

# Game Loop

Kristián Petráš

# Úvod ku game loopu

Motivácia:

- Chceme vytvoriť hru

```
while (true)
{
    game();
}
```

- Interakcia s užívateľom

# Iterácia designu

```
while (true)
{
    string command = readCommand();
    handleCommand(command);
}
```

- ▶ Ako ukončíme program?
- ▶ Viac typov eventov

# Iterácia designu

```
while (gameRunning)
{
    Event event = awaitEvent();
    handleEvent(event);
}
```

- ▶ Riešenie v praxi
- ▶ Hra nečaká na nás

# Iterácia designu

```
while (true)
{
    MSG msg;
    while (PeekMessage(&msg, NULL, 0, 0) > 0)
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }

    OneIteration();
}
```

# Iterácia designu

```
//init();  
while (gameRunning)  
{  
    processInput(); // abstrakcia  
    update();  
    render();  
}  
//dispose();
```

► Hotovo?

## Fixed step

```
while (gameRunning)
{
    double time = getCurrentTime();

    processInput();
    update();
    render();

    sleep(time + FRAME_MILLISECONDS - getCurrentTime());
}
```

- ▶ Čo ak to počítač nestíha
- ▶ Lepšie využitie času

## Variable step

```
double lastTime = getCurrentTime();

while (gameRunning)
{
    double current = getCurrentTime();
    double elapsed = current - lastTime;
    processInput();
    update(elapsed);
    render();
    lastTime = current;
}
```

► Multiplayer



## Fixed update, variable render

```
double previous = getCurrentTime();
double lag = 0.0;
while (gameRunning)
{
    double current = getCurrentTime();
    double elapsed = current - previous;
    previous = current;
    lag += elapsed;
    processInput();
    while (lag >= UPDATE_MILLISECONDS)
    {
        update();
        lag -= UPDATE_MILLISECONDS;
    }
    render();
}
```

- Problém s renderováním

## Fixed update, variable render interpolation

```
double previous = getCurrentTime();
double lag = 0.0;
while (gameRunning)
{
    double current = getCurrentTime();
    double elapsed = current - previous;
    previous = current;
    lag += elapsed;
    processInput();
    while (lag >= UPDATE_MILLISECONDS)
    {
        update();
        lag -= UPDATE_MILLISECONDS;
    }
    render(lag / UPDATE_MILLISECONDS);
}
```

# Zhrnutie

## Kto vlastní Game Loop

- ▶ Platforma
- ▶ Engine
- ▶ Ty

## Procesor

- ▶ 100% výkon
- ▶ Obmedziť výkon

## Kontrola plynutia hry

- ▶ Fixná doba ticku bez synchronizácie
- ▶ Fixná doba ticku so synchronizáciou
- ▶ Dynamická doba ticku
- ▶ Fixná doba updatu, dynamická doba renderu

# Unity monobehaviour

