

A wide-angle photograph of a rugged, mountainous landscape. In the foreground, two people wearing backpacks and riding dirt bikes are standing on a rocky outcrop. One person is facing away from the camera, while the other is partially visible behind them. The background features a deep valley with a winding road, a large lake with small islands, and distant mountains under a blue sky with scattered clouds.

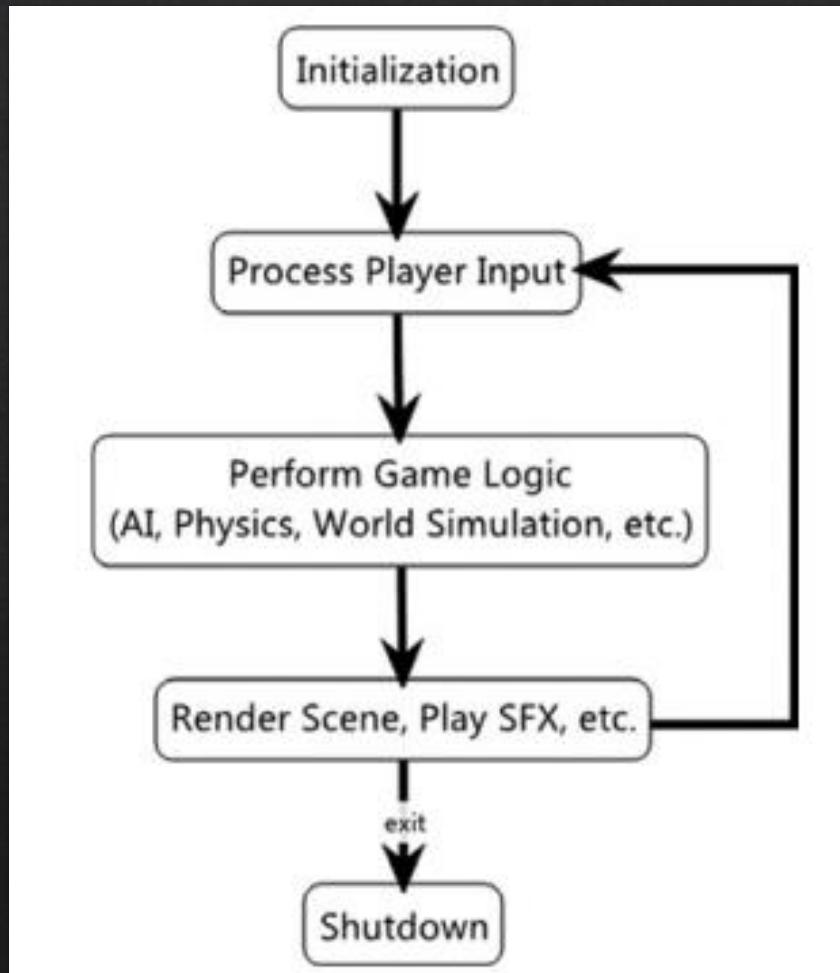
GAME PROGRAMMING IN C++

2020

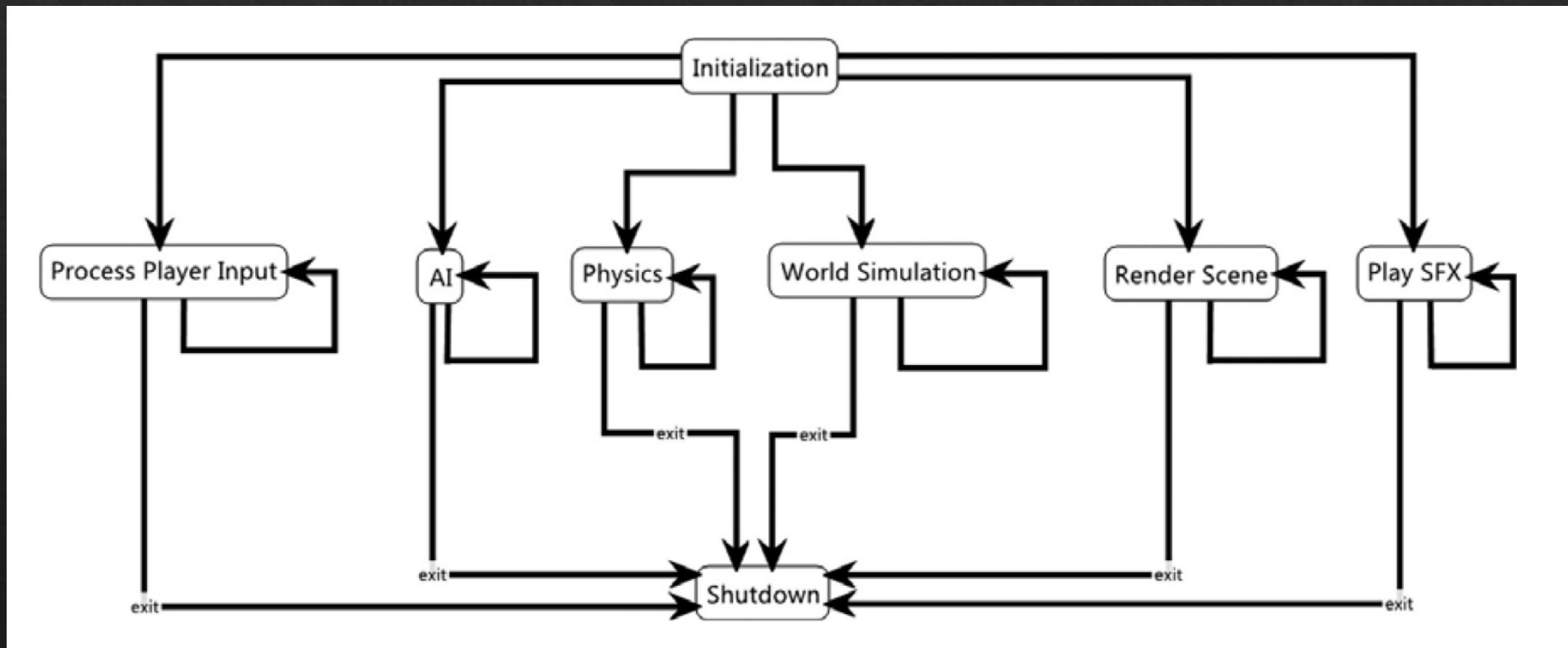
Game Loop

- ❖ Svaka igra ima niz operacije koje se dešavaju u krug u krug i tako menjaju svoje stanje i prikazuju ga igraču
- ❖ Tipičan game loop obrađuje sledeće sisteme:
 - ❖ Input, AI, animacije, fizika, simulacija sveta, render, zvuk, umrežavanje...
- ❖ Svaka iteracija sme da traje najviše 33ms
 - ❖ Neki sistemi se češće obrađuju, neki redje

Game Loop



Game Loop

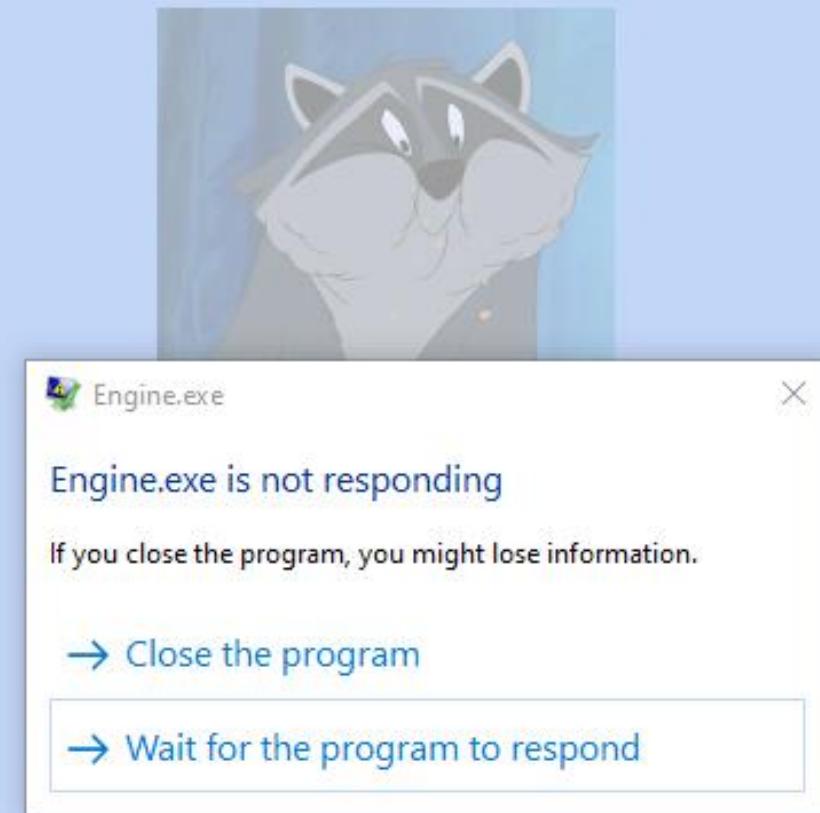


Game Loop + Operativni Sistem

- ❖ Windows podržava multitasking – moramo da naučimo da se slažemo sa Windowsom i ostalim aplikacijama koje se izvršavaju u isto vreme kad i naša
- ❖ Windows aplikacije su event-driven
- ❖ Operativni sistem komunicira sa aplikacijom porukama na koje aplikacija **mora** da reaguje
- ❖ Window procedure - svaki prozor ima posebnu funkciju koja obrađuje ove poruke
 - ❖ Nama SDL daje wrapper za to
- ❖ Tipovi poruka:
 - ❖ Stvaranje prozora, pomeranje miša, pritisak dugmeta na tastaturi, pomeranje prozora, menjanje veličine prozora, ulazak miša u prozor, izlazak miša iz prozora, cut, copy, paste, dobijanje i gubljenje fokusa...
 - ❖ [Lista svih poruka](#)

Windows Message Loop

Ukoliko aplikacija ne reaguje
na poruke



Windows Message Loop

- ❖ Windows dokumentacija nam pomaže da napišemo kod za Message Loop

// WARNING: Don't actually write your loop this way.

```
MSG msg = { };
while (1)
{
    GetMessage(&msg, NULL, 0, 0); → This function removes the first message from the head of the queue. If the queue is empty, the function blocks until another message is queued.
    TranslateMessage(&msg); → The TranslateMessage function is related to keyboard input. It translates keystrokes (key down, key up) into characters. You do not really have to know how this function works; just remember to call it before DispatchMessage.
    DispatchMessage(&msg); → The DispatchMessage function tells the operating system to call the window procedure of the window that is the target of the message. In other words, the operating system looks up the window handle in its table of windows, finds the function pointer associated with the window, and invokes the function.
}
```

Windows Message Loop

- ❖ [Windows dokumentacija](#) nam pomaže da napišemo kod za Message Loop

```
MSG msg = { };  
while (PeekMessage(&msg, hwnd, 0, 0, PM_REMOVE)) →  
{  
    TranslateMessage(&msg);  
    DispatchMessage(&msg);  
}
```

PeekMessage is similar to the GetMessage function; both check a message queue for a message that matches the filter criteria and then copy the message to an MSG structure. The main difference between the two functions is that GetMessage does not return until a message matching the filter criteria is placed in the queue, whereas PeekMessage returns immediately regardless of whether a message is in the queue.

SDL Message Loop

- ❖ [LazyFoo SDL tutorial](#) nam pomaže da napišemo kod za Message Loop

```
SDL_Event e;
while (SDL_PollEvent( &e ) != 0)
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```

SDL Message Loop - Zadatak

- ❖ Uspešno izaći iz igre ^_(ツ)_/^-
- ❖ Naš main loop je u Engine::Application::Run()
- ❖ Zameniti `while(true)` petlju sa `SDL_PollEvent` petljom
- ❖ Reagovati na `SDL_Quit` u petlji
- ❖ Hint: http://lazyfoo.net/tutorials/SDL/03_event_driven_programming/index.php

Input uređaji



Figure 9.1. Standard joypads for the Xbox 360 and PlayStation 3 consoles.



Figure 9.2. The DualShock 4 joypad for the PlayStation 4.



Figure 9.3. The innovative Wii Remote for the Nintendo Wii.



Figure 9.4. The Wii U controller by Nintendo.



Figure 9.6. Many specialized input devices are available for use with consoles.



Figure 9.7. Steering wheel adapter for the Nintendo Wii.

Input uređaji – osnovne komponente

- ❖ Digitalni
 - ❖ Dugme – pritisnuto, nije pritisnuto
- ❖ Analogni
 - ❖ 1-, 2-, 3-axis
 - ❖ Apsolutne vs relativne koordinate
 - ❖ Dead Zones
 - ❖ Sensitivity
- ❖ Kamere

Svi moderni input uređaji su kombinacija ovih komponenti

Input uređaji – low level

- ❖ Polling
 - ❖ Aplikacija uzima stanje direktno od uređaja kad joj zatreba i reaguje na promene
 - ❖ Manje kašnjenje, ali je moguće propustiti neke inpute
 - ❖ Windows: GetAsyncKeyState() , DirectInput: IDirectInputDevice::GetDeviceState()
- ❖ Interrupts / Callbacks
 - ❖ Uređaji javljaju operativnom sistemu svoje stanje, a aplikacija reaguje na ove promene kad joj je zgodno
- ❖ Wireless uređaji
 - ❖ Često u posebnom threadu održavaju wireless komunikaciju

Sistemi igre ne treba da reaguju direktno na hardverske događaje

Input uređaji – high level

- ❖ Detekcija pritiska i puštanja dugmeta
- ❖ Detekcija više pritisnutih dugmića odjednom
- ❖ Detekcija dugog držanja dugmeta
- ❖ Input action -> Input Mapping -> Input remapping
- ❖ Input context
- ❖ Local multiplayer

Primer arhitekture input sistema

Input Manager

Input Context

Input Action

Button



Naša arhitektura

- ❖ Nećemo koristiti InputContext
 - ❖ Slobodni ste da ih dodajete ili ćemo to zajedno uraditi ako nam ostane vremena na kraju kursa
- ❖ `SDL_PollEvent()` vs `GetAsyncKeyState()`

Naša arhitektura

```
- enum class EInputAction
{
    PlayerMoveLeft = 0,
    PlayerMoveRight,
    PlayerMoveUp,
    PlayerMoveDown,
    PlayerFire,
    PlayerJump,
    PlayerAccelerate,
    Player2MoveLeft,
    Player2MoveRight,
    Player2MoveUp,
    Player2MoveDown,
    Player2Fire,
    Player2Jump,
    Player2Accelerate,
    PanCameraLeft,
    PanCameraRight,
    PanCameraUp,
    PanCameraDown,
    ToggleDebugMode,
    PauseGame,
    RestartGame,
    MenuSelect,

    InputActionCount
};
```

```
- enum class EInputActionState
{
    None = 0,
    JustPressed,
    Pressed,
    Released,
    InputActionStateCount
};
```

Naša arhitektura

```
namespace Engine
{
    class InputManager
    {
        public:
            bool Init();
            void Update(float dt);
            void Shutdown();

        private:
            void ProcessInput();
            bool IsButtonJustPressed(EInputAction _eAction) const;
            bool IsButtonPressed(EInputAction _eAction) const;
            bool IsButtonReleased(EInputAction _eAction) const;

        private:
            void InitKeybinds();

            std::unordered_map<EInputAction, KeyboardButton> m_InputActions{ };
            std::unordered_map<EInputAction, EInputActionState> m_InputActionStates{ };
    };
}
```

Input sistem - Zadatak

Popuniti Switch-Case u
InputManager::ProcessInput()