

# Game Engines

CMPM 164, F2019

Prof. Angus Forbes (instructor)

[angus@ucsc.edu](mailto:angus@ucsc.edu)

Montana Fowler (TA)

[mocfowle@ucsc.edu](mailto:mocfowle@ucsc.edu)

Website: [creativecoding.soe.ucsc.edu/courses/cmpm164](http://creativecoding.soe.ucsc.edu/courses/cmpm164)

Slack: <https://ucsccmpm164.slack.com>

# Class information

## Class website:

<https://creativecoding.soe.ucsc.edu/courses/cmpm164>

## Slack is the main form of class communication:

<https://ucsccmpm164.slack.com>

## Our TA is Montana Fowler

They will lead the lab sections starting Friday

# Lab Sessions

## Lab:

Ming Ong Computer Center, Windows Lab – Merrill 103

Tuesdays 11am-12noon

Wednesdays 3pm-4pm

Thursdays 1pm-2pm

## Lab website:

<https://its.ucsc.edu/computer-labs/descriptions/mingong.html>

# Discussion

- What did you watch?
- What conference was it from?
- What did you learn that you didn't know before?
- What terms or topics were you not familiar with or were confusing?
- Is there something you saw that you saw that you would like to learn how to implement?

# Game Engines

- Why are there different game engines?
- Why use an existing one rather than creating your own?
- What is difficult about creating an engine?
- What design decisions do you have to make?
- What does a FPS need to emphasize? A Platformer? A fighting game? A racing game? An RTS?

# Game Engines

*Coordination* between various hardware components and software processes:

Communicating state

Sharing resources

Passing data

Organizing and ordering information

Prioritizing actions

Responding to inputs

Synchronizing outputs

# Game Engines

Which components/processes need to coordinate with each other?

How do they communicate to each other?

Is that the best way for them to communicate?

Is it the only way these components can be choreographed and linked together?

# Game Engines

Different temporal granularities:

- 60fps for graphics output
- 120fps for physics simulations
- 44,100hz for audio processing
- AI bots that only make decisions once they have reached a particular objective
- Characters that move only in response to user input, etc.

If all goes well, we have a seamless illusion of time

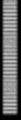
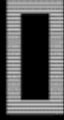


# Game Engines

## The Game Loop

- In an early arcade game like Pong or Space Invaders?
- In a contemporary multiplayer game like Fortnite?

(in class exercise)



# Game Engines

Pong:

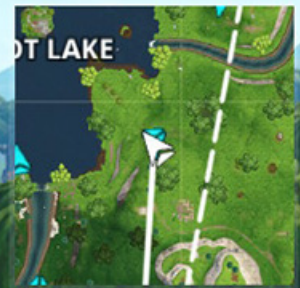
```
while (true) {
    readPlayerDevices();
    movePaddles();
    moveBall();
    collideAndBounceBall();
    if (ballHitsSide(LEFT)) {
        incrementScore(RIGHT); resetBall();
    } else if (ballHitsSide(RIGHT)) {
        incrementScore(LEFT); resetBall();
    }
    render();
}
```



ODDERZBreens  
#26627c8a9d-45460

- Nina.best
- ODDERZBreens
- teamteach
- WFE.Yolo\_Rouge

W 0 386 300 NW 330 343 N 0 180



6:29 43:43 0

- Switch Seat
- Exit

- WFE.Yolo\_Rouge
- Nina.best

152 135 112

6 23 III

0 | 100

+ 100 | 100

A ↑

Hotbar items: Pickaxe, Shotgun (24), Sniper Rifle (29), Medkit (1), Rocket Launcher (24)

X L R

# For Wednesday

- Keep going through Unreal learning paths, keep notes on anything that you don't understand completely.
- Homework 2 will be introduced on Wednesday.