CSD1130
Game Implementation Techniques

Lecture 4

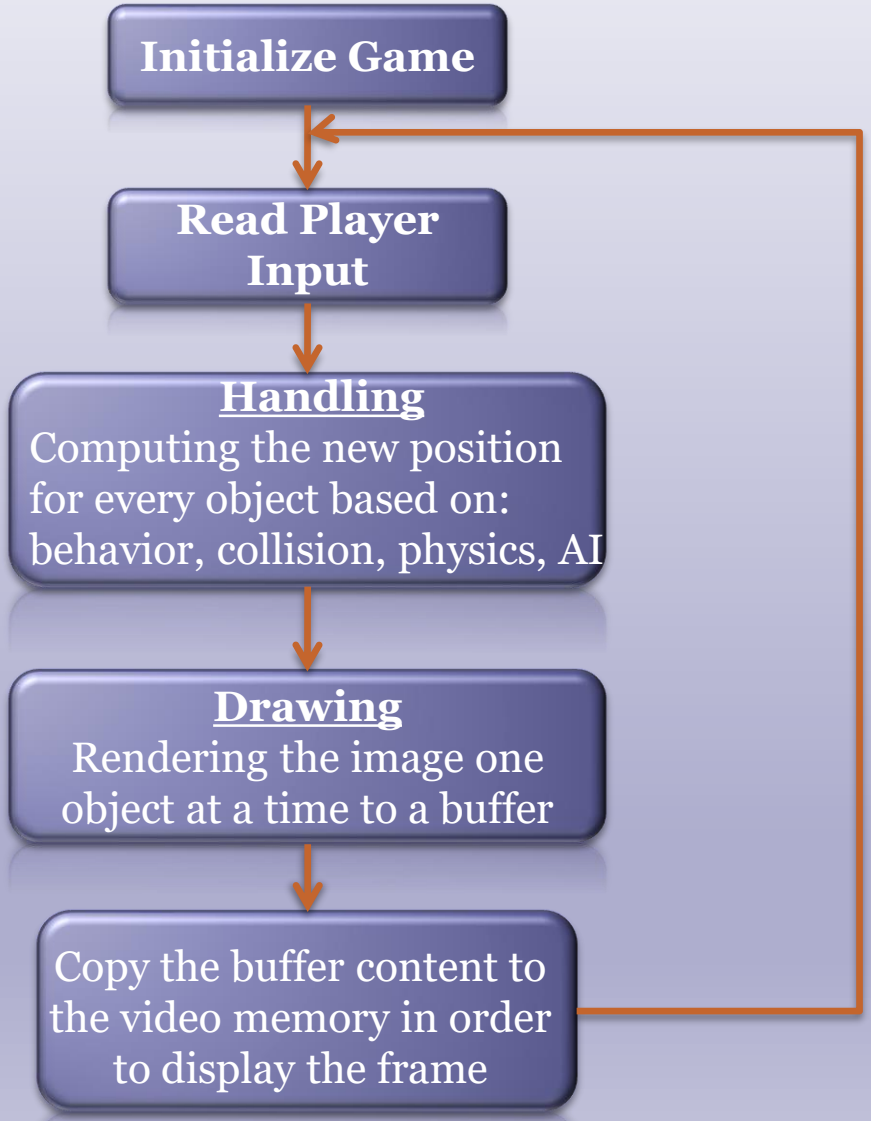
Questions?

- Game State Manager
- Function Pointers

Overview

- Frame Rate Controller

What is a Game Loop?



Simple Game Loop

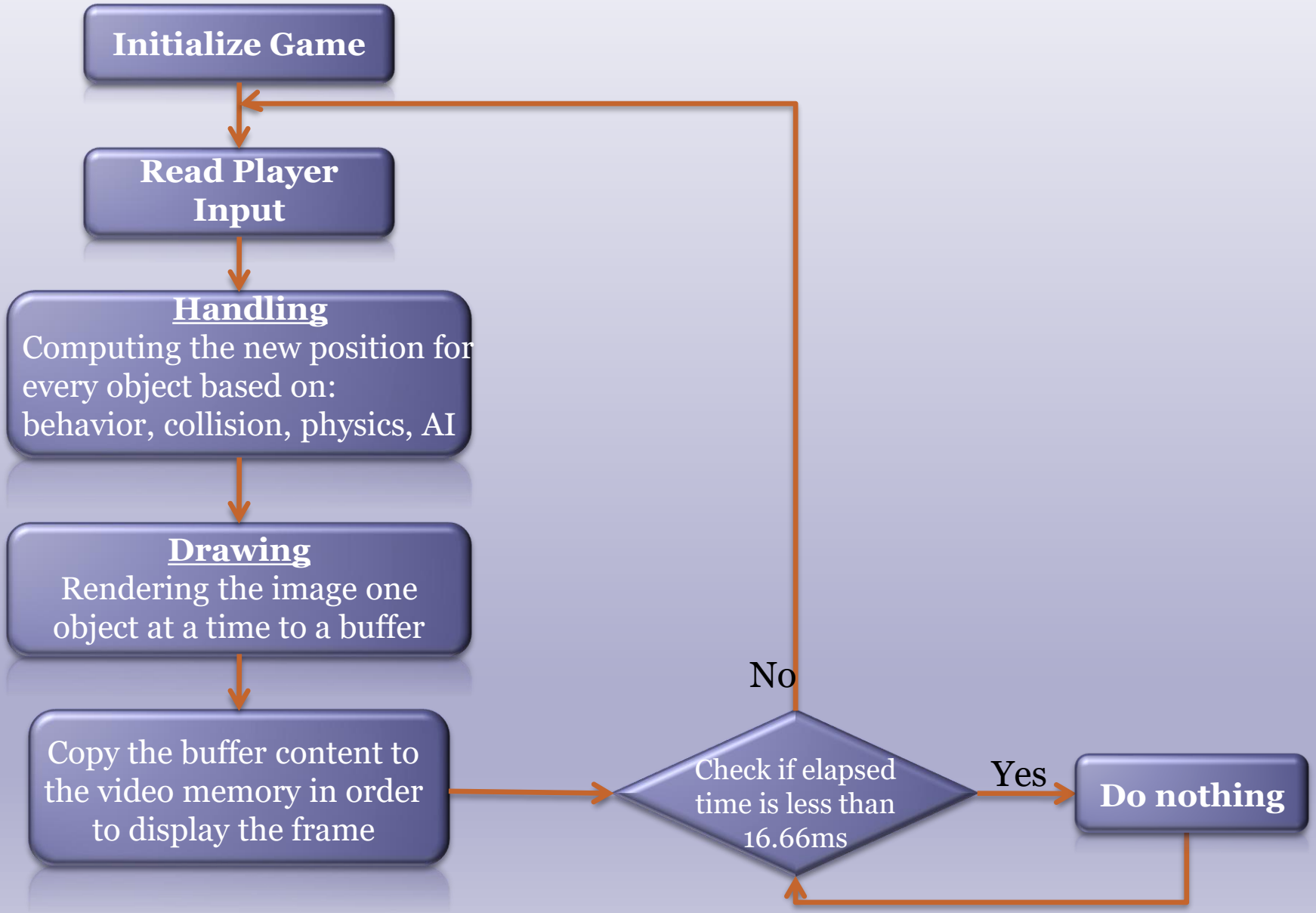
```
Initialize_Game_Objects();

while(!quit)
{
      Read_Input();
      Update_Game_Objects();
      Draw_Game_Objects();
}
```

- Varying frame rate will lead to uneven animation objects

Game Loop - Frame Based (1/2)

- Cap maximum frame rate of simulation - say to 60fps
- However, frame rate may still fall below capped maximum causing uneven animation


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
Check if elapsed time is less than 16.66ms
No
Yes
Do nothing

Game Loop – Frame Based (2/2)

```
double currTime = time( ); // measure time at start of frame

Initialize_Game_Objects( );

while (!quit)
{
   currTime = time(); // measure time at end of previous frame or time at start
                              // of current frame
   Update_Game_Objects( );
   Draw_Game_Objects( );

   do
   {
         newTime = time();
   }
   while ((newTime - currTime) < FRAME_TIME_MIN) // FRAME_TIME_MIN = 1.0/60.0
}
```

Game Loop – Time Based (1/2)

- Objects are no longer updated based on a pre-determined time between successive frames
- Instead, time interval to complete current frame is used in kinematics calculations to determine objects' displacements
  - Computing time interval to complete current frame is non-trivial problem
  - Instead, good compromise is to use time interval of previous frame

Game Loop - Time Based (2/2)

```
double t = 0.0f; // game time (in seconds)
double currTime = time( ); // measure time at start of frame
Initialize_Game_Objects( t, 0.0f );

while (!quit)
{
   double newTime = time( ); // measure time at end of previous frame or time at start
                             // of current frame
   double dt = newTime - currTime; // time interval for previous frame (in seconds)
   currTime = newTime; // time at start of current frame
   Update_Game_Objects( t, dt );
   Draw_Game_Objects( );


   t += dt; // update game time with time interval of previous frame
}
```

Game Loop - Combined

```
double t = 0.0f; // game time (in seconds)
double currTime = time( ); // measure time at start of frame
Initialize_Game_Objects( t, 0.0f );

while (!quit)
{
   double newTime = time( ); // measure time at end of previous frame or time at start
                             // of current frame
   double dt = newTime - currTime; // time interval for previous frame (in seconds)
   currTime = newTime; // time at start of current frame
   Update_Game_Objects( t, dt );
   Draw_Game_Objects( );
   do
   {
        newTime = time();
        dt = newTime - currTime;
   }
   while (dt < FRAME_TIME_MIN) // FRAME_TIME_MIN = 1.0/60.0
   t += dt; // update game time with time interval of previous frame
}
```