A tale of three UI frameworks

A decade of evolving developer and designer workflows in a game engine

unity

Who are we?

Adam Mechtley

Lead Developer

DOTS Physics & Rigging

Damian Campeanu

Developer

Editor & UI

What are we going to talk about?

- Brief overview of Unity Architecture
- IMGUI
- uGUI
- UI Toolkit
- Q&A

What is Unity and how do people make stuff with it?

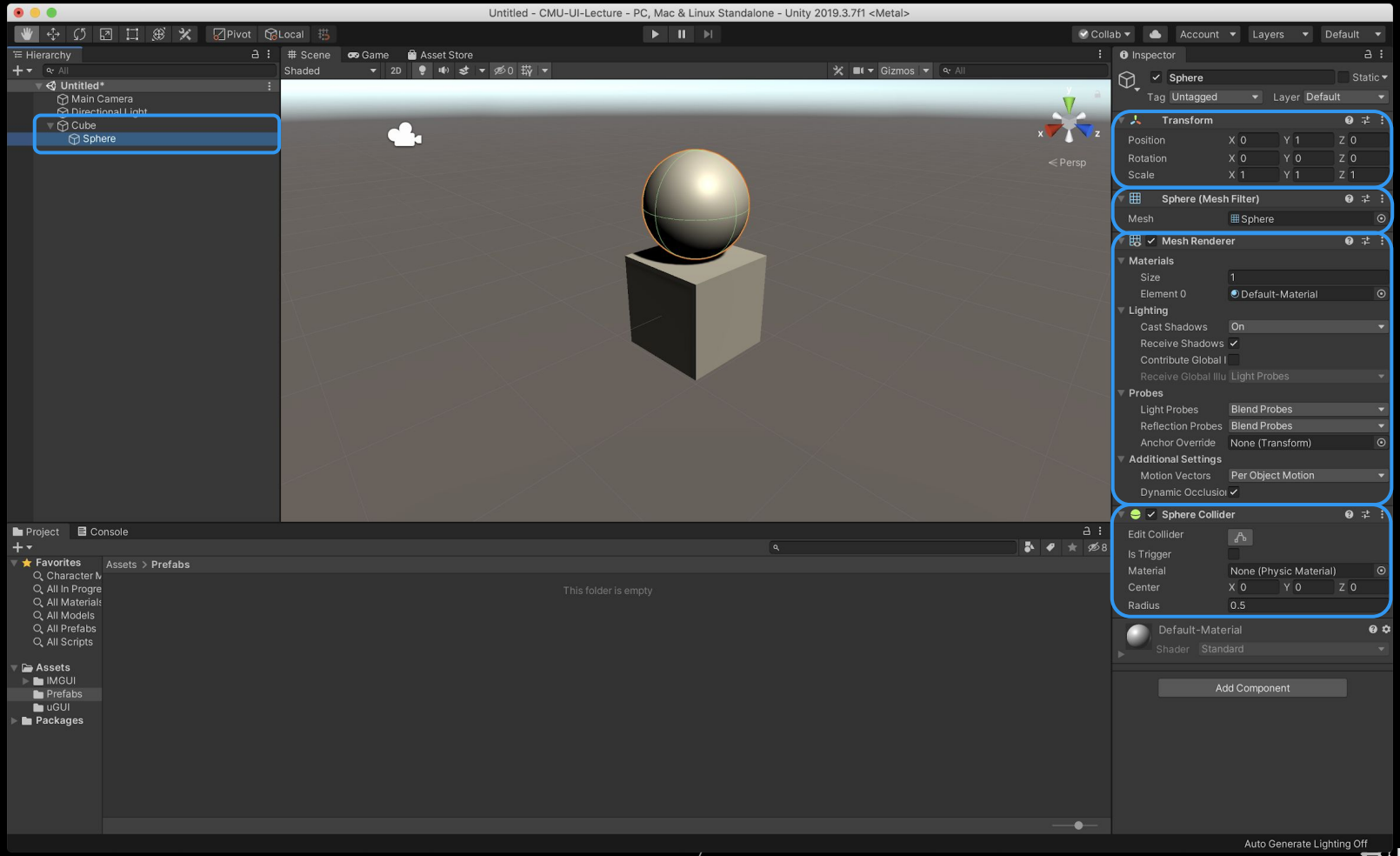
* Oversimplified version

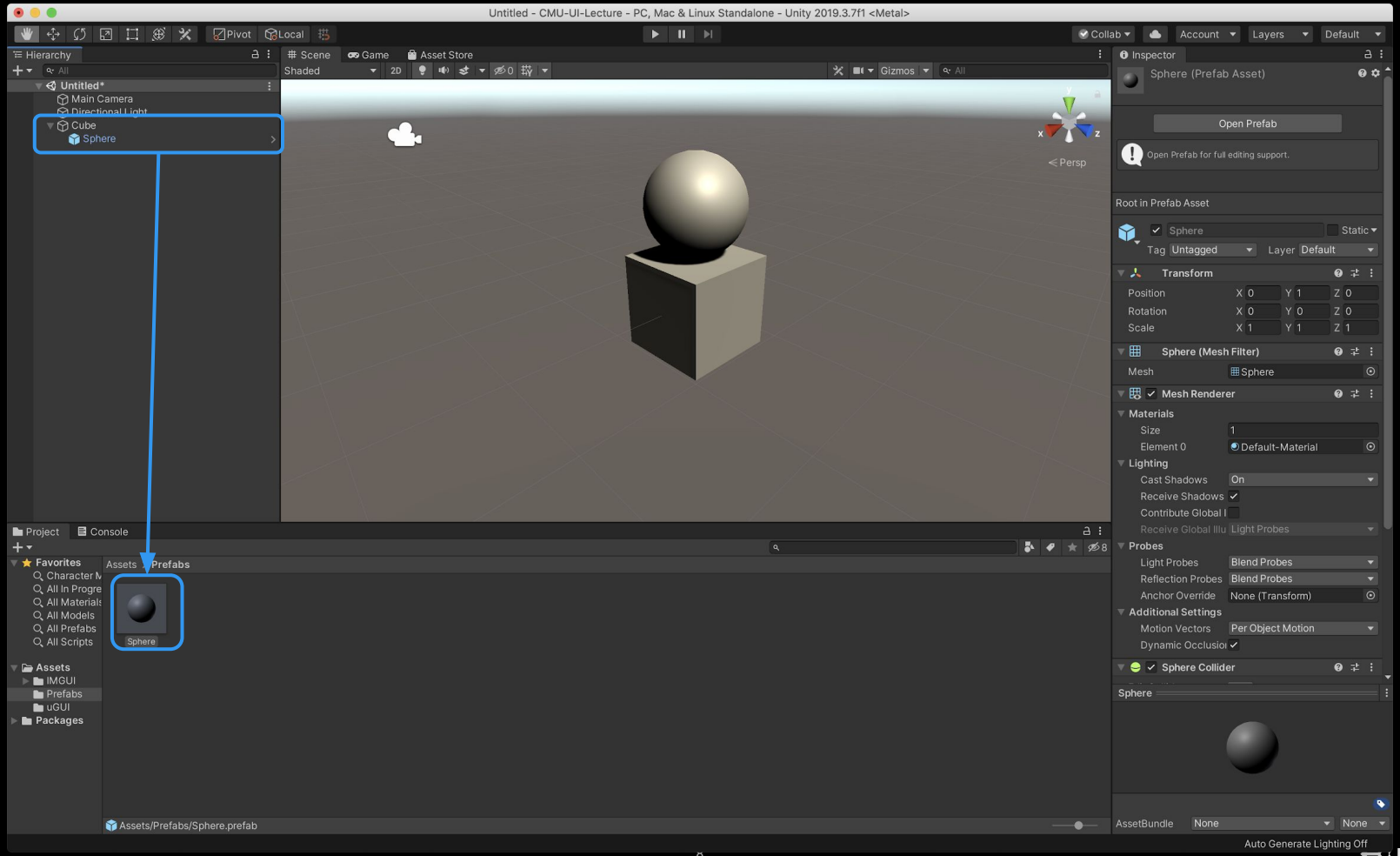
More than just a game engine!

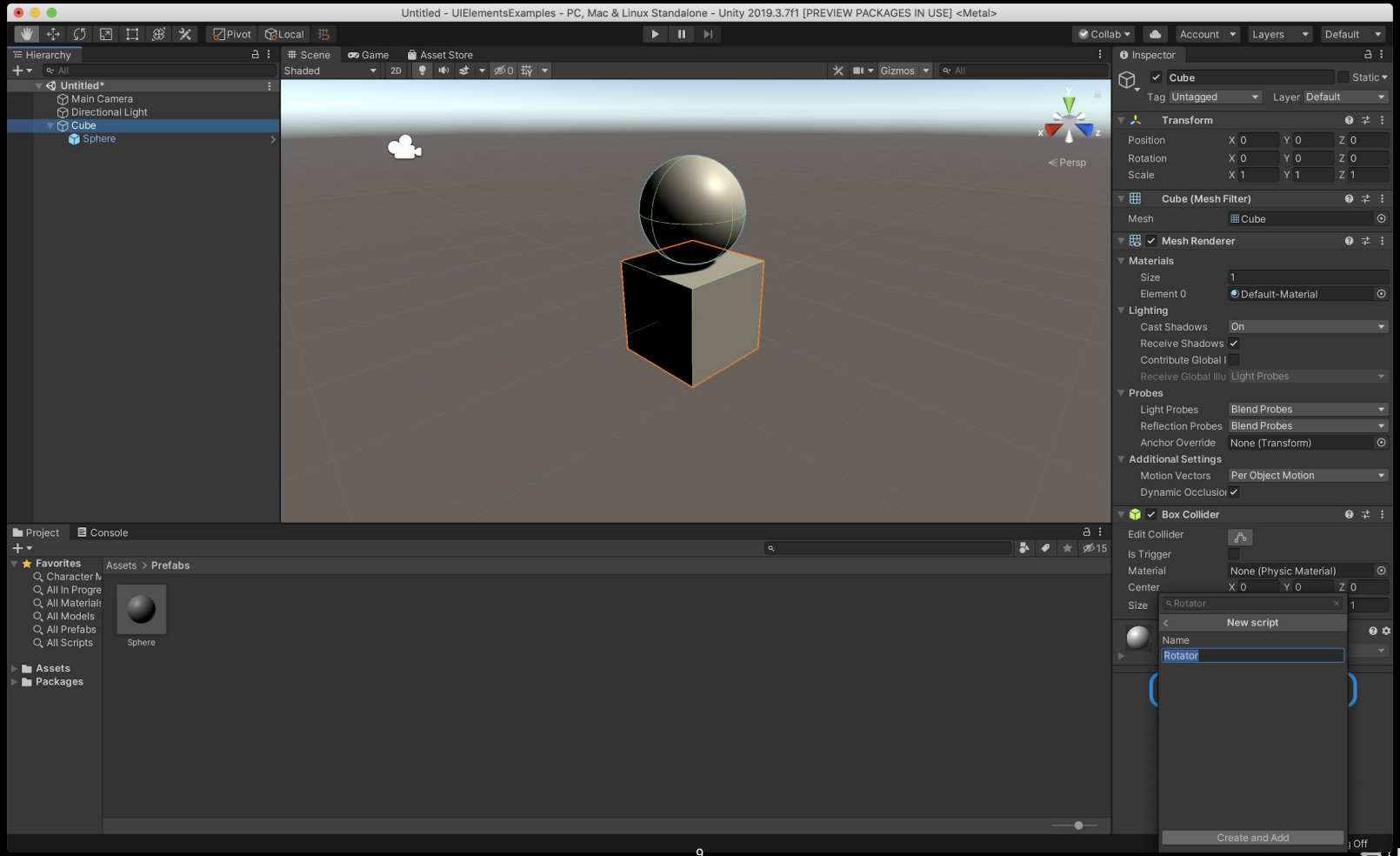
- Run-Time
  - Built-in libraries (input, animation, physics, UI, rendering, etc.)
  - Compatible with much of .NET ecosystem
- Editor
- Services
  - Analytics, live ops, etc.
- Asset Store
- Millions of users, hundreds of thousands active monthly

How do users make stuff with Unity?

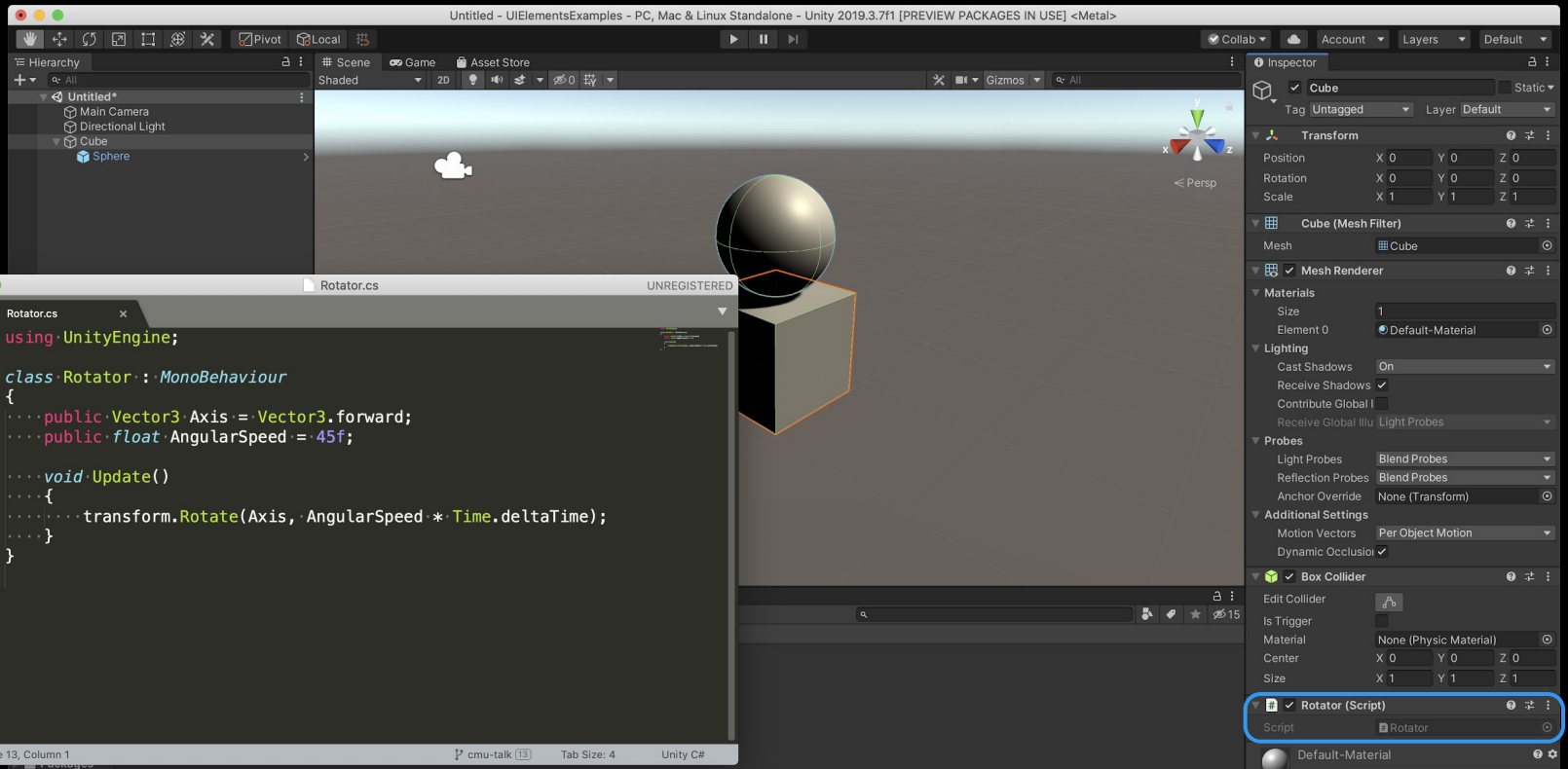
- Create **GameObjects** and add **Components** to produce behavior
  - Create new Components via C#
- Create reusable **Prefabs** from GameObjects
  - Prefabs can be nested in each other with overrides

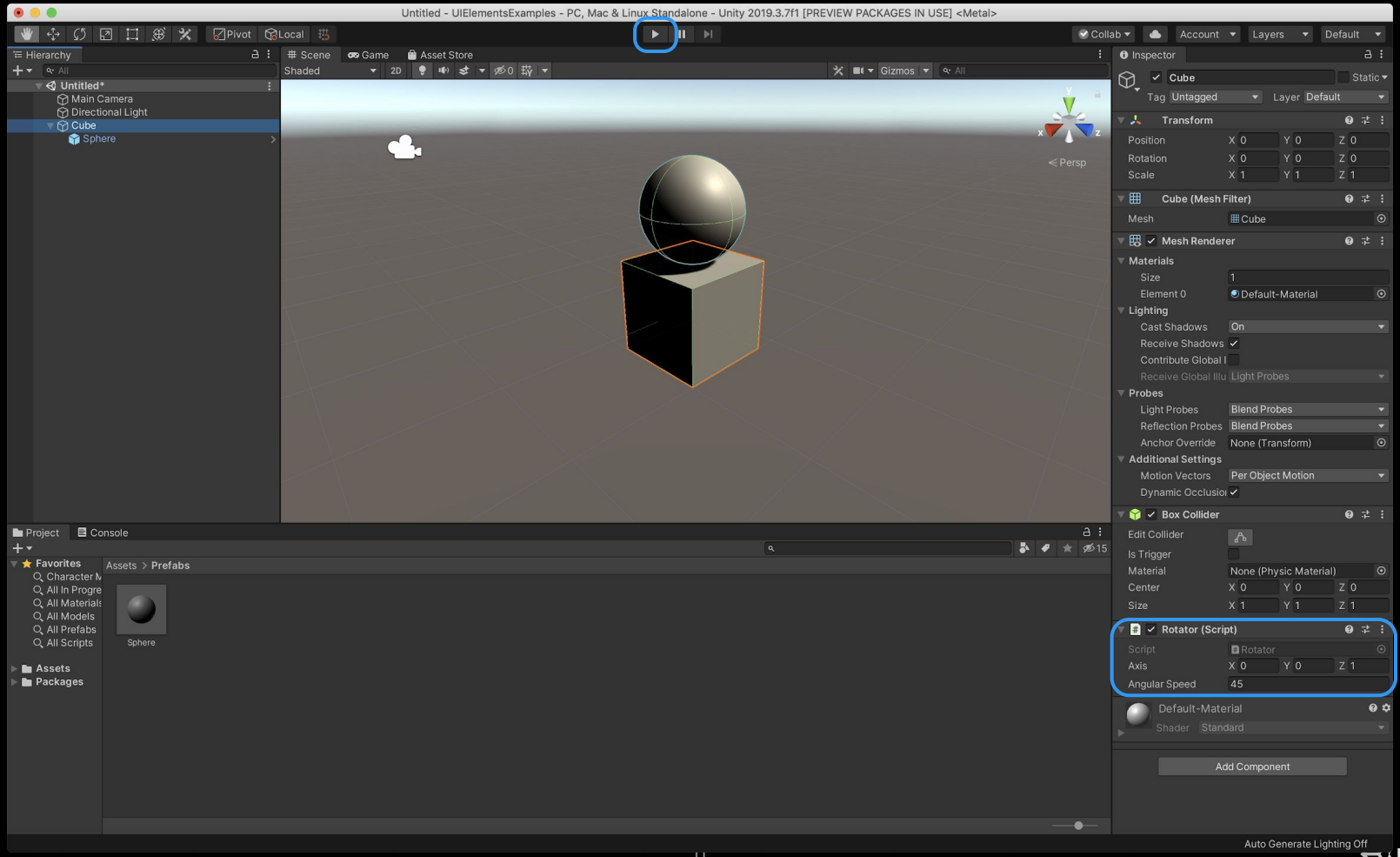

Untitled - CMU-UI-Lecture - PC, Mac & Linux Standalone - Unity 2019.3.7f1 <Metal>
Pivot
Local
Collab
Account
Layers
Default
Hierarchy
Untitled*
Main Camera
Directional Light
Cube
Sphere
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp
Inspector
Sphere
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 1 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Sphere (Mesh Filter)
Mesh Sphere
Mesh Renderer
Materials
Size 1
Element 0 Default-Material
Lighting
Cast Shadows On
Receive Shadows
Probes
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Additional Settings
Motion Vectors Per Object Motion
Sphere Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 0 Y 0 Z 0
Radius 0.5
Default-Material
Shader Standard
Add Component
Project
Console
Favorites
All Prefabs
All Scripts
Assets
IMGUI
Prefabs
uGUI
Packages
Assets > Prefabs
This folder is empty
Auto Generate Lighting Off
unity

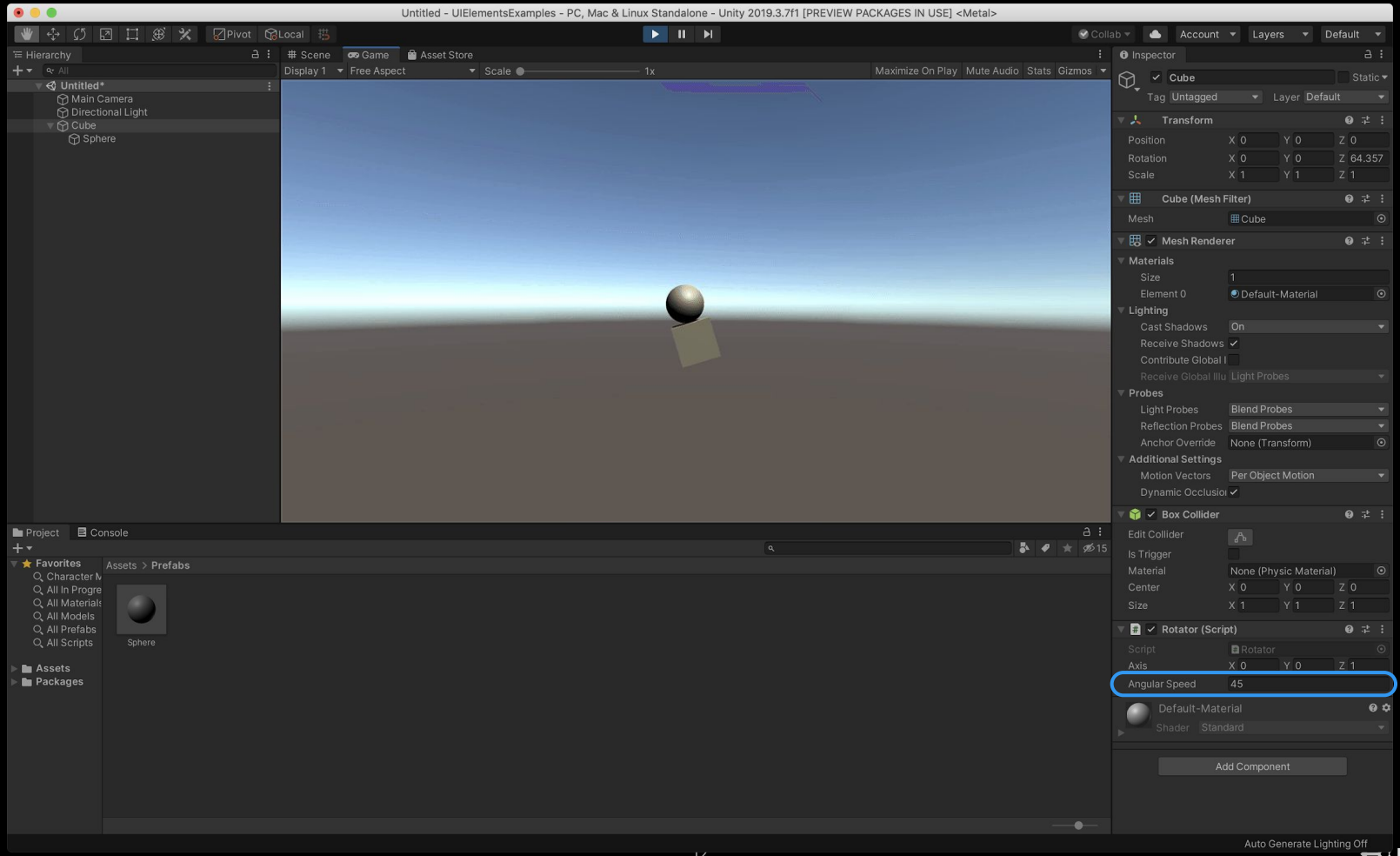

Untitled - CMU-UI-Lecture - PC, Mac & Linux Standalone - Unity 2019.3.7f1 <Metal>
Hierarchy
Untitled*
Main Camera
Directional Light
Cube
Sphere
Scene
Game
Asset Store
Persp
Inspector
Sphere (Prefab Asset)
Open Prefab
Open Prefab for full editing support.
Root in Prefab Asset
Sphere
Static
Tag Untagged
Layer Default
Transform
Position
Rotation
Scale
Sphere (Mesh Filter)
Mesh
Mesh Renderer
Materials
Size
Element 0
Default-Material
Lighting
Cast Shadows
Receive Shadows
Probes
Light Probes
Blend Probes
Reflection Probes
Anchor Override
None (Transform)
Additional Settings
Motion Vectors
Per Object Motion
Sphere Collider
Project
Console
Favorites
Assets
Prefabs
Sphere
IMGUI
uGUI
Packages
Assets/Prefabs/Sphere.prefab
AssetBundle
None
Auto Generate Lighting Off
unity


Untitled - UIElementsExamples - PC, Mac & Linux Standalone - Unity 2019.3.7f1 [PREVIEW PACKAGES IN USE] <Metal>
Pivot
Local
Collab
Account
Layers
Default
Hierarchy
Untitled*
Main Camera
Directional Light
Cube
Sphere
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp
Inspector
Cube
Static
Tag Untagged
Layer Default
Transform
Position
Rotation
Scale
Cube (Mesh Filter)
Mesh
Mesh Renderer
Materials
Size
Element 0
Default-Material
Lighting
Cast Shadows
On
Receive Shadows
Probes
Light Probes
Blend Probes
Reflection Probes
Blend Probes
Anchor Override
None (Transform)
Additional Settings
Motion Vectors
Per Object Motion
Box Collider
Edit Collider
Is Trigger
Material
None (Physic Material)
Center
Size
Rotator
New script
Name
Rotator
Create and Add
Project
Console
Favorites
All In Progre
All Materials
All Models
All Prefabs
All Scripts
Assets
Packages
Assets > Prefabs
Sphere

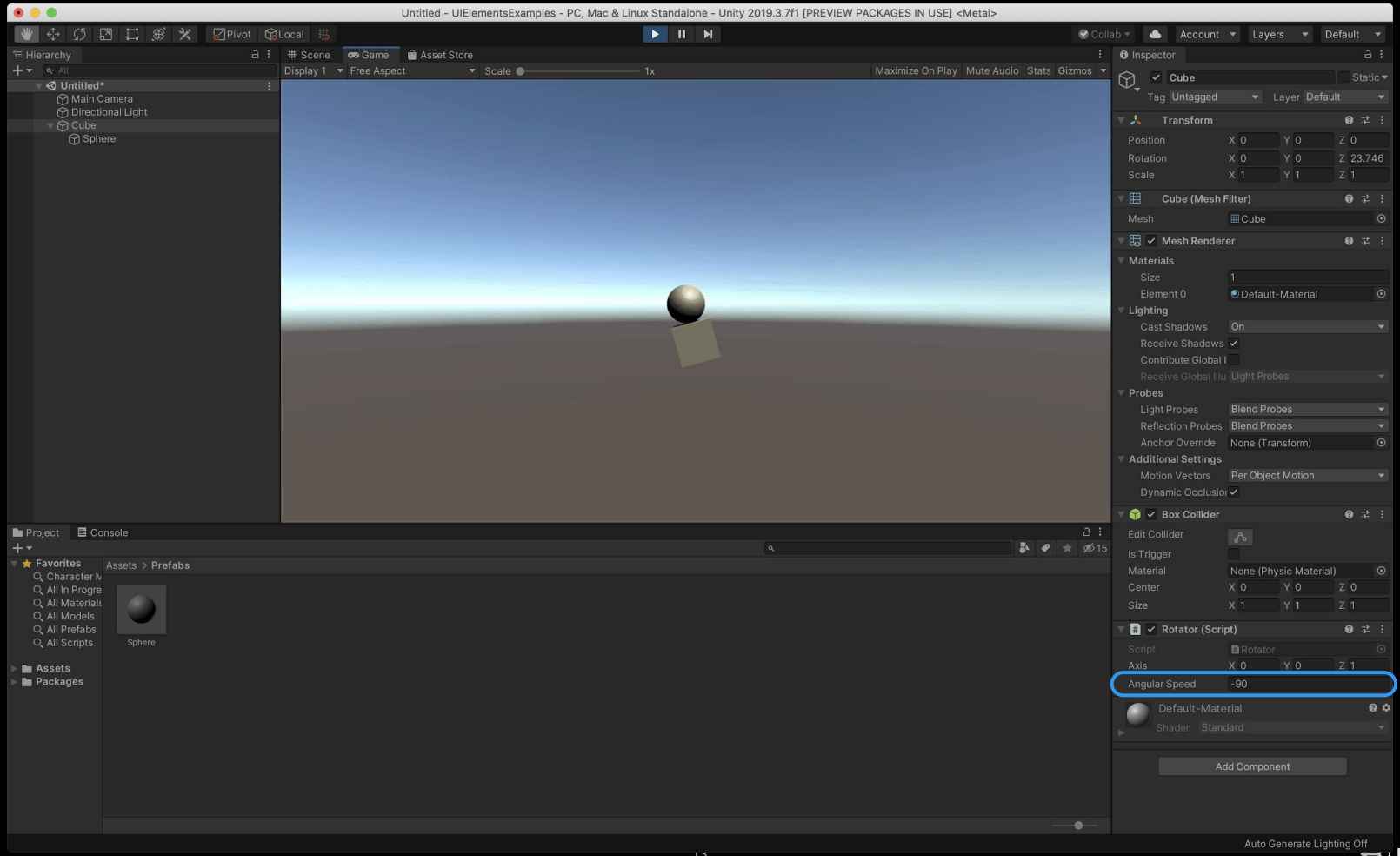
unity


Untitled - UIElementsExamples - PC, Mac & Linux Standalone - Unity 2019.3.7f1 [PREVIEW PACKAGES IN USE] <Metal>
Hierarchy
Untitled*
Main Camera
Directional Light
Cube
Sphere
Scene
Game
Asset Store
Persp
Rotator.cs
UNREGISTERED
using UnityEngine;
class Rotator : MonoBehaviour
{
public Vector3 Axis = Vector3.forward;
public float AngularSpeed = 45f;
void Update()
{
transform.Rotate(Axis, AngularSpeed * Time.deltaTime);
}
}
Line 13, Column 1
Tab Size: 4
Unity C#
Inspector
Cube
Transform
Position
Rotation
Scale
Cube (Mesh Filter)
Mesh
Mesh Renderer
Materials
Element 0
Default-Material
Lighting
Cast Shadows
Receive Shadows
Probes
Light Probes
Reflection Probes
Anchor Override
Additional Settings
Motion Vectors
Box Collider
Edit Collider
Is Trigger
Material
Center
Size
Rotator (Script)
Script
Shader
Standard
Add Component
Auto Generate Lighting Off


Untitled - UIElementsExamples - PC, Mac & Linux Standalone - Unity 2019.3.7f1 [PREVIEW PACKAGES IN USE] <Metal>
Pivot
Local
Collab
Account
Layers
Default
Hierarchy
Untitled*
Main Camera
Directional Light
Cube
Sphere
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp
Inspector
Cube
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Cube (Mesh Filter)
Mesh Cube
Mesh Renderer
Materials
Size 1
Element 0 Default-Material
Lighting
Cast Shadows On
Receive Shadows
Contribute Global I
Receive Global Illu Light Probes
Probes
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Additional Settings
Motion Vectors Per Object Motion
Dynamic Occlusio
Box Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1
Rotator (Script)
Script Rotator
Axis X 0 Y 0 Z 1
Angular Speed 45
Default-Material
Shader Standard
Add Component
Project
Console
Favorites
All In Progre
All Materials
All Models
All Prefabs
All Scripts
Assets
Packages
Assets > Prefabs
Sphere
Auto Generate Lighting Off
unity


Untitled - UIElementsExamples - PC, Mac & Linux Standalone - Unity 2019.3.7f1 [PREVIEW PACKAGES IN USE] <Metal>
Pivot
Local
Collab
Account
Layers
Default
Hierarchy
Scene
Game
Asset Store
Inspector
Untitled*
Main Camera
Directional Light
Cube
Sphere
Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Audio
Stats
Gizmos
Cube
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 64.357
Scale X 1 Y 1 Z 1
Cube (Mesh Filter)
Mesh Cube
Mesh Renderer
Materials
Size 1
Element 0 Default-Material
Lighting
Cast Shadows On
Receive Shadows
Probes
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Additional Settings
Motion Vectors Per Object Motion
Box Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1
Rotator (Script)
Script Rotator
Axis X 0 Y 0 Z 1
Angular Speed 45
Default-Material
Shader Standard
Add Component
Project
Console
Favorites
All Prefabs
All Scripts
Assets
Packages
Assets > Prefabs
Sphere
Auto Generate Lighting Off

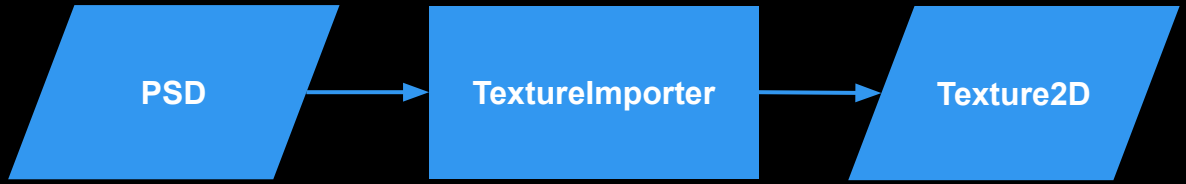
unity

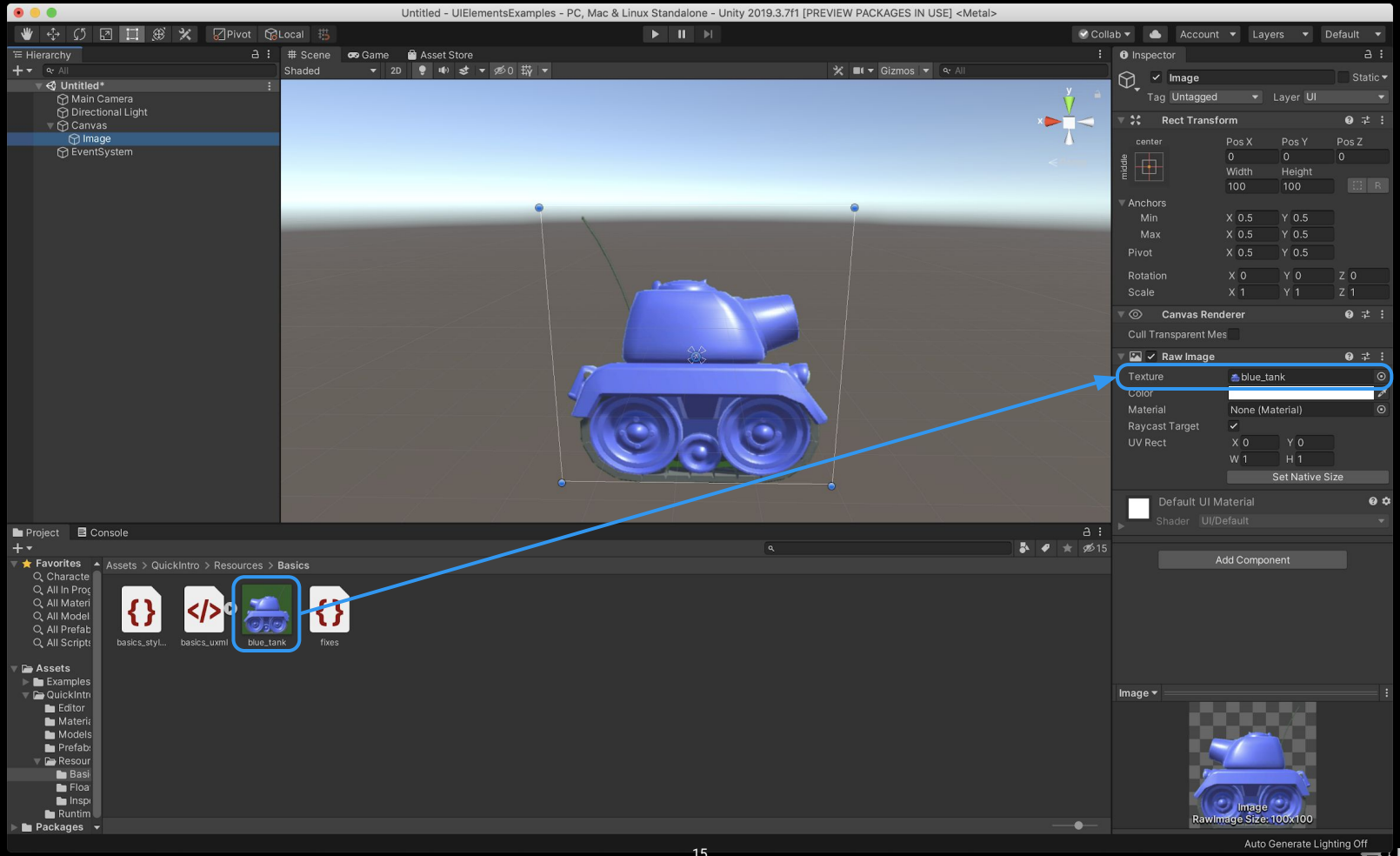

Untitled - UIElementsExamples - PC, Mac & Linux Standalone - Unity 2019.3.7f1 [PREVIEW PACKAGES IN USE] <Metal>
Pivot
Local
Collab
Account
Layers
Default
Hierarchy
Scene
Game
Asset Store
Inspector
Untitled*
Main Camera
Directional Light
Cube
Sphere
Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Audio
Stats
Gizmos
Cube
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 23.746
Scale X 1 Y 1 Z 1
Cube (Mesh Filter)
Mesh Cube
Mesh Renderer
Materials
Size 1
Element 0 Default-Material
Lighting
Cast Shadows On
Receive Shadows
Probes
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Additional Settings
Motion Vectors Per Object Motion
Box Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1
Rotator (Script)
Script Rotator
Axis X 0 Y 0 Z 1
Angular Speed -90
Default-Material
Shader Standard
Add Component
Project
Console
Favorites
All Prefabs
All Scripts
Assets
Packages
Assets > Prefabs
Sphere
Auto Generate Lighting Off

unity

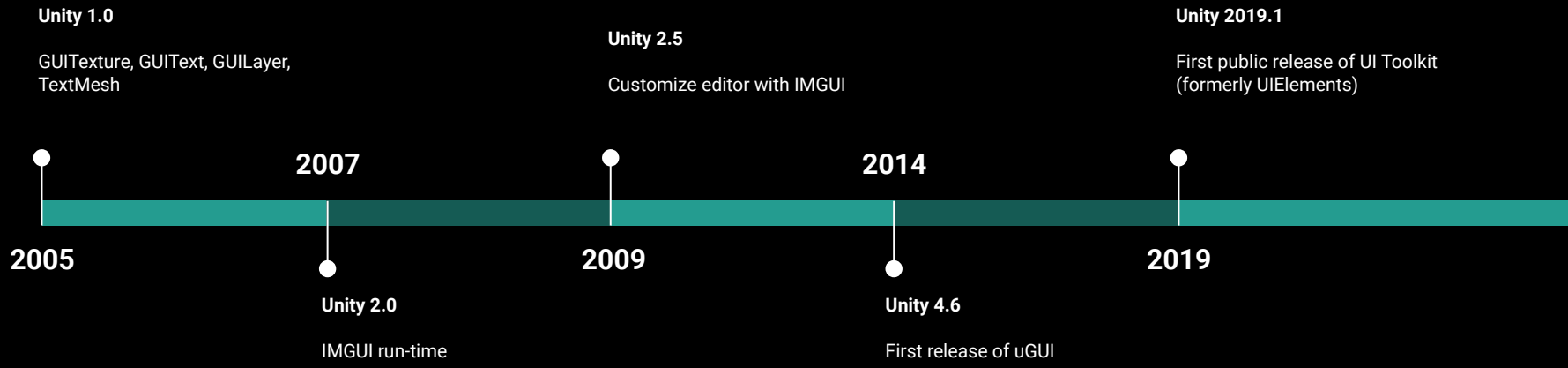
How do users create content for it?

- Save source file in project folder
- Built-in or custom asset importer generates artifact(s)



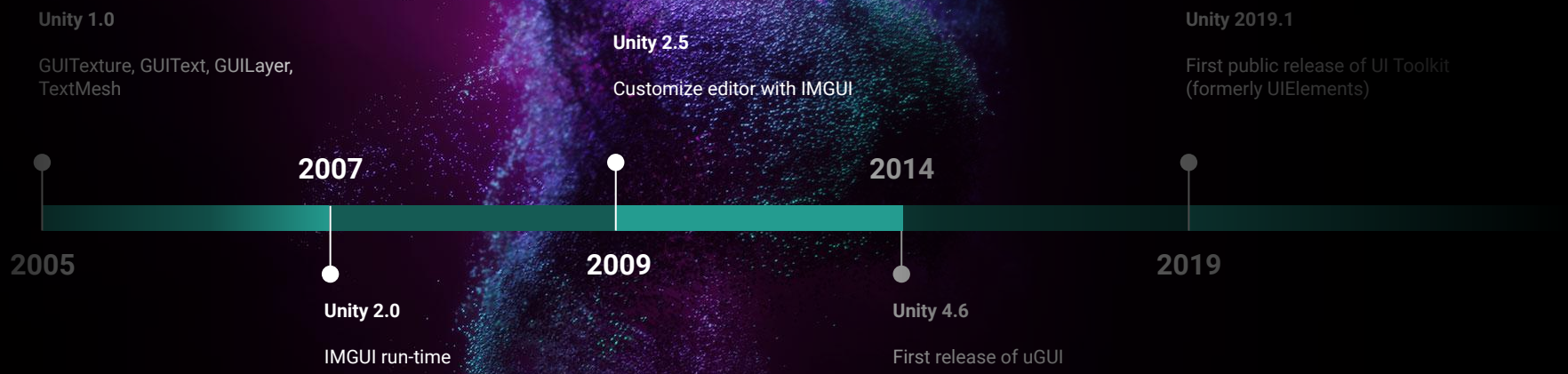

Untitled - UIElementsExamples - PC, Mac & Linux Standalone - Unity 2019.3.7f1 [PREVIEW PACKAGES IN USE] <Metal>
Hierarchy
Untitled*
Main Camera
Directional Light
Canvas
Image
EventSystem
Scene
Game
Asset Store
Inspector
Image
Rect Transform
Canvas Renderer
Raw Image
Texture
blue_tank
Color
Material
None (Material)
Raycast Target
UV Rect
Set Native Size
Default UI Material
Shader
UI/Default
Add Component
Project
Console
Assets > QuickIntro > Resources > Basics
basics_styl...
basics_uxml
blue_tank
fixes
Image
Image
RawImage Size: 100x100
Auto Generate Lighting Off

How do users create UI with it?



IMGUI

A simple UI framework for an ever-changing world



Design considerations

- Unity needs a UI framework! (both run-time and editor)
- Most Unity projects…
  - …are small web player experiences
  - …are created by small teams with few/broad role specializations
- Most game UI…
  - …communicates frequently updating values
  - …is non-diegetic overlays

IMGUI API

- `OnGUI()` callback
  - Event loop with `Event.current`
  - Call order determines event handling priority
- Library of static methods in `GUI` class for common functionality
  - `GUILayout` variants to assist with `Rect` calculations
  - Both run-time and editor-only variants for most types
- `GUIStyle` class
- `GUISkin` asset

API and Workflow Demo

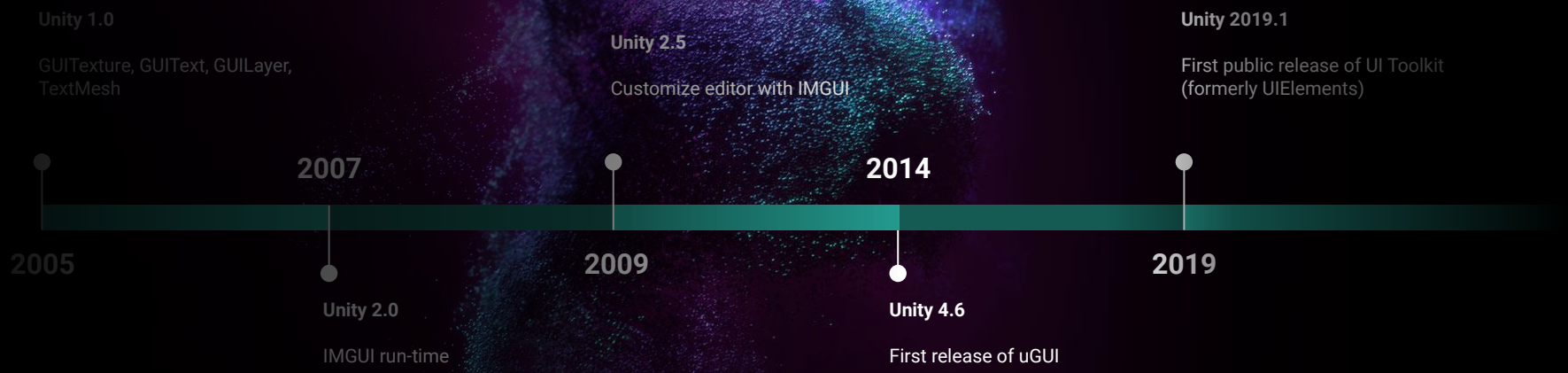
IMGUI advantages and disadvantages

+ Gathering and responding to input is trivial
+ Fast for programmers to prototype with
+ Works well for property grids
+ Simple API organization
+ Predictable performance

— But not very great performance
— Limited designer workflows
— No control over rendering pipeline
— Only supports non-diegetic UI
— Lots of manual work making new controls

uGUI

A framework for to make UI feel more like the rest of Unity



Design considerations

- Unity needs needs to empower designers to be productive more independently
- Most Unity projects…
  - …are created by teams with clearer role specializations
  - …run on mobile platforms where draw calls are expensive and display specifications vary wildly
- Most game UI…
  - …contains diegetic/spatial and non-diegetic elements
  - …is richly animated with effects

Fagerholt & Lorentzon (2009)

Exists in game fiction?	Visualized in 3D world? No	Visualized in 3D world? Yes
No	**Non-diegetic**	**Spatial**
Yes	**Meta**	**Diegetic**

uGUI API

- **UIBehaviour** base class inherits **MonoBehaviour**
  - Selectable, Graphic, etc sub-classes
- **Canvas** and **CanvasScalar** control rendering of hierarchies of elements
  - **RectTransform** (inherits **Transform**) use for layout
  - Most components draw **Sprite** assets
  - Set geometry, materials, etc. on child **CanvasRenderer**
- **StandaloneInputModule** and **EventSystem** gather and delegate input events
  - **BaseRaycaster** of some kind finds event handlers
  - **IPointerDownHandler**, **IPointerUpHandler**, **IDragHandler**, etc

API and Workflow Demo

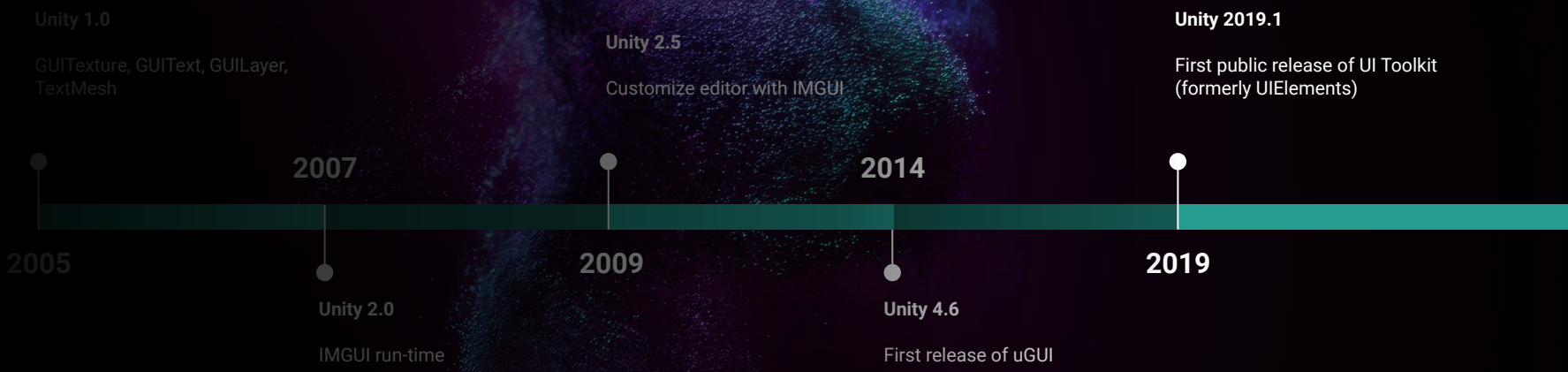
uGUI advantages and disadvantages

+ Designer workflows that fit with other Unity features (prefabs, animation, etc.)
+ Serializable event handlers
+ Automatic atlasing and scaling based on physical size, DPI, etc.
+ Diegetic UI
+ Common rendering pathway with everything else

— Performance overhead from GameObjects and Components
— Authoring data format hard to read and debug at a glance
— No centralized styling
— Canvases require specialized knowledge to optimize

UI Toolkit

A framework to make Unity feel more like the rest of the world



Generative Art — Made with Unity

unity

Design considerations

Collaboration
Different team members can work on different parts of the same UI.

Reusability
Share styles and templates within or across projects.

Iteration Speed
Quickly develop and validate UI for different contexts.

Extensibility
Customize and extend existing styles and templates or build custom ones.

Familiarity
UI authoring tools and workflows are familiar and easy to learn.

Rich Content
Build engaging UI that performs well as it scales.

UI Toolkit API

1

```
/* styles.uss */
.container { font-size: 40px; }
#random-explosion { color: blue; }
```

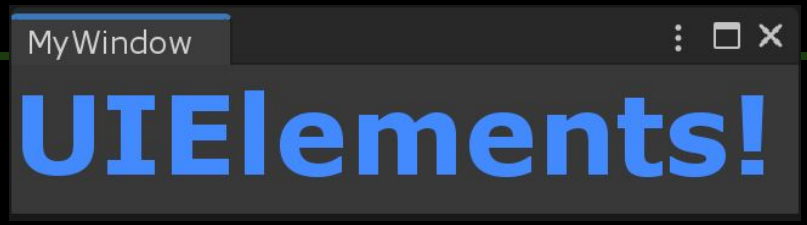
2

```
<!-- hierarchy.uxml -->
<UXML xmlns="UnityEngine.UIElements">
  <VisualElement class="container">
      <Style src="styles.uss" />
      <Label
          name="random-explosion"
          text="UIElements!" />
  </VisualElement>
</UXML>
```

3

```
// MyWindow.cs
void OnEnable() {
    var a = AssetDatabase.LoadAssetAtPath(
        "Assets/hierarchy.uxml");
    VisualElement row = a.CloneTree();
    var label =
        row.Q<Label>("random-explosion");
    label.RegisterCallback<MouseUpEvent>(
        evt => evt.StopPropagation());

    rootVisualElement.Add(row);
}
```



UI Toolkit API

1

```
/* styles.uss */
.container { font-size: 40px; }
#random-explosion { color: blue; }
```

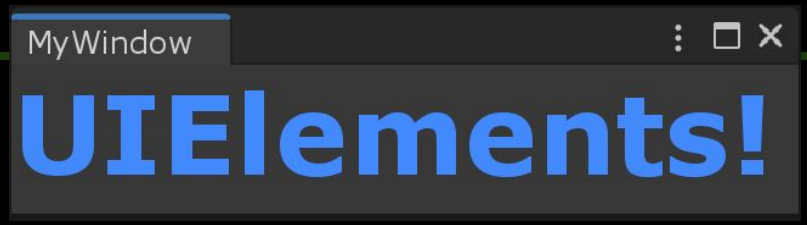
2

```
<!-- hierarchy.uxml -->
<UXML xmlns="UnityEngine.UIElements">
  <VisualElement class="container">
      <Style src="styles.uss" />
      <Label
          name="random-explosion"
          text="UIElements!" />
  </VisualElement>
</UXML>
```

3

```
// MyWindow.cs
void OnEnable() {
    var a = AssetDatabase.LoadAssetAtPath(
        "Assets/hierarchy.uxml");
    VisualElement row = a.CloneTree();
    var label =
        row.Q<Label>("random-explosion");
    label.RegisterCallback<MouseUpEvent>(
        evt => evt.StopPropagation());

    rootVisualElement.Add(row);
}
```



UI Toolkit API

1

```
/* styles.uss */
.container { font-size: 40px; }
#random-explosion { color: blue; }
```

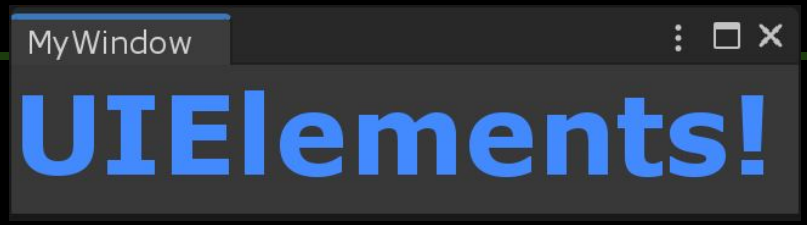
2

```
<!-- hierarchy.uxml -->
<UXML xmlns="UnityEngine.UIElements">
  <VisualElement class="container">
      <Style src="styles.uss" />
      <Label
          name="random-explosion"
          text="UIElements!" />
  </VisualElement>
</UXML>
```

3

```
// MyWindow.cs
void OnEnable() {
    var a = AssetDatabase.LoadAssetAtPath(
        "Assets/hierarchy.uxml");
    VisualElement row = a.CloneTree();
    var label =
        row.Q<Label>("random-explosion");
    label.RegisterCallback<MouseUpEvent>(
        evt => evt.StopPropagation());

    rootVisualElement.Add(row);
}
```



UI Toolkit API

1

```
/* styles.uss */
.container { font-size: 40px; }
#random-explosion { color: blue; }
```

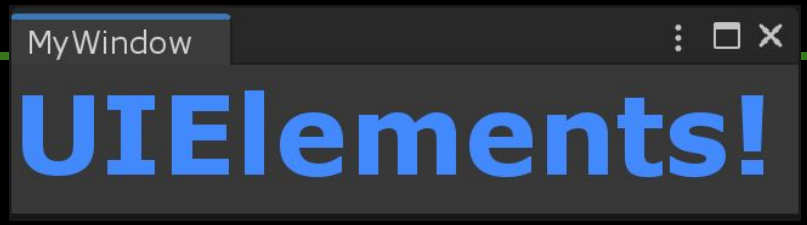
2

```
<!-- hierarchy.uxml -->
<UXML xmlns="UnityEngine.UIElements">
  <VisualElement class="container">
      <Style src="styles.uss" />
      <Label
          name="random-explosion"
          text="UIElements!" />
  </VisualElement>
</UXML>
```

3

```
// MyWindow.cs
void OnEnable() {
    var a = AssetDatabase.LoadAssetAtPath(
        "Assets/hierarchy.uxml");
    VisualElement row = a.CloneTree();
    var label =
        row.Q<Label>("random-explosion");
    label.RegisterCallback<MouseUpEvent>(
        evt => evt.StopPropagation());

    rootVisualElement.Add(row);
}
```



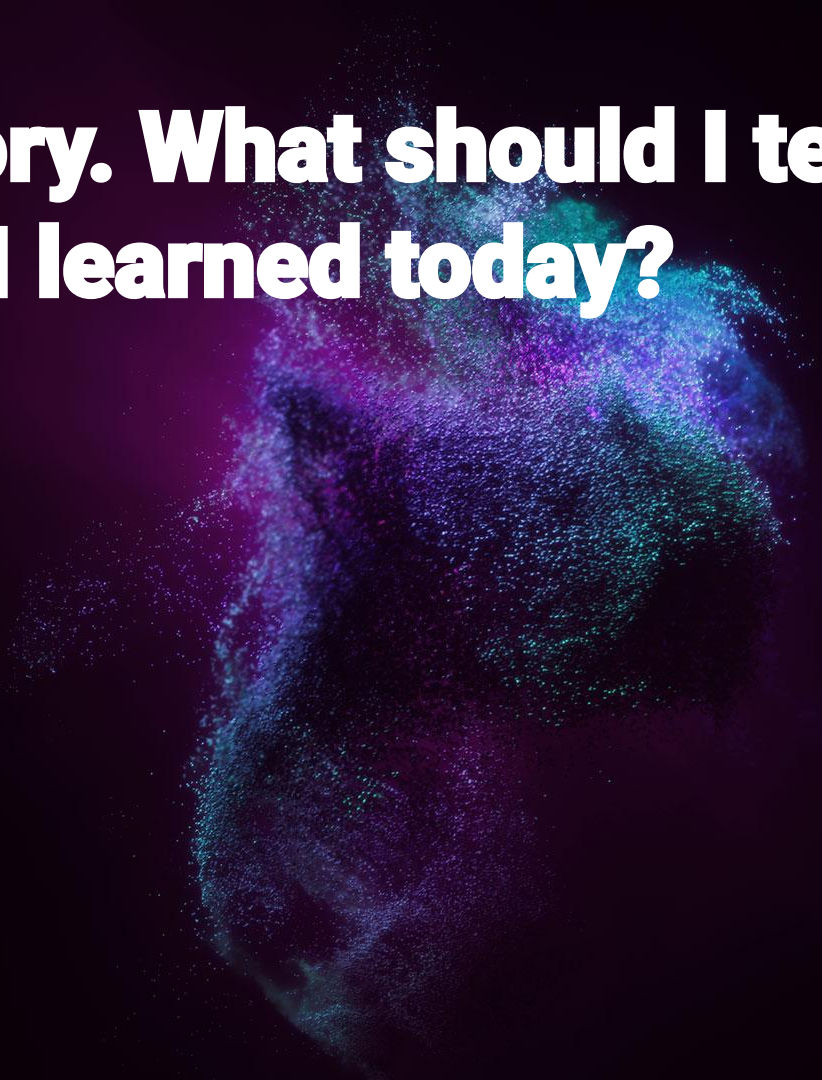
API and Workflow Demo

UI Toolkit advantages and disadvantages

+ Great performance for most use cases
+ Powerful automatic layouting via Flexbox
+ Centralized styling using standard paradigms (CSS)
+ Visual authoring without writing code
+ One API for both Editor and Runtime

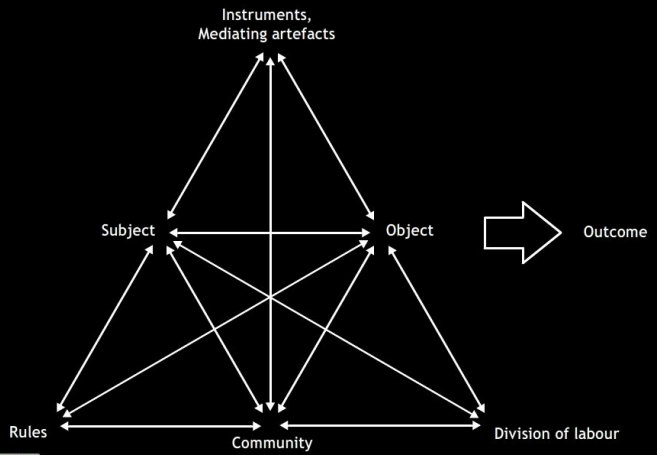
— Name-based handles can easily break
— Inefficient when lots of things are changing at once
— More complicated Event-based value bindings
— More complicated bindings to Unity objects and gameplay

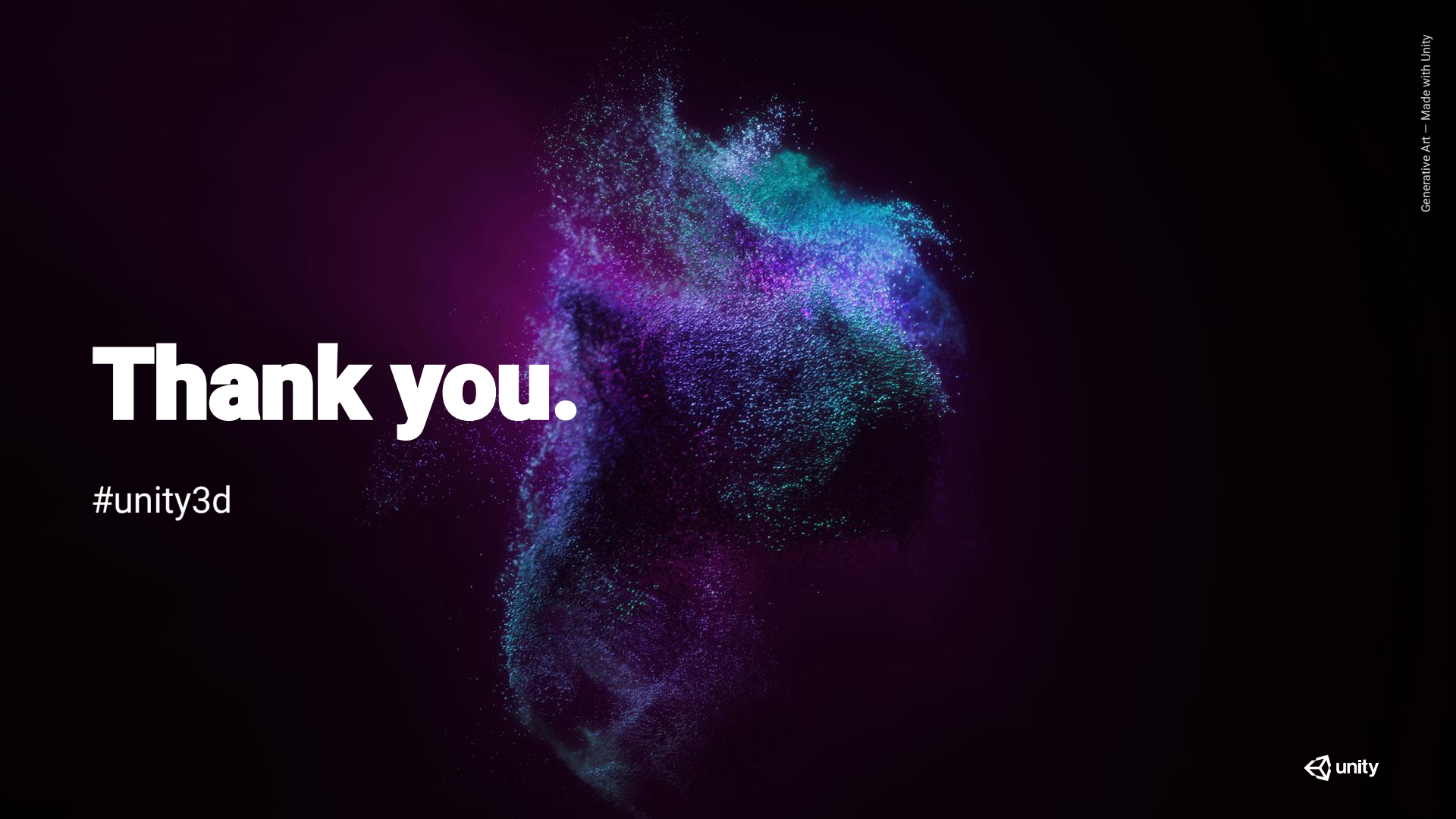
Cool story. What should I tell my friends I learned today?



Final thoughts

- Immediate-mode and retained-mode GUI each have strengths and disadvantages in different situations
- As the rest of the world evolves, so, too, must your API
  - Everything comes with a maintenance cost
- Reasonableness of API design decisions is very contextual
  - Aesthetic tastes of the historical moment
  - Technical requirements of target hardware
  - Tools ecosystem
- Design influences users' expressive capabilities




Thank you.
#unity3d
Generative Art – Made with Unity
unity