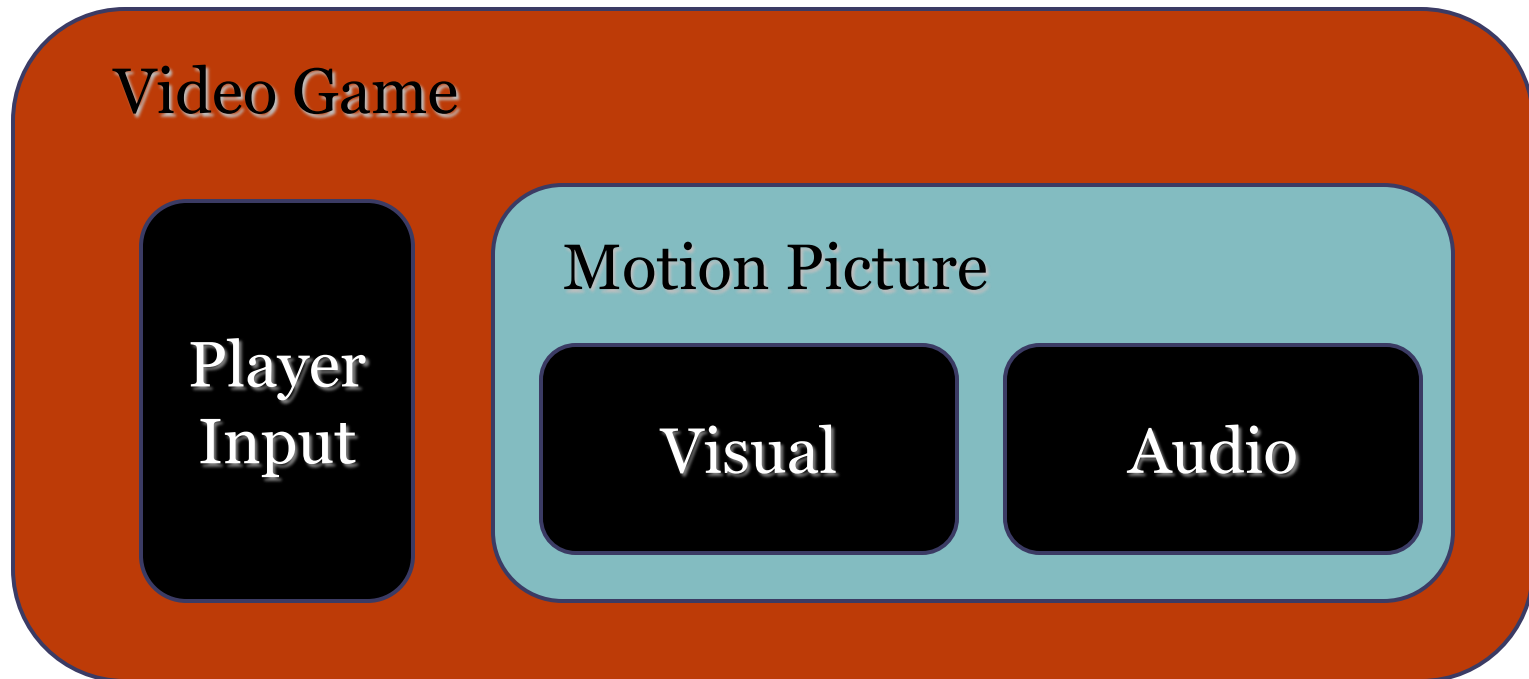
Questions?

- Resolution
- CRT
- Refresh rate & Frame Rate
- Vertical Sync
- LCD Monitors

Overview

- Game Engine
- Game Engine Components
 - System Components
 - Game Logic Components

What is a Game?



It's like a movie ... with interaction!!!

Simulation

VS

Games

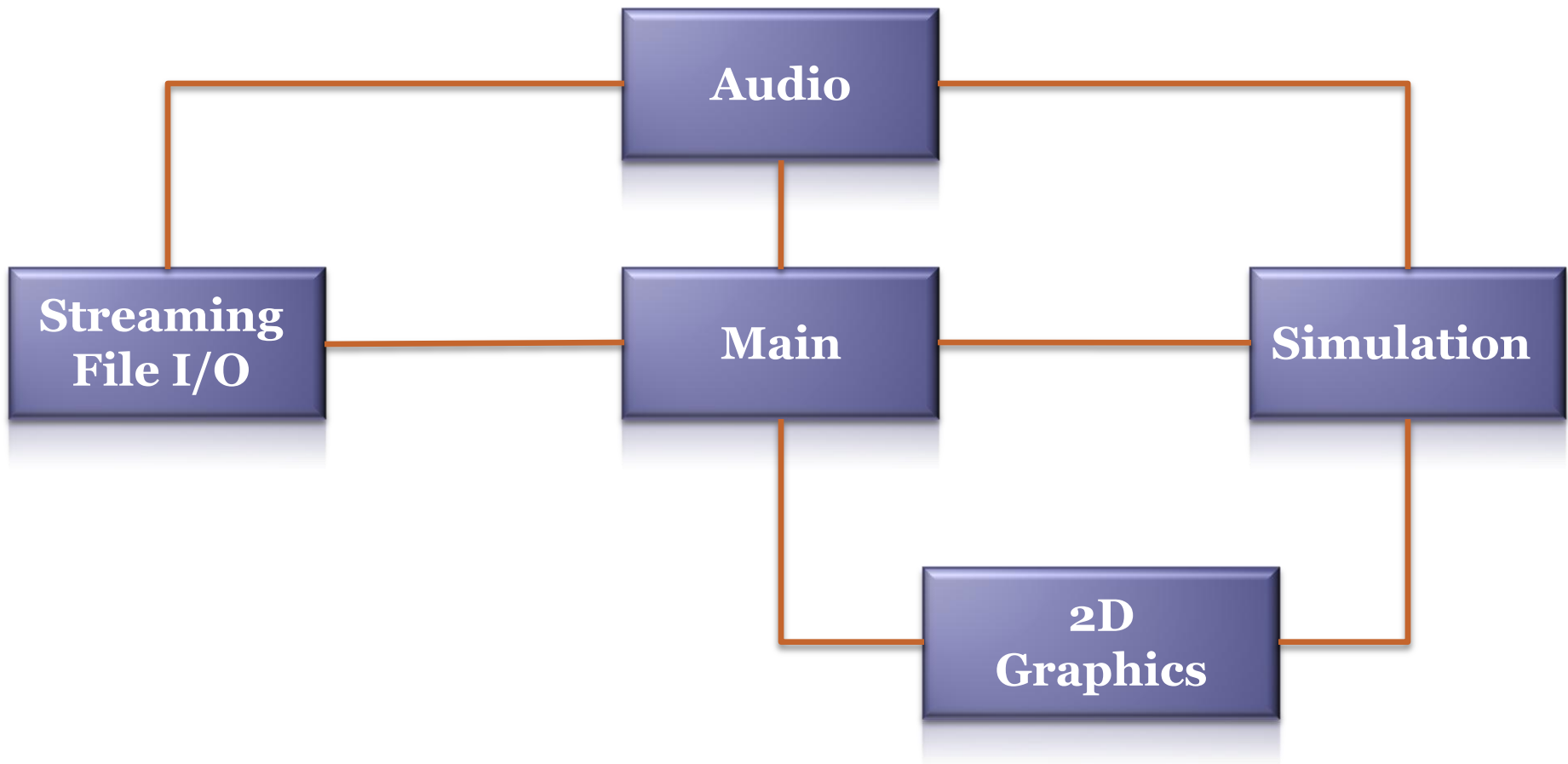
- Scientific representation of a real phenomenon
- For computational purposes

- Artistically simplified representation of a phenomenon
- For educational or entertainment purposes

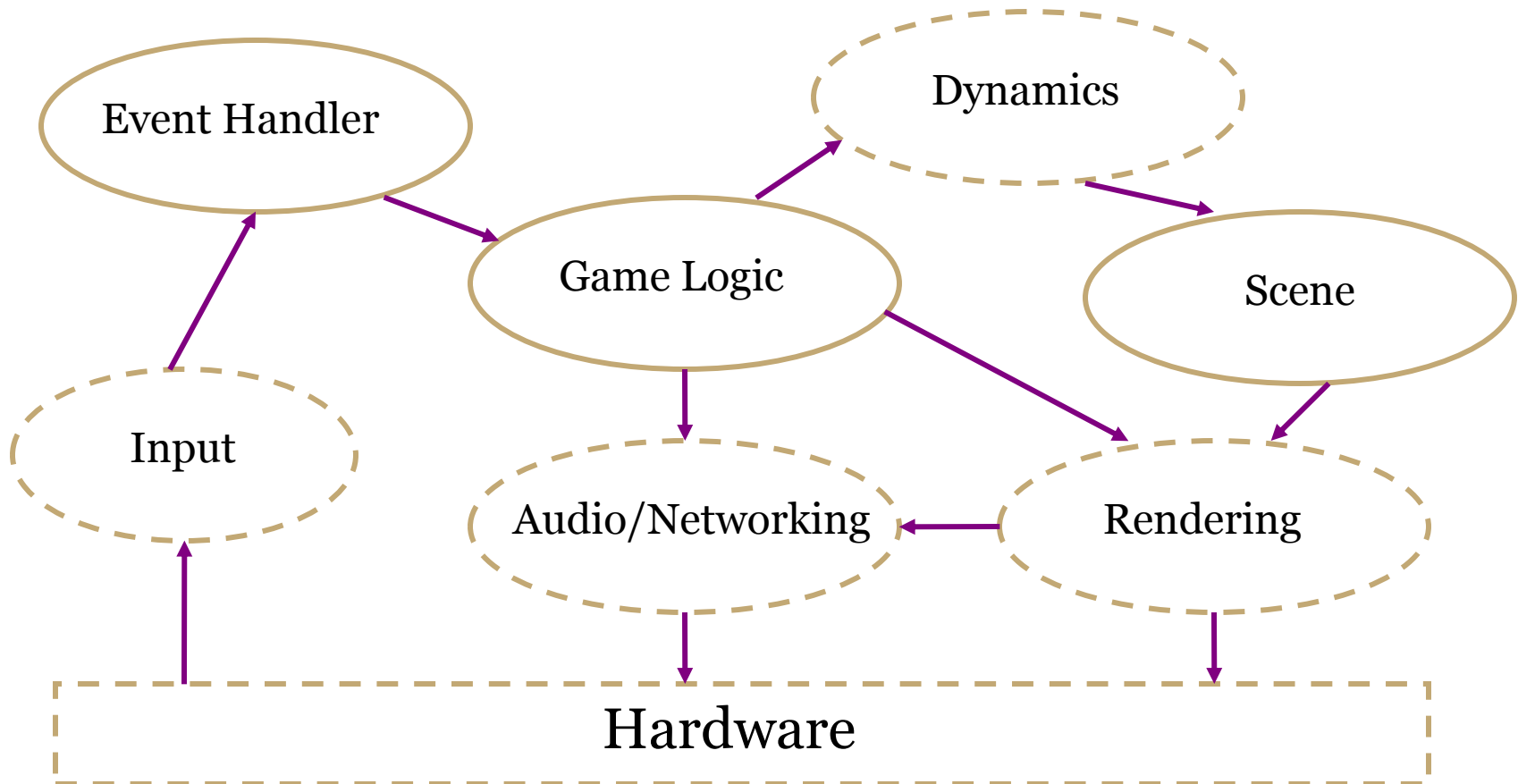
What is a Game Engine?

- It's the operating system of games
- Software system that enables game developers to build games without concerning themselves with system's internal structure

Structure of a Simple 2D Game Engine



Structure of a Game Engine



Game Engines

- A game engine includes all elements in schematic that have no effect on game content
 - Reusable elements indicated by dashed ovals
- In practice, most game engines are tuned to a particular content style.

Example:

An engine tuned for flight simulators may not be appropriate for games that take place in tunnels and dungeons.

Overview

- Game Engine
- Game Engine Components
 - System Components
 - Game Logic Components

Game Engine Components

- System component is responsible for isolating the game logic component from communicating with the hardware directly.
- The isolation makes it easier to port the game to another platform
- Subcomponents should belong entirely to either one of the main components.

System Components (1 / 2)

- The system's subcomponents are:
 - Frame Rate Controller
 - Memory Manager
 - File Manager
 - Graphics Manager
 - Audio Manager
 - Input Manager

System Components (2/2)

- Subcomponents dealing with the hardware should be initialized at the very beginning of the application.
- If any initialization fails, the application should quit.
- Upon exiting the application all the devices that were allocated should be released.

Frame Rate Controller

- Has two primary jobs:
 - To ensure a consistent frame rate for the current game state and decide when the frame buffer should be swapped
 - Tracks useful information:
 - Number of frames since the game started
 - Number of frames since the game state started
 - How much time the last frame used (in seconds)

Memory Manager

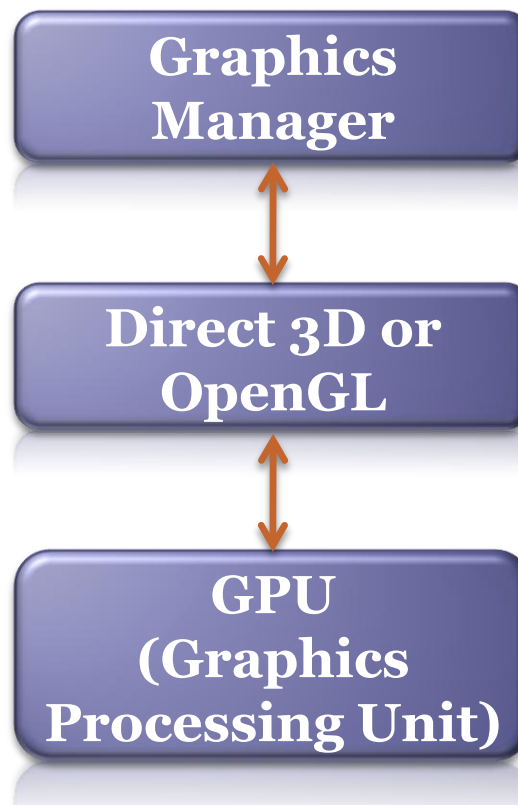
- Features that a memory manager should have:
 - Optimized memory allocation/deallocation for the game
 - Hierarchical memory heap
 - Easy way to track memory usage
 - Simpler way to track memory leaks
 - Runtime memory de-fragmentation

File Manager

- Features that a file manager should have:
 - Background file loading
 - Decompress data during load time
 - Automatic file sharing
- File managers run on a different thread

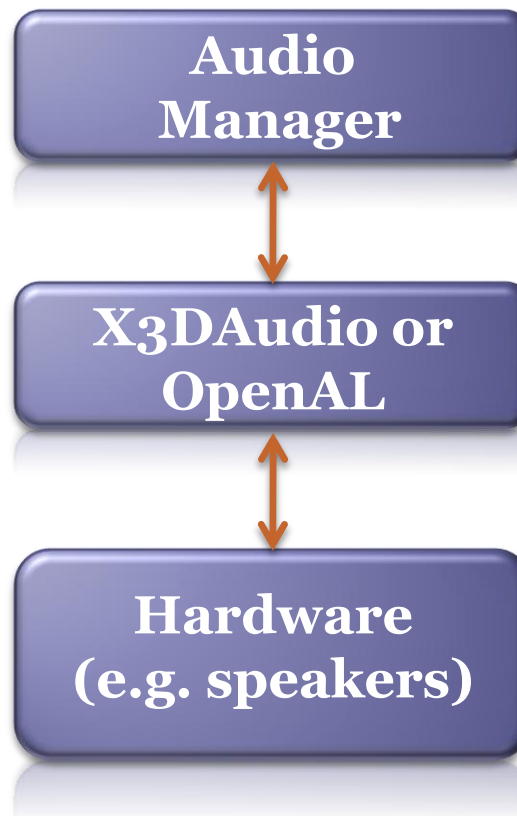
Graphics Manager

- A layer on top of the graphics hardware



Audio Manager

- Deals with the sound hardware



Input Manager

- The input manager keeps track, among other things:
 - Current pressed keys
 - Newly pressed keys
 - Newly released keys
 - A history of pressed keys

Game Logic Components

- The game logic's subcomponents are:
 - Game State Manager
 - Simulation Manager
 - Collision Detection Manager
 - Object Manager
 - Environment Manager
 - Camera System

Game State Manager (GSM) (1 / 2)

- A game is always in a state. A game could be in “Main Menu”, in “Level 1”, in “Loading screen”...
- The GSM is responsible for game state switching, the game loop and the frame rate controller.
- Each state is associated with a set of functions that manages that state's cycle.

Game State Manager (GSM) (2/2)

- The cycle functions are:
 - Load
 - Initialize
 - Update
 - Draw
 - Free
 - Unload

Simulation Manager

- It's the physics component of the engine
- Deals with the kinematics and dynamics of the game objects

Collision Detection Manager

- Takes care of the objects' impact and their collision updates
- Collision systems might be written from simple mathematical intersection functionalities to a complex partitioned collision sections
 - Quadtree
 - Octree
 - BSP Tree
 - Etc...

Object Manager

- Objects that are “Alive” are managed by the object manager that is responsible for:
 - Loading game objects and initialize them
 - Create and remove object instances
 - Updating objects
 - Sending objects to the graphics manager

Environment Manager

- The environment manager deals with static objects.
- Static objects are objects that the user does not interact with directly

Camera System

- Sounds very simple, yet, a bad camera could practically turn a perfectly good game into horrible game.
- There are four parameters commonly used to control the camera
 - CamPosition
 - CamTarget
 - CamUp
 - CamFOV