

# PROTOTYPES AND INPUT

DEMONSTRATION: DEBUGGING, USER INPUT, HIERARCHY

TOMÁŠ POLÁŠEK [IPOLASEK@FIT.VUTBR.CZ](mailto:IPOLASEK@FIT.VUTBR.CZ)

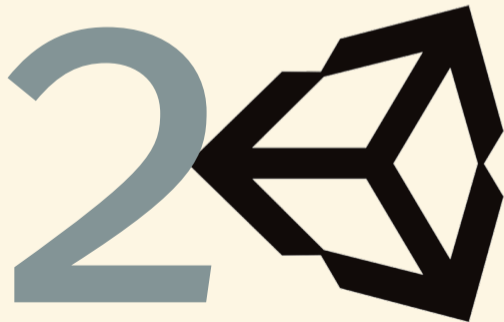
BRNO UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

DCGM, [CPhoto@FIT](mailto:CPhoto@FIT)

FACULTY OF FINE ARTS

GAME MEDIA STUDIO



# DEBUG AND PROFILING

- Messages
- Live Editor
- Break Points
- Unit Tests

→ Demonstration

# PROFILING WINDOW



# DEBUGGING ECS

- Entities & Systems Hierarchy
- Entity Debugger
- Burst Compiled Code

The screenshot displays the Unity Entity Debugger interface. On the left, a tree view shows the 'All Entities (Default World)' hierarchy, including 'Initialization' and 'Update' system groups. The main panel shows a table of entities with columns for 'Index' and 'Name'. On the right, a 'Chunk Utilization' graph shows two bars: one for 'WorldTime' (utilization 127) and one for 'WorldTimeQueue' (utilization 96). Below the graph, a list of components is shown, including 'BoxCollider', 'LocalToWorld', 'MeshFilter', 'MeshRenderer', 'PhysicsBodyAuthoring', 'PhysicsCollider', 'Rotation', 'Transform', and 'Translation'.

Index	Name
1	Ground
2	WorldTime
0	GameObject Scene: Main
3	Bullet
6	Entity 6
8	Entity 8
10	Entity 10
12	Entity 12
14	Entity 14
5	Enemy
7	Enemy
9	Enemy
11	Enemy
13	Enemy
15	Player
16	Enemy
18	Enemy
31	Enemy
22	Enemy

# USER INPUT

# INPUT MODULE

- Default **Input** Method
- “The Old Way”
- Stateless & Axis-Based
- Input Manager

```
void ProcessInput()
{
    if (Input.GetKeyDown(KeyCode.W)) { MoveForward(); }
    if (Input.GetButtonDown("Jump")) { DoJump(); }
    if (Input.GetAxis("Horizontal") > 0.0) { MoveCamera(); }
}
```

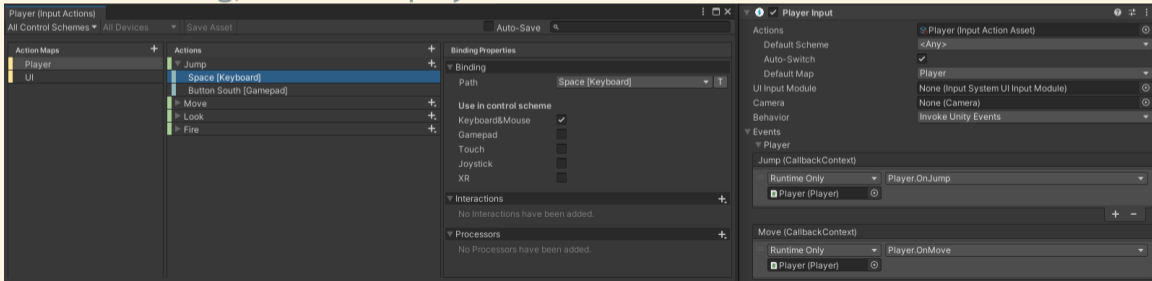


# INPUT SYSTEM

- **Input System** Package
- “The New Way”
- Stateless & Action-Based
- Mapping, Bindings, Virtual
- Higher Complexity
- Rebinding, Local Multiplayer

```
public void OnMove(InputAction.CallbackContext ctx)
{ DoMove( ctx.ReadValue<Vector2>()); }

public void OnJump(InputAction.CallbackContext ctx)
{ if (ctx.started) { DoJump(); } }
```



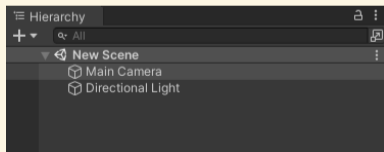


# SCENE HIERARCHY

# USING GAMEOBJECT

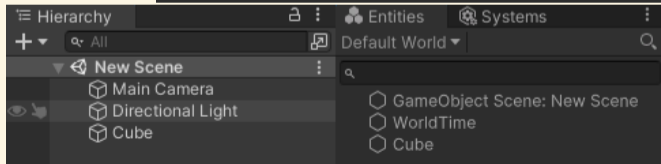
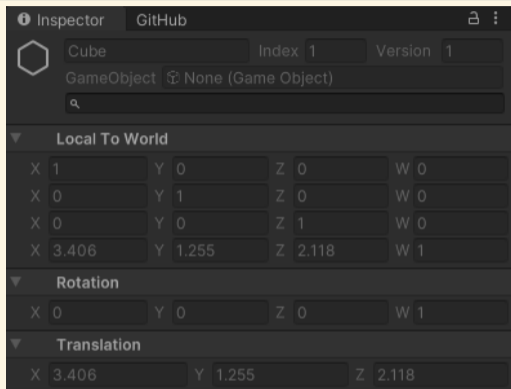
- The Main Hierarchy
- Hybrid **OOP** with **Components**
  - ▶ **GameObject** = (Almost) Empty Container
  - ▶ **Component** as Base
  - ▶ **Behavior** can be Enabled
  - ▶ **MonoBehavior** adds Functionality
- Editor Integration

→ Demonstration



- The Experimental Hierarchy
- Pure **Entity-Component-System**
  - ▶ **Entity** = Identifier
  - ▶ **Component** = Pure Data
  - ▶ **System** = Logic & Behavior
  - ▶ **World** = Entity Groups
- **Creation vs Conversion** Workflow

→ **Demonstration**



## ADDITIONAL RESOURCES

- [Manual] Unity User Manual
- [Manual] Unity Learn
- [YouTube] Luna Meier: Entity Interaction

LUMBER: 3700

GOLD: 2100

Thanks For  
Your Attention!

Warcraft: Orcs & Humans



# REFERENCES I