Unity Platform

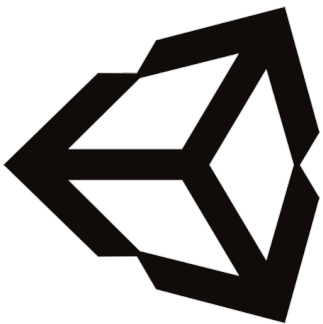
Demonstration: Game Loop, Tools, Project

Tomáš Polášek ipolasek@fit.vutbr.cz

Brno University of Technology

Faculty of Information Technology
DCGM, cPhoto@FIT
Faculty of Fine Arts
Game Media Studio

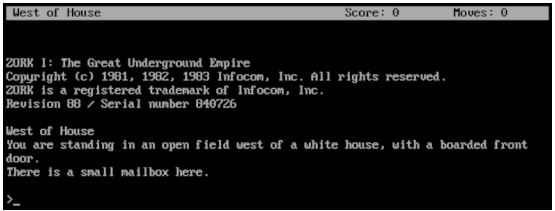
1



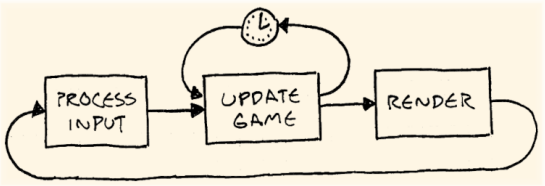
Game Loop

What is Looping?

- “Keep the Game Running” → Interactivity
- Execution Blocks
  1. Input Processing
  2. Game Update
  3. Render
- Soft Real-Time System
- Frames & Deadline



Source: Zork

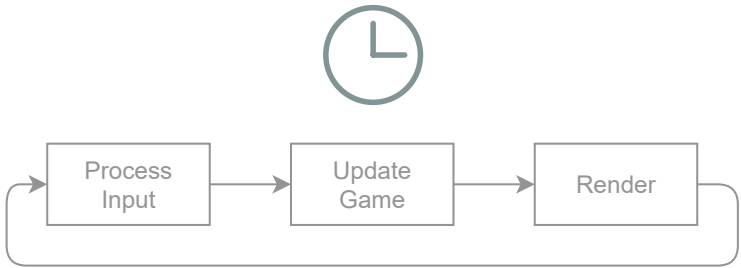


Source: Robert Nystrom – Game Programming Patterns [1]

© 2023, Tomáš Polášek, Martin Čadík

Simple Game Loop

```
while (!quit) {
    processInput();
    updateGame();
    render();
}
```



© 2023, Tomáš Polášek, Martin Čadík

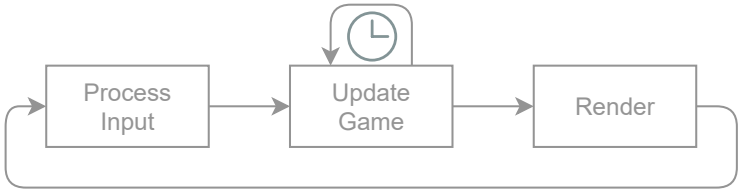
Delayed Game Loop

```
while (!quit) {
    var start = getTime();

    processInput();
    updateGame();
    render();

    sleep(FRAME_MS - (getTime() - start));
}
```

© 2023, Tomáš Polášek, Martin Čadík

Timed Game Loop

```
var previous = getTime();
var lag = 0.0;
while (!quit) {
    lag += getTime() - previous;
    previous = getTime();

    processInput();
    while (lag >= FRAME_MS)
    { updateGame(); lag -= FRAME_MS}
    render();
}
```



© 2023, Tomáš Polášek, Martin Čadík

Proper Game Loop

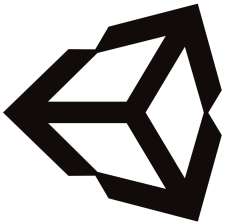
Unity

© 2023, Tomáš Polášek, Martin Čadík

Unity Engine

Tools of the Trade

- Unity
- GitHub for Unity
- JetBrains Rider
- Optionally:
  - Blender
  - GIMP
  - Inkscape
  - Krita



© 2023, Tomáš Polášek, Martin Čadík

Demonstration

Additional Resources

- [Manual] Unity User Manual
- [Manual] Unity Learn

© 2023, Tomáš Polášek, Martin Čadík

Thanks For
Your Attention!
150
143
SPREAD
CONCSN
MISSILE
020
LOCK
Descent

References I

[1] R. Nystrom. ***Game Programming Patterns***. UK: Genever Benning, 2014. ISBN: 0990582906.