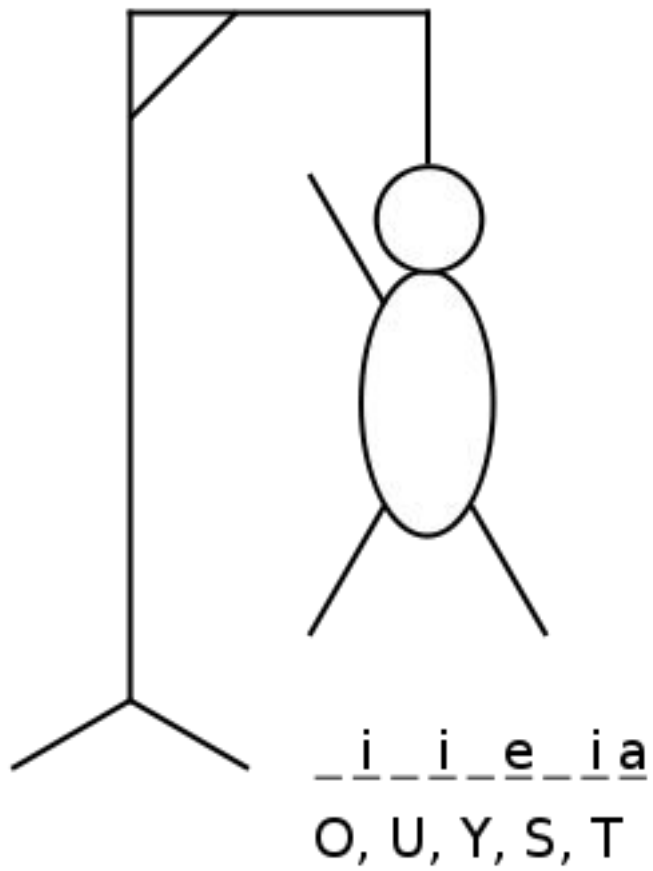


# Game Loops

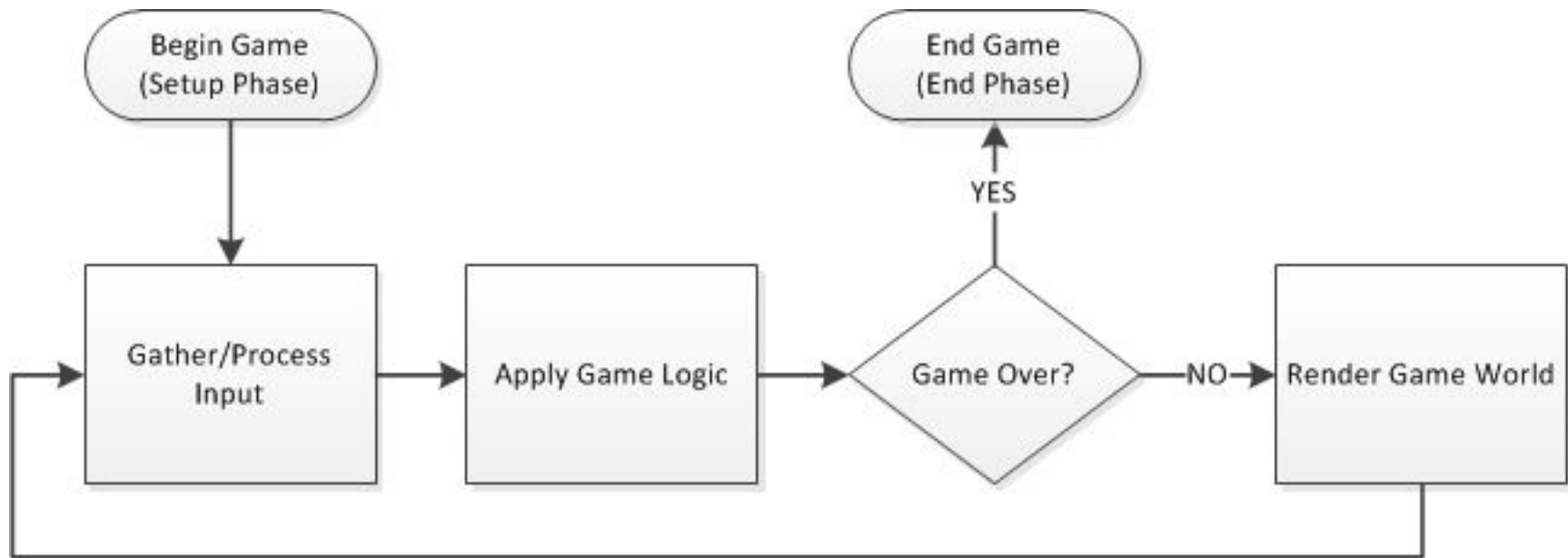
CIS 580 - Fundamentals of Game  
Programming



Hangman



## Game Phases



Game Loop

# Turn-based vs. Real-Time



Turn-Based

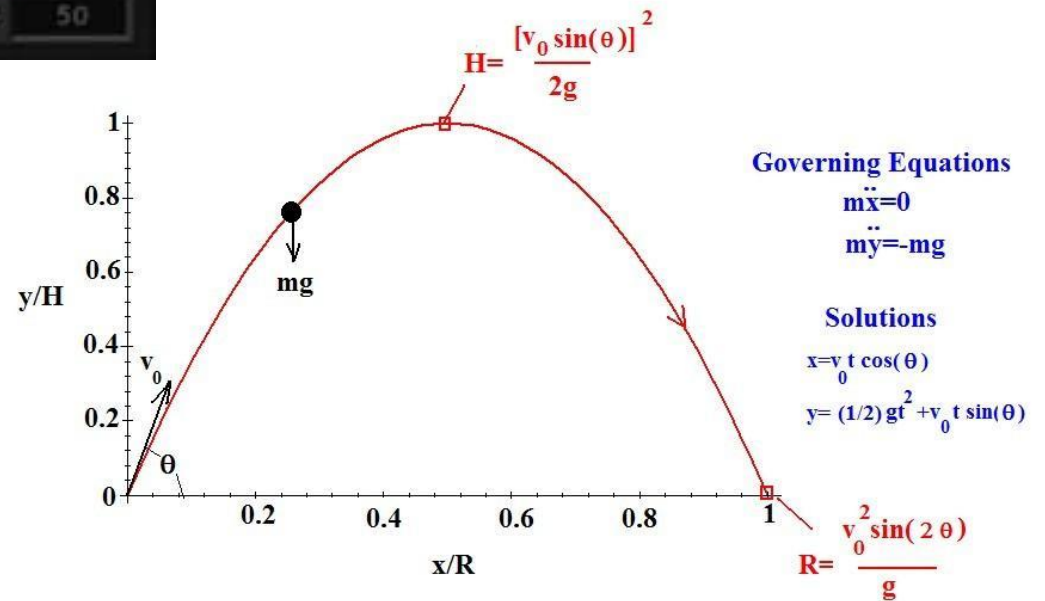


# Fixed-Timestep vs. Variable Timestep

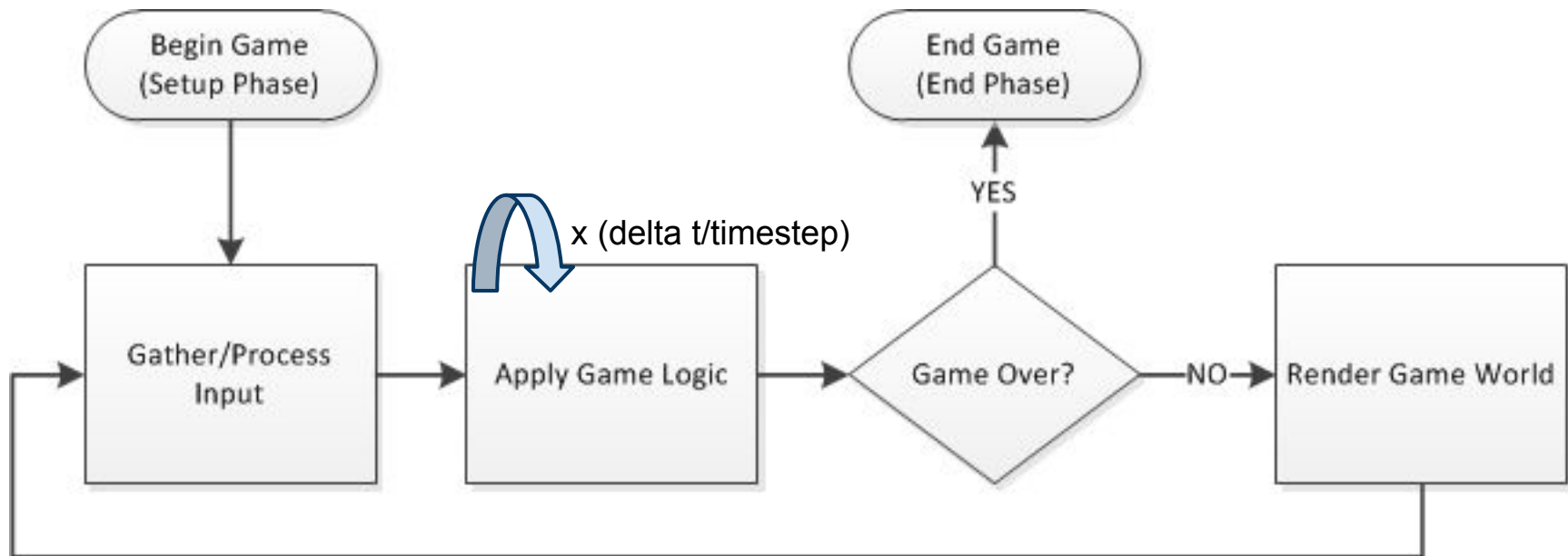




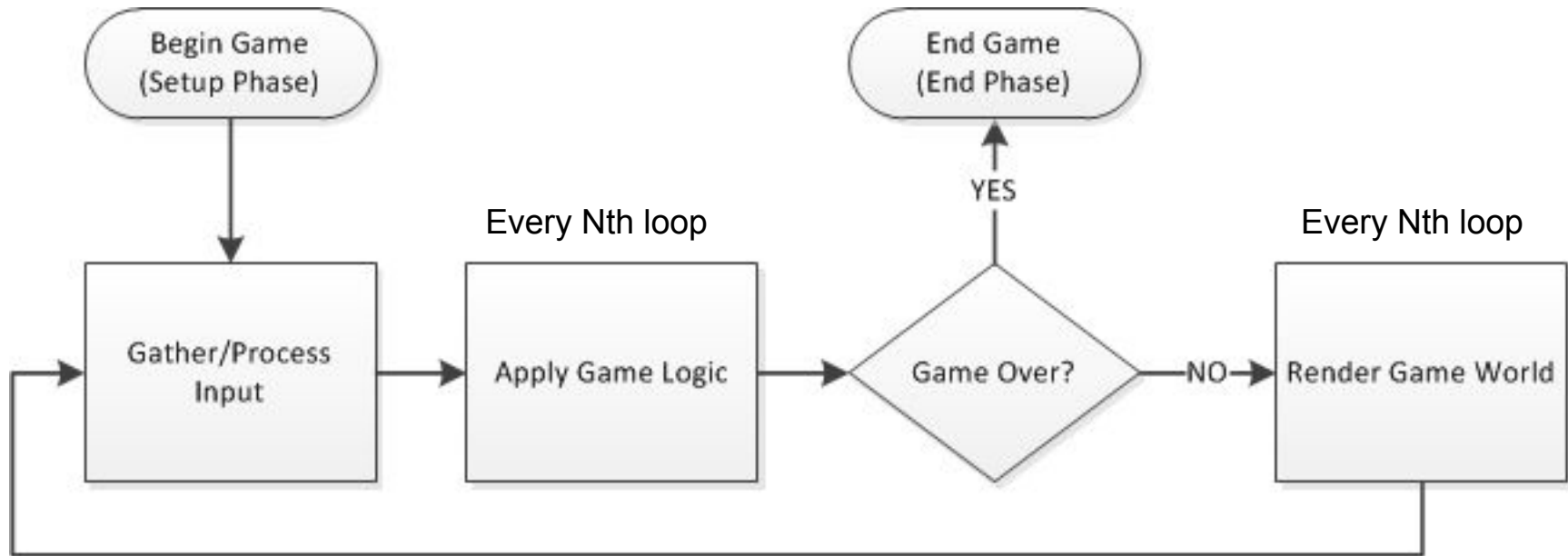
PARABOLIC TRAJECTORY OF A PROJECTILE



# Example - Ballistic Motion



Hybrid Game Loop



Variable update/render Game Loop

# Coding Game Loops

```
function loop() {  
    setTimeout( loop, 16.666); // Asynchronous  
    processInput();  
    update();  
    render();  
}
```

```
// 0.016666s = 1/60s  
setTimeout( loop, 16.666);
```

setTimeout game loop

```
function loop() {  
    processInput();  
    update();  
    render();  
}
```

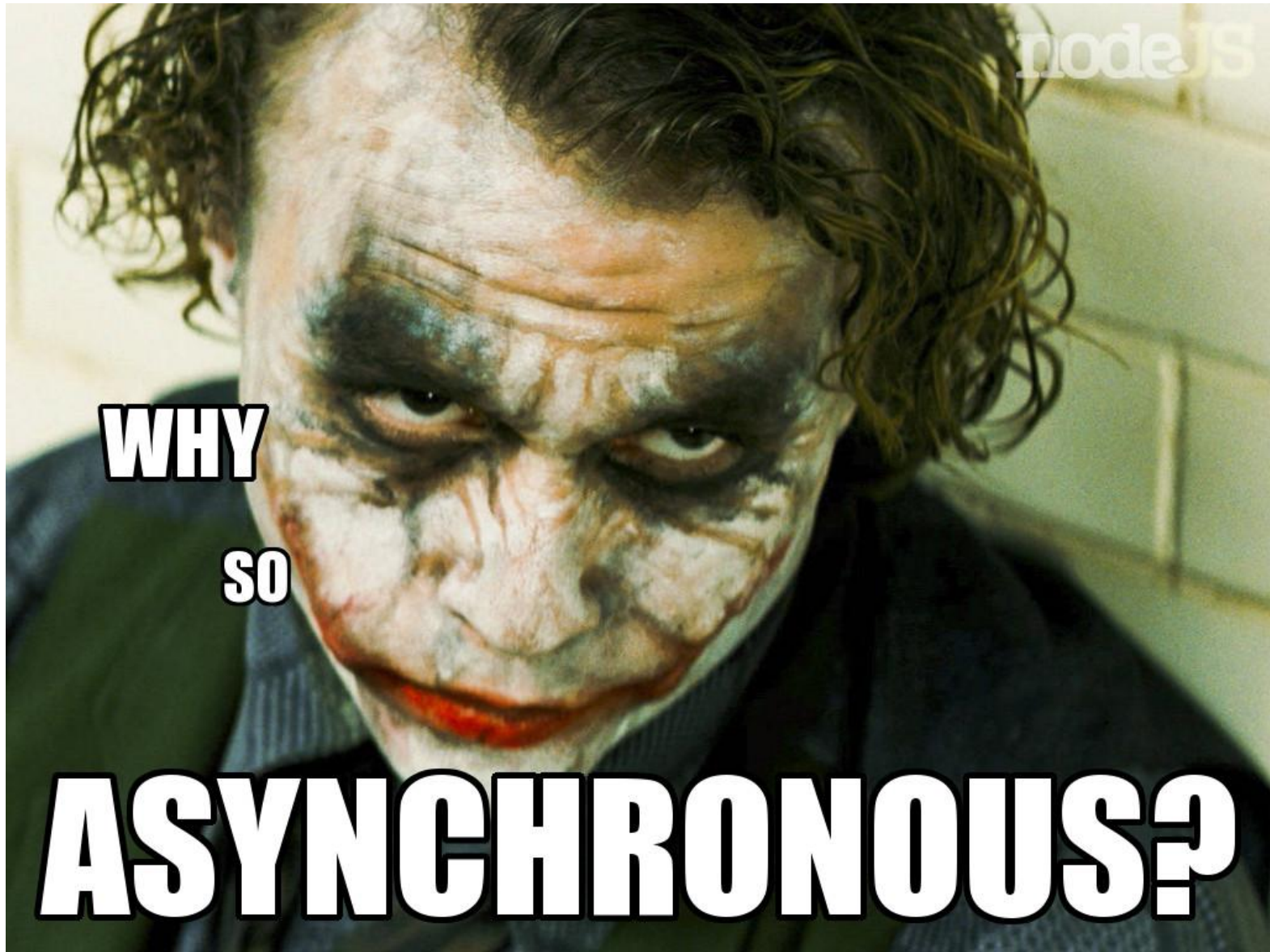
```
// 0.016666s = 1/60s
```

```
var loopHandle = setInterval( loop, 16.666);
```

setInterval game loop

```
function loop() {  
    processInput();  
    update();  
    render();  
    loop();  
}
```

recursive game loop



**WHY**

**SO**

**ASYNCHRONOUS?**



```
function loop(timestamp) {  
    processInput();  
    update();  
    render();  
    requestAnimationFrame(loop);  
}
```

```
requestAnimationFrame(loop);
```

Simple request animation frame game loop

```
var start = null;

function loop(timestamp) {
    var progress;
    if (start === null) start = timestamp;
    progress = timestamp - start;
    processInput();
    update(progress);
    render(progress);
    requestAnimationFrame(loop);
}

requestAnimationFrame(loop);
```

“Variable”-rate setTimeout game loop

```
var start = null;
var timestep = 16.6666; // 0.016666s = 1/60s
function loop(timestamp) {
    var progress;
    if (start === null) start = timestamp;
    progress = timestamp - start;
    processInput();
    while(progress > timestep) {
        update(progress);
        progress -= timestep;
        start += timestep;
    }
    render(progress);
    requestAnimationFrame(loop);
}
requestAnimationFrame(loop);
```

Fixed Timestep setTimeout game loop

```
1 // http://paulirish.com/2011/requestanimationframe-for-smart-animating/
2 // http://my.opera.com/emoller/blog/2011/12/20/requestanimationframe-for-smart-er-animating
3
4 // requestAnimationFrame polyfill by Erik Möller. fixes from Paul Irish and Tino Zijdel
5
6 // MIT license
7
8 (function() {
9     var lastTime = 0;
10    var vendors = ['ms', 'moz', 'webkit', 'o'];
11    for(var x = 0; x < vendors.length && !window.requestAnimationFrame; ++x) {
12        window.requestAnimationFrame = window[vendors[x]+'RequestAnimationFrame'];
13        window.cancelAnimationFrame = window[vendors[x]+'CancelAnimationFrame']
14            || window[vendors[x]+'CancelRequestAnimationFrame'];
15    }
16
17    if (!window.requestAnimationFrame)
18        window.requestAnimationFrame = function(callback, element) {
19            var currTime = new Date().getTime();
20            var timeToCall = Math.max(0, 16 - (currTime - lastTime));
21            var id = window.setTimeout(function() { callback(currTime + timeToCall); },
22                timeToCall);
23            lastTime = currTime + timeToCall;
24            return id;
25        };
26
27    if (!window.cancelAnimationFrame)
28        window.cancelAnimationFrame = function(id) {
29            clearTimeout(id);
30        };
31 }());
```

request animation frame polyfill