CSD1130
Game Implementation Techniques

Lecture 1

Overview

- RTIS: Real Time Interactive Simulation
  - Why Concurrent?
  - Why Interactive?
  - Why Real-Time?
  - Game Loop
    - Adding Interaction To The Game Loop
    - Game Flow
- Resolution
- CRT
  - Refresh rate & Frame Rate
  - Vertical Sync
- LCD Monitors

Why Concurrent?

- Many events are happening at the “same time”
  - Objects moving
  - Testing for input
  - Sound effects
  - Collision tests
  - AI
  - Updating HUD
  - etc..
- CPU can’t do all those simultaneously

Simulating Concurrent Events

- Several events need to be executed at the same time
  - Impossible
- Solution:
  - Execute all events sequentially
  - Draw the objects once all events are executed
  - Display the frame

Why Interactive?

- Players decide:
  - When & where to move the ship
  - When & where to shoot a bullet
  - Scores are updated while playing
  - AI reacts to players' actions

Why Real-Time?

- When objects move, their positions are calculated at run time
- Collision are determined at run time
- HUDs are updated at run time

Overview

- RTIS: Real Time Interactive Simulation
  - Why Concurrent?
  - Why Interactive?
  - Why Real-Time?
  - Game Loop
    - Adding Interaction To The Game Loop
    - Game Flow
- Resolution
- CRT
  - Refresh rate & Frame Rate
  - Vertical Sync
- LCD Monitors

Game Loop (1/3)

- A single frame is prepared by:
  - Executing all events sequentially
  - Drawing all the objects
  - Display the frame
- The above iteration is called a “Game Loop”

Game Loop (2 / 3)

- Game loop duration greatly affects the illusion of concurrent events
- If the game iteration is relatively long (Let's say 0.1s)
  - Simulation will feel slow
  - Reactions to events happen only 10 times a second
- If the game iteration is short (Let's say 0.016s)
  - Simulation will feel smooth
  - Reactions to events happen 60 times a second

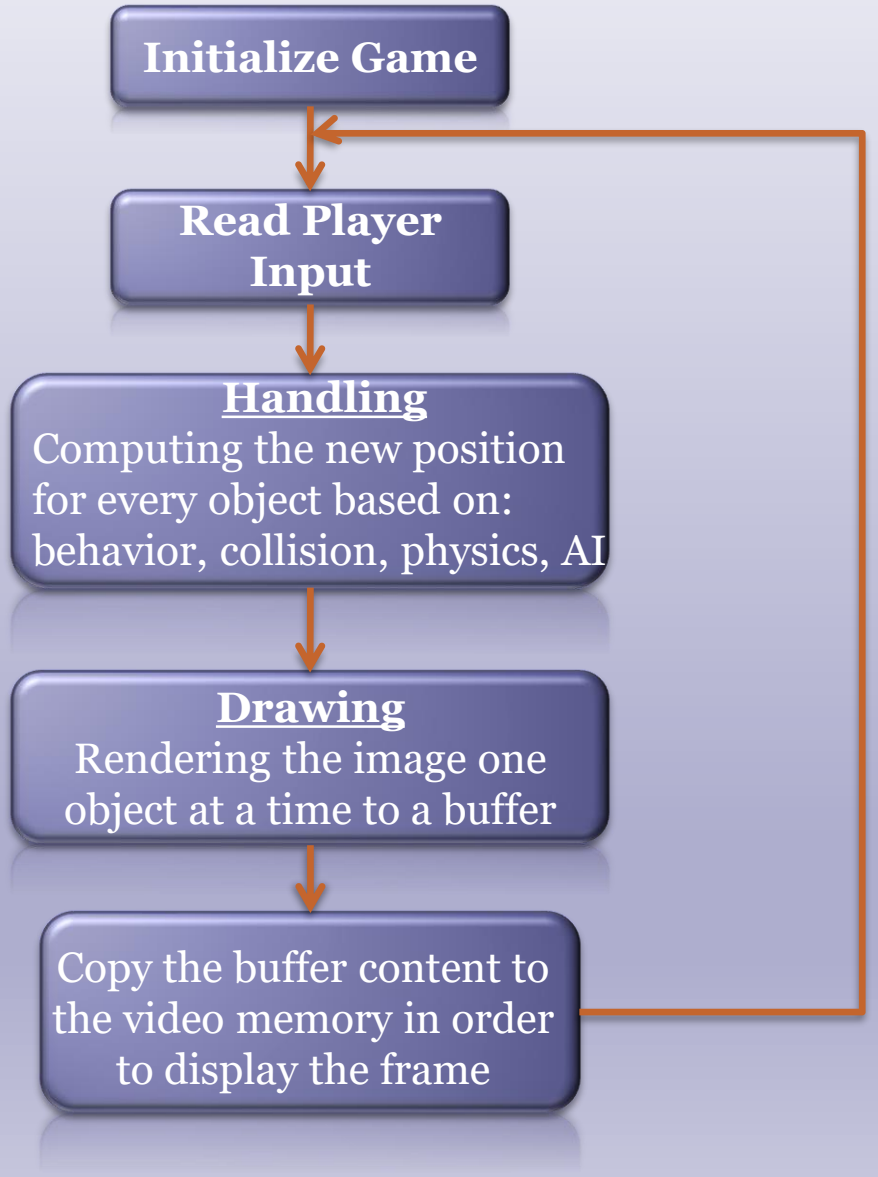
Game Loop (3/3)

- The duration of a frame (One game iteration) is called the “Frame Time”
- Game’s speed is measured in “Frames per Second”
- Example:
  - If a game is running at 60 FPS, its game loop duration is 1/60 seconds, or 0.016 seconds

Adding Interaction To The Game Loop

- Register the input at the beginning of the game loop
- All game components inquire from that input state
  - Guarantees input uniformity throughout a single game loop

Game Flow


Initialize Game
Read Player Input
Handling
Computing the new position for every object based on: behavior, collision, physics, AI
Drawing
Rendering the image one object at a time to a buffer
Copy the buffer content to the video memory in order to display the frame

Overview

- RTIS: Real Time Interactive Simulation
  - Why Concurrent?
  - Why Interactive?
  - Why Real-Time?
  - Game Loop
    - Adding Interaction To The Game Loop
    - Game Flow
- Resolution
- CRT
  - Refresh rate & Frame Rate
  - Vertical Sync
- LCD Monitors

Resolution

- What is a pixel?
  - Smallest element of a picture
  - Derived from “Picture Element”
  - Tiny squares
- Resolution is defined by the number of pixels
  - Usually written as Width*Height. Ex: 640*480
  - The higher the resolution, the more memory is needed to store the picture’s data

RGB Concept

- Color are composed out of 3 colors: Red, Green & Blue
  - Red = 0, Green = 0 & Blue = 0: Black
  - Full red, Full Green & Full Blue: White
- Color bit mode
  - 8 bit (2^8 = 256 colors)
  - 16 bit (2^{16} = 65,536 colors)
  - 24 bit (2^{24} = 16,777,216 colors)
  - 32 bit (2^{32} = 4,294,967,295 colors)

Overview

- RTIS: Real Time Interactive Simulation
  - Why Concurrent?
  - Why Interactive?
  - Why Real-Time?
  - Game Loop
    - Adding Interaction To The Game Loop
    - Game Flow
- Resolution
- CRT
  - Refresh rate & Frame Rate
  - Vertical Sync
- LCD Monitors

CRT Concept (1/3)

- Cathode Ray Tube
- Electron gun shoots electrons at phosphor targets
  - Light is emitted for a *short* period of time
- Direction of the beam is controlled by deflection plates, or by a magnetic field
- Color CRTs have 3 phosphors: Red, Green & Blue
- Output from the computer is converted by a digital-to-analog converter

CRT Concept (2/3)

- Electron gun shoots electrons at all the phosphor targets, *sequentially*
- Timing:
  - H-Blank is time needed to raster the next row
  - V-Blank is time needed to go from the last pixel of the last line to the first pixel of the first line
- Problem: Phosphors are lit for a very short amount of time

CRT Concept (3/3)

- Solution: Image must be refreshed at least 50 times a second
  - Called: Refresh rate of the monitor. Measured in Hz
- Interlaced:
  - Odd numbered lines are refreshed during one frame
  - Even numbered lines are refreshed during the next one, and so on…

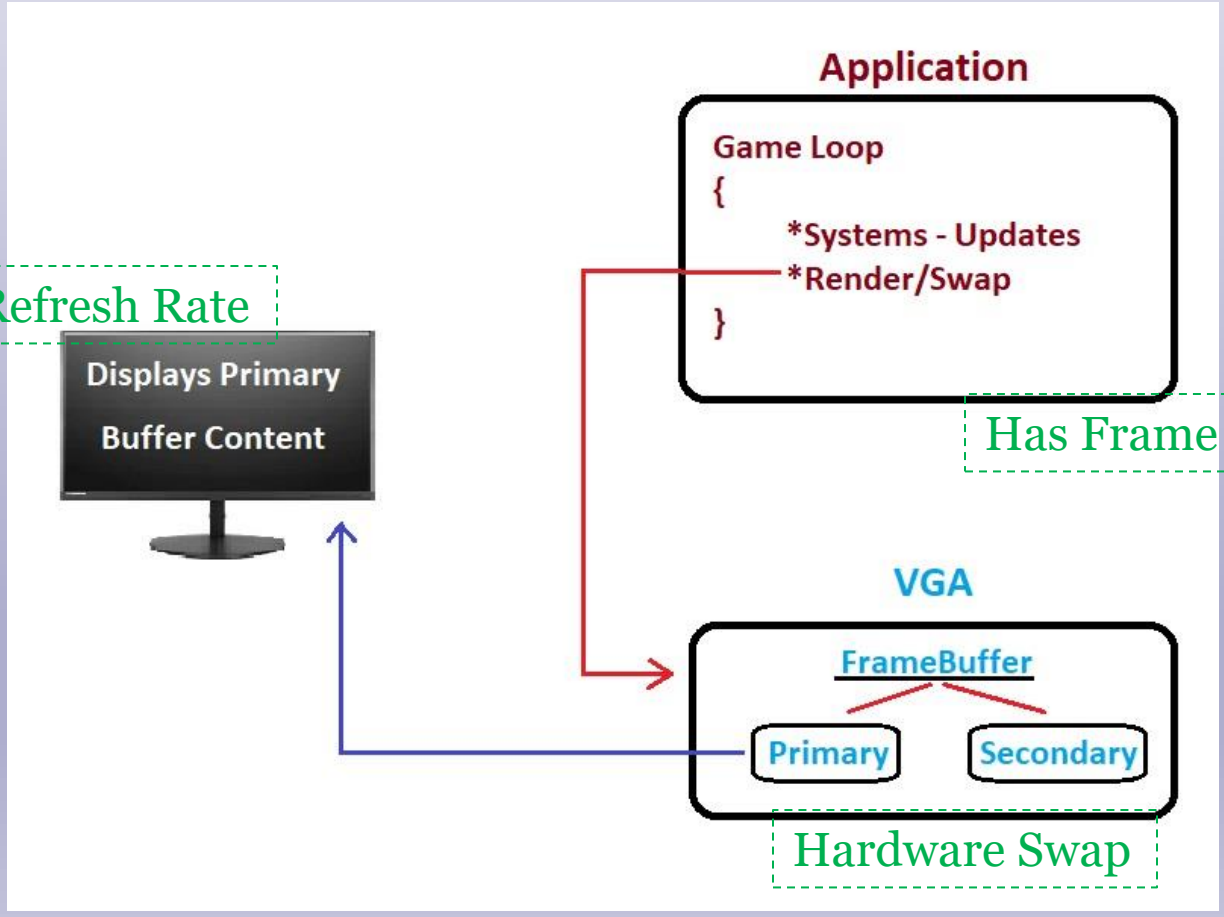
Refresh Rate

- Number of times per second the display is refreshed
- Different than the frame rate of an application
- Example:
  - A movie playing on a projector

Refresh Rate & Frame Rate

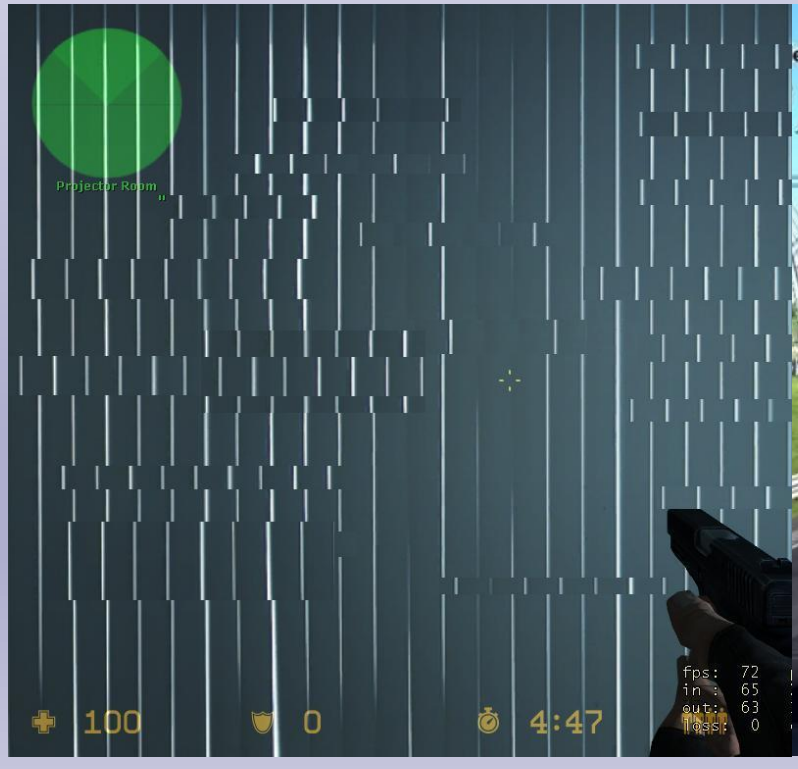
- The monitor and the video card do **not** have to be in sync
- Each time the monitor needs to refresh itself, it takes the content of the primary frame buffer and displays it
  - Frame buffer: Array of colors found on the video card

Refresh Rate & Frame Rate



Out of Sync?

- If the application's FPS is different than the monitor's refresh rate, a tearing effect will take place



Vertical Sync (1/4)

- Called vSync
- Sync the graphics card to the monitor's refresh rate
  - Done by making the graphics card wait for the V-Blank before changing the content of the frame buffer
  - Guarantees that the monitor will never display parts of different images in a single refresh
- Works perfectly when the game's frame rate is higher than the monitor's refresh rate
  - What if it's not the case?

Vertical Sync(example) (2/4)

- Application running at 50 FPS
  - Frame Time = 1/50 seconds = 0.02 seconds
- Monitor running at 60 Hz
  - Refresh Time = 1/60 seconds = 0.016 seconds

Vertical Sync(example) (3/4)

Time	Frame Buffer	Monitor
0.0	Blank	Nothing yet
0.016 (monitor refresh time)	Blank	Blank
0.02 (app frame rate)	Contains frame 1	Blank
0.032 (monitor refresh time)	Contains frame 1	Displays frame 1 (Now the game can start working on frame 2)
0.048 (monitor refresh time)	Contains frame 1	Display frame 1
0.052 (app frame rate)	Contains frame 2	Still displaying frame 1
0.064 (monitor refresh time)	Contains frame 2	Displays frame 2 (Now the game can start working on frame 3)
0.08 (monitor refresh time)	Contains frame 2	Displays frame 2
0.084 (app frame rate)	Contains frame 3	Still displaying frame 2
0.096 (monitor refresh time)	Contains frame 3	Displays frame 3 (Now the game can start working on frame 4)

Vertical Sync(example) (4/4)

- Game skips the monitor's V-Blank, therefore it must wait for the next one
- Monitor displays each image twice
- Game's FPS drops to 30
  - Half the monitor's refresh rate
  - Third the monitor's refresh rate if it's already below 30
  - And so on..

Overview

- RTIS: Real Time Interactive Simulation
  - Why Concurrent?
  - Why Interactive?
  - Why Real-Time?
  - Game Loop
    - Adding Interaction To The Game Loop
    - Game Flow
- Resolution
- CRT
  - Refresh rate & Frame Rate
  - Vertical Sync
- LCD Monitors

LCD Monitors (1/2)

- No electron gun
- Array of liquid crystal between 2 layers of polarized glass
- Light is sent towards all the pixels *simultaneously*
- A crystal's orientation is changed to alter the light passing through it
  - Time needed is called Response Time
- V-Blank is non-existent on LCDs
  - What about vertical sync?

LCD Monitors (2/2)

- LCD monitors emulate CRTs' refresh rate
  - Sending fake V-Blanks

Adaptive Sync

- For further readings, take a look at G-Sync and FreeSync both from Nvidia and AMD
- Both under Adaptive Sync technology