Building a unified cross-project UI framework (Unity)

Natalia Rebrova
UI/UX specialist SYBO games

GAME DEVELOPERS CONFERENCE
MARCH 18–22, 2019 | #GDC19

NEXT 30 MIN:

- Why building a UI framework can be beneficial for a multiple teams / projects company
- How to approach
- Overview of our main UI framework components
- Benefits

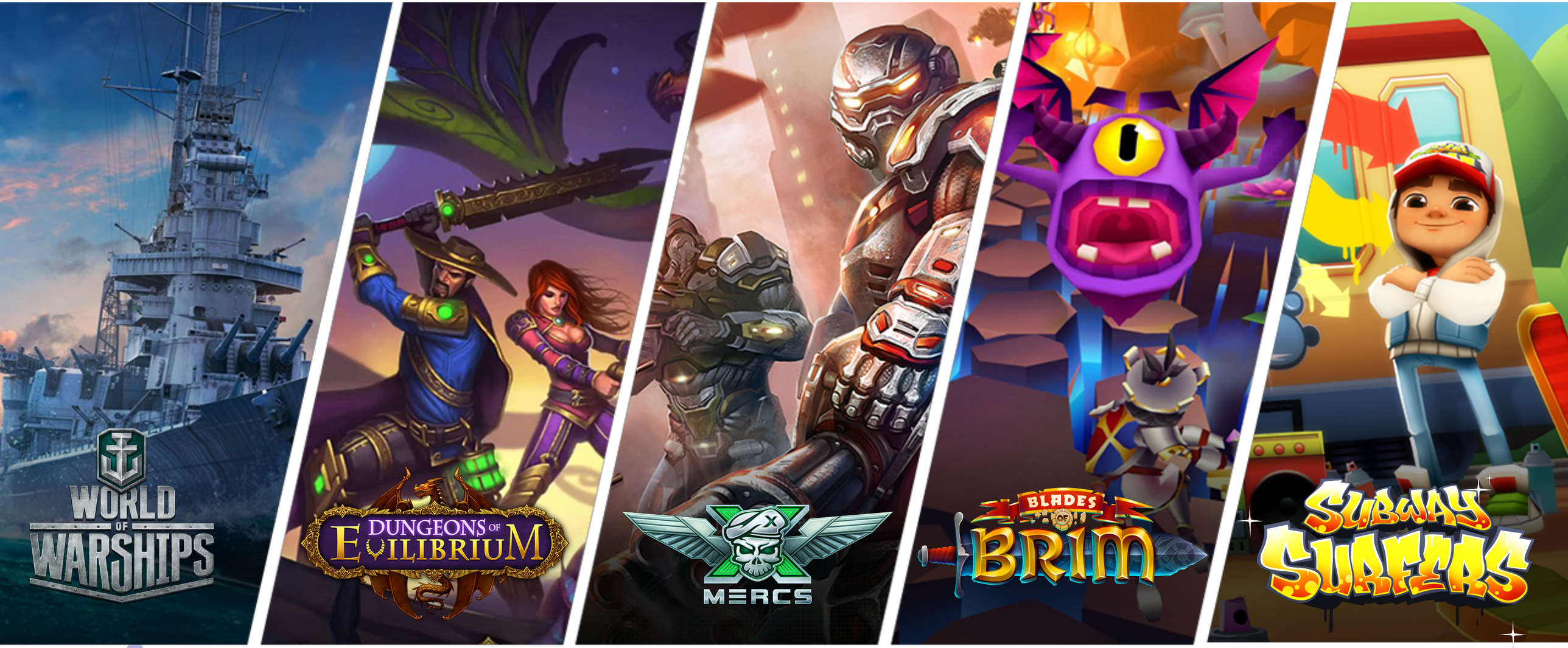
WARGAMING.NET
LET'S BATTLE

ZILLION WHALES

GameInsight

SYBO

SYBO



+ OTHER NON-RELEASED & UNANNOUNCED PROJECTS

GDC

GAME DEVELOPERS CONFERENCE
MARCH 18–22, 2019 | #GDC19

WHY BUILD A UI FRAMEWORK?

FRAMEWORKS

WEB

Multiple front-end frameworks
(Bootstrap, Material UI)

Multiple tools for easy design integration
(React, Framer)

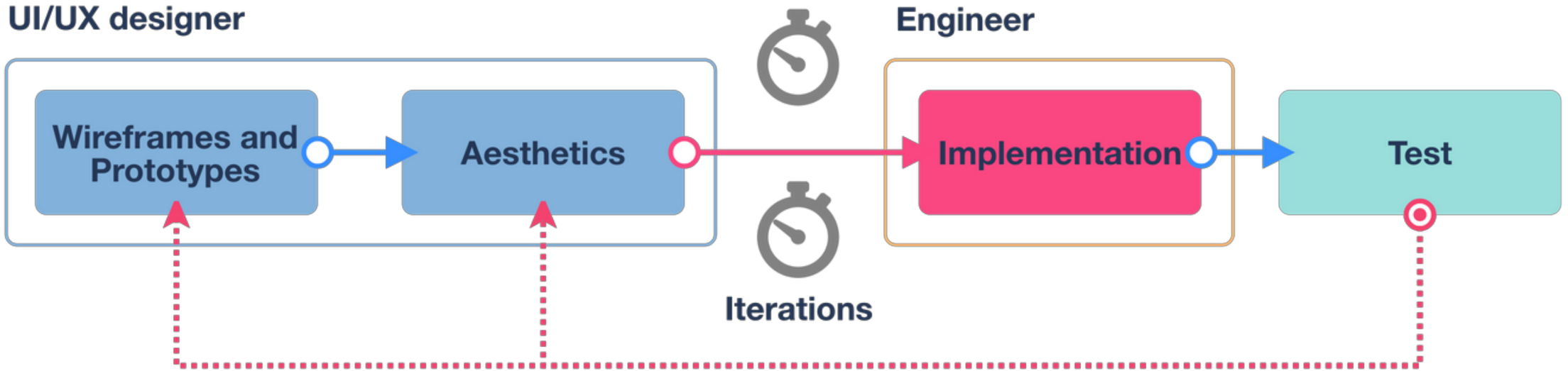
Games

Build-in solution requires programming knowledge for advanced use

3rd party solutions are usually not well supported or poorly adapted for mobile

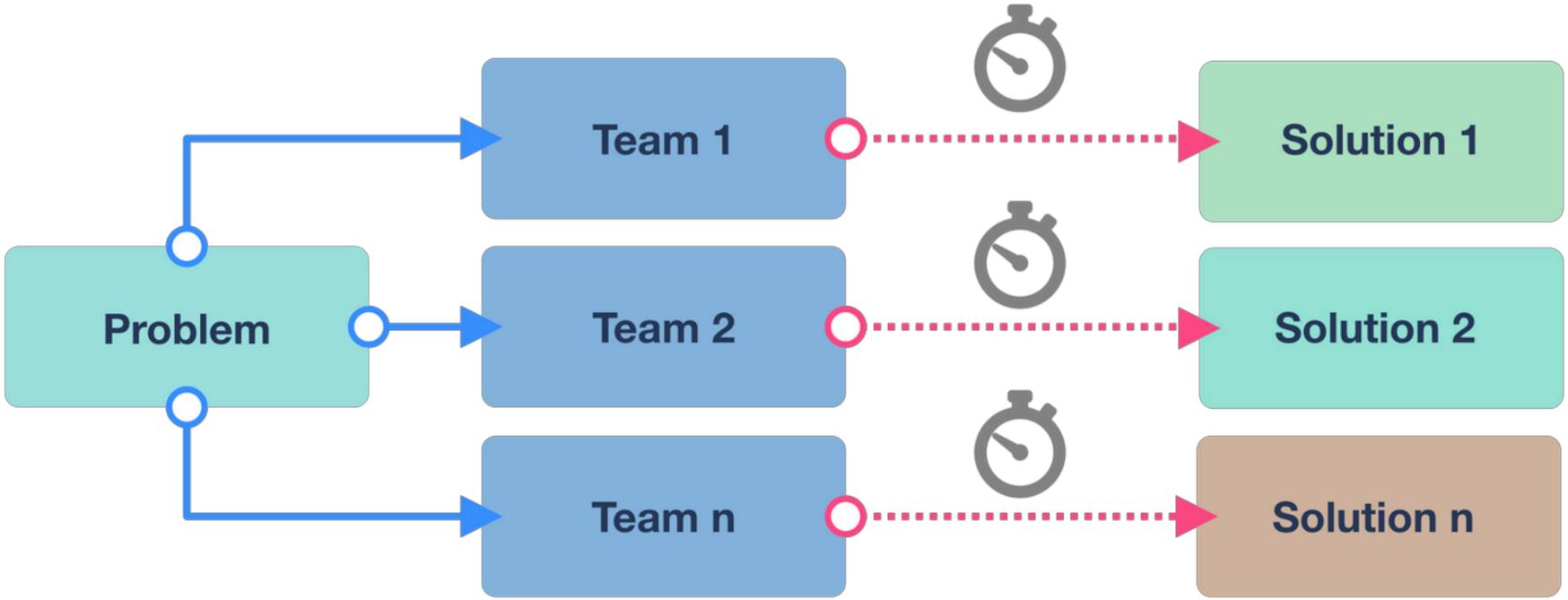
UNITY UI SYSTEM WAS NOT ENOUGH

DESIGNERS DIDN'T HAVE ENOUGH CONTROL OVER THE IMPLEMENTATION



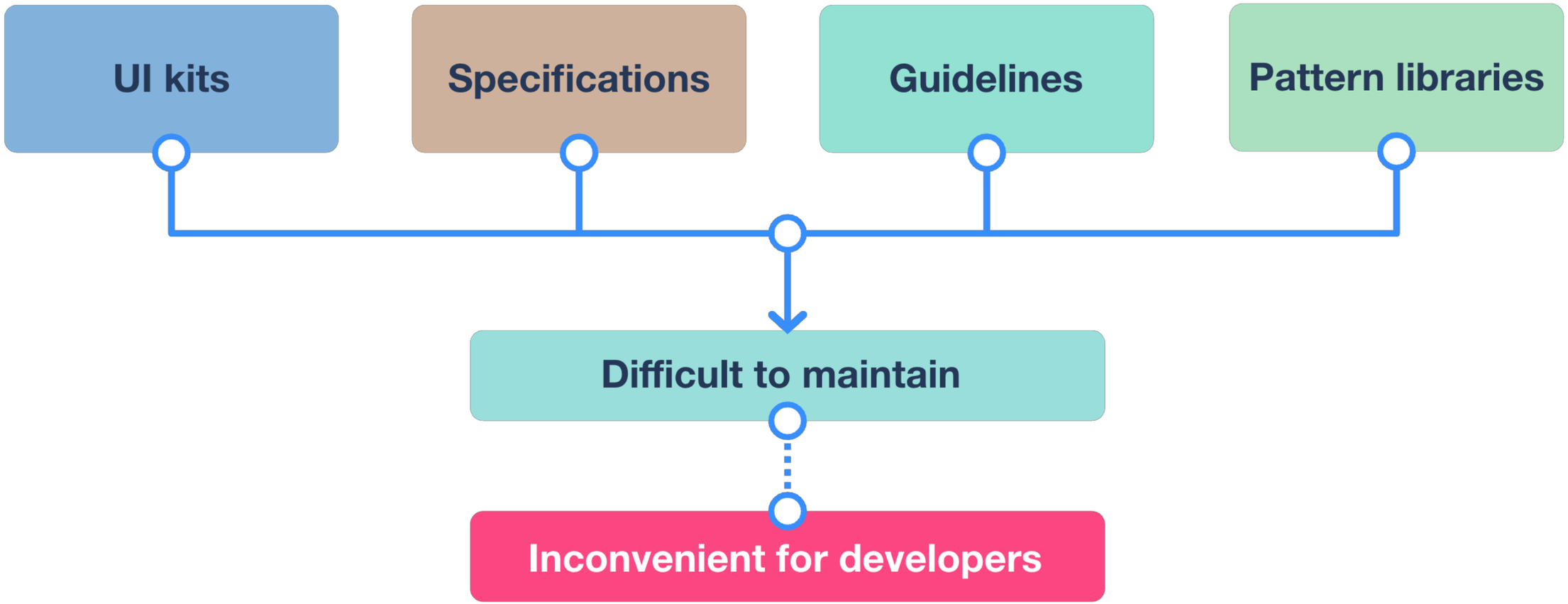
- Process of redesigning was slow and demanding
- Sometimes designs were implemented incorrectly
- Not unified technical solutions across the projects

MULTIPLE TEAMS HAD TO SOLVE SIMILAR PROBLEMS

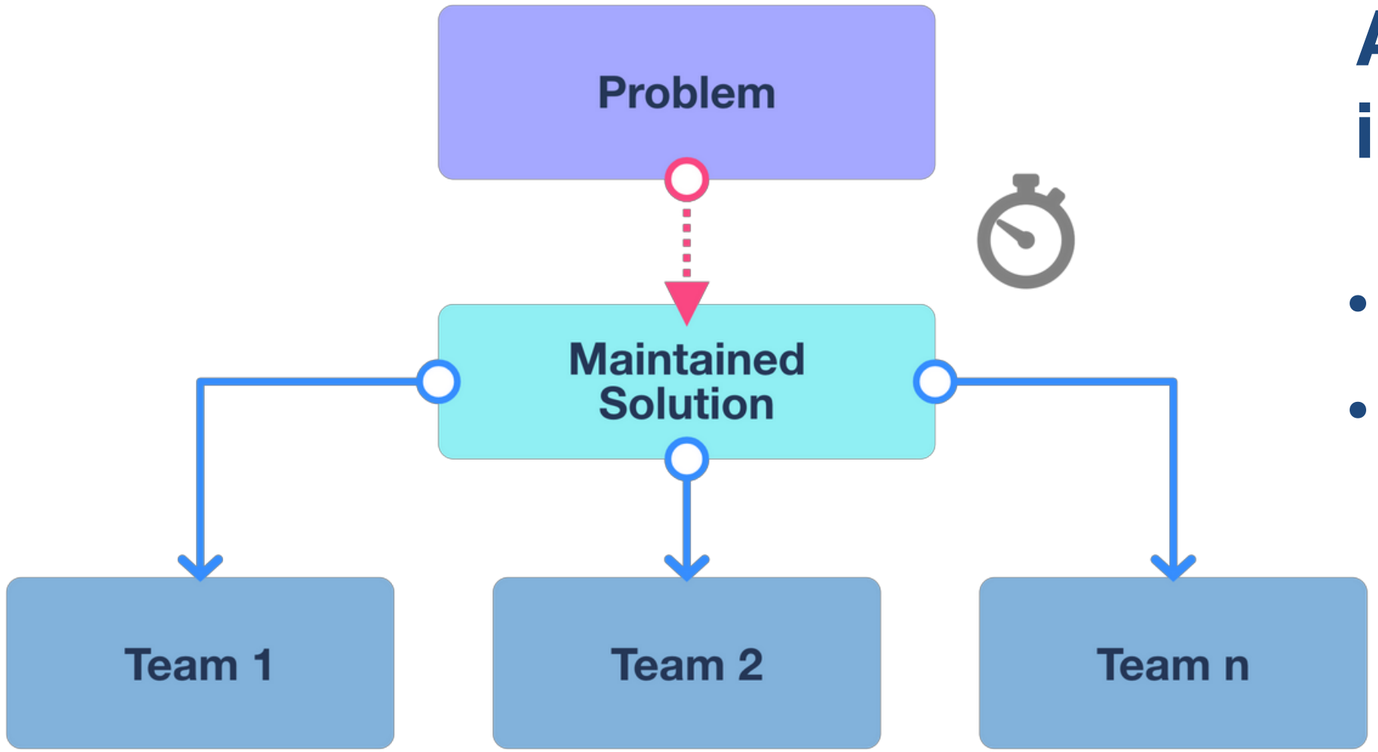


- Inconsistent solutions
- Time consuming
- Difficult to maintain
- Less flexible teams

OLD APPROACHES



UNIFIED UI ON IMPLEMENTATION LEVEL



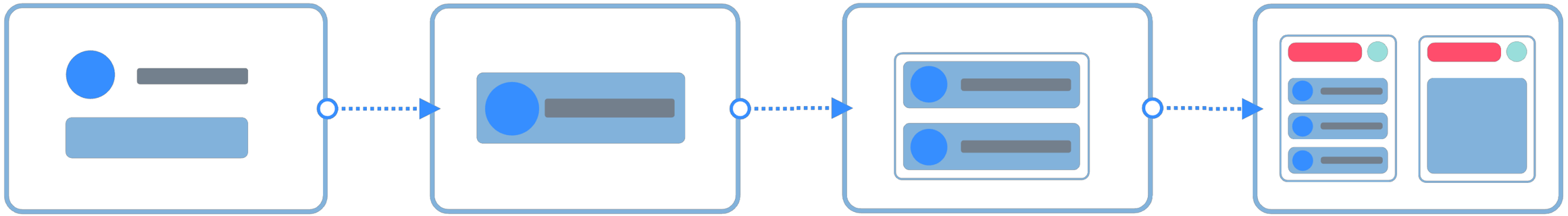
Accurate and reusable implementation

- Gives confidence in quality
- Easier to maintain

HOW TO APPROACH

1

PRINCIPLES AND COMPONENTS



Atoms

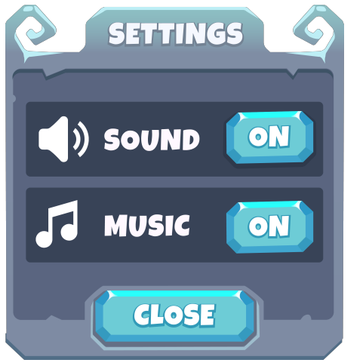
- Buttons
- Labels
- Images
- Transitions

Molecules

- Radio groups
- List items
- Scroll lists
- Tabs

Organisms

- Slice panels



Screens

- Views

UI FRAMEWORK FOR MULTIPLE PROJECTS

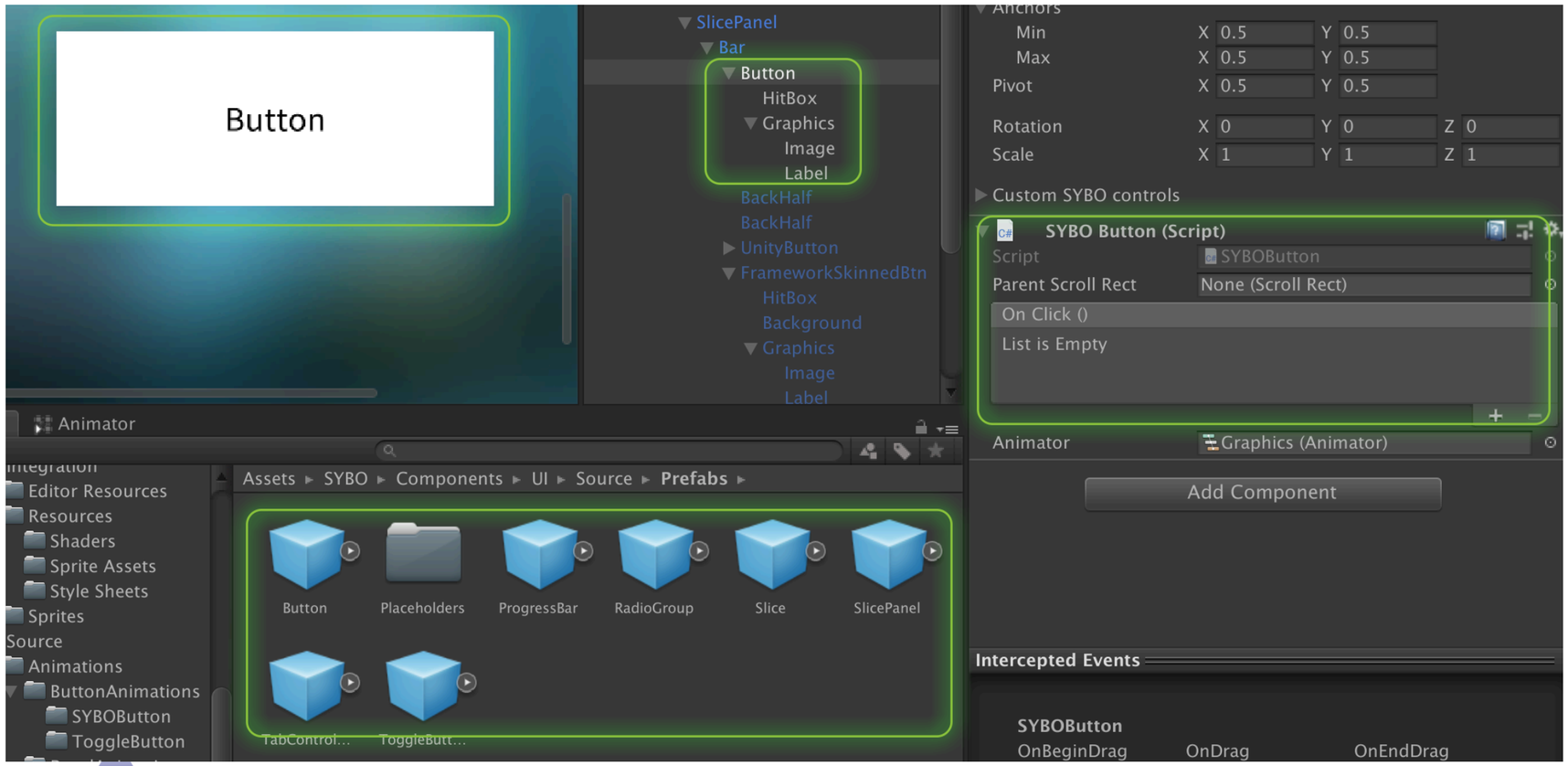
- **Technologically unified components**

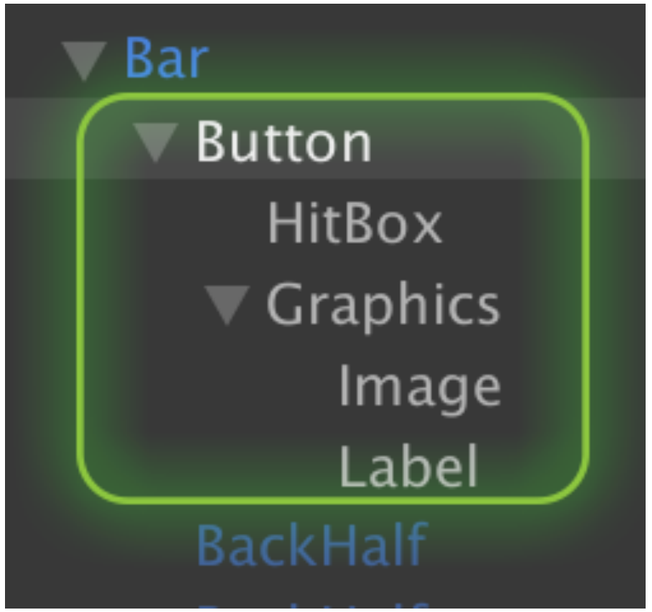
  That can be skinned and combined into templates, which then can be combined into screens

- **Reusable behavioural patterns and transitions**

- **Visual language is on the project level**

- Defined principles and components built as prefabs with minimum hierarchy for desirable functionality





EXPANDS UNITY UI

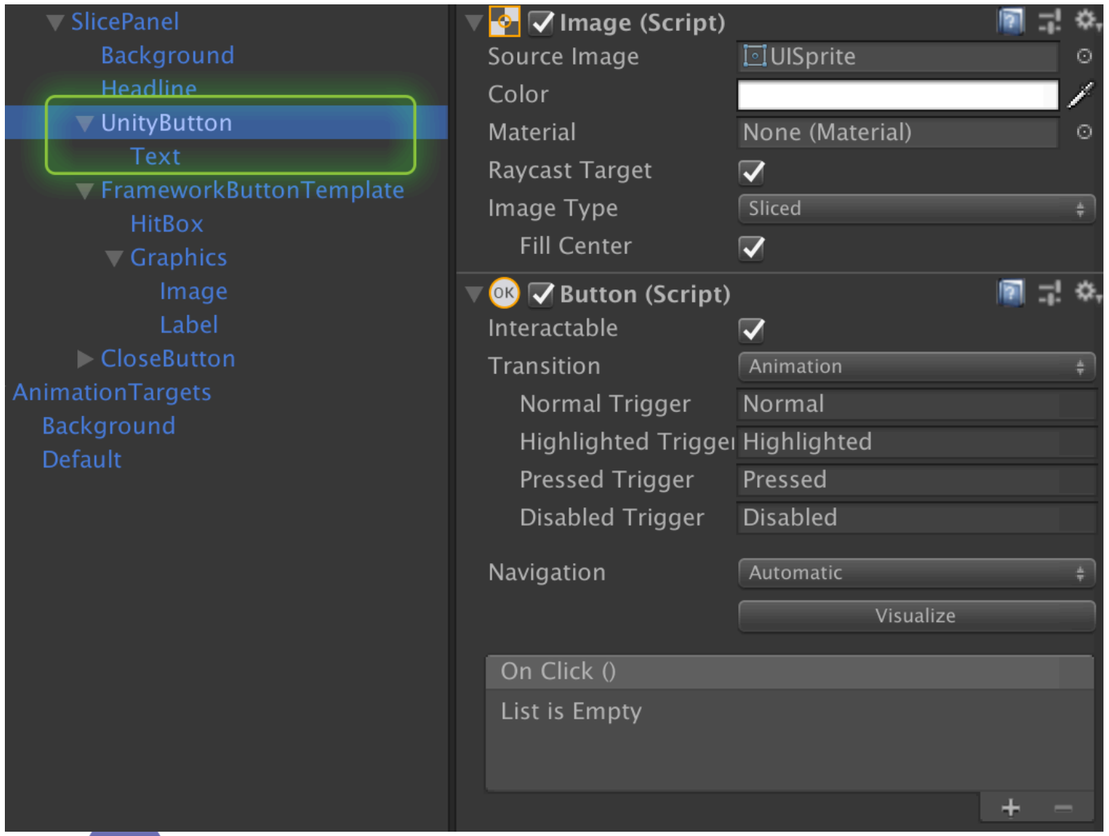
Unity UI system

- Some components are not accommodated well for mobile
- Sometimes provides unnecessary functionality and doesn't provide what is needed
- Requires programming knowledge for advanced use

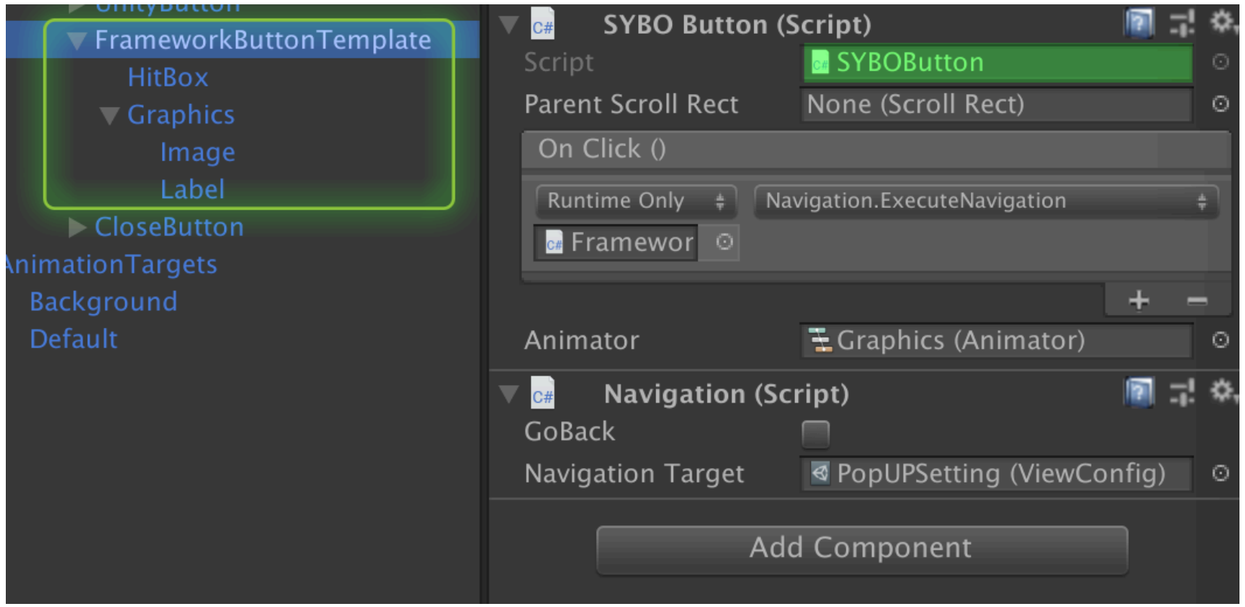
EXAMPLE: FRAMEWORK BUTTON

- Improved structure for providing feedback, more thorough hierarchy

Unity Button



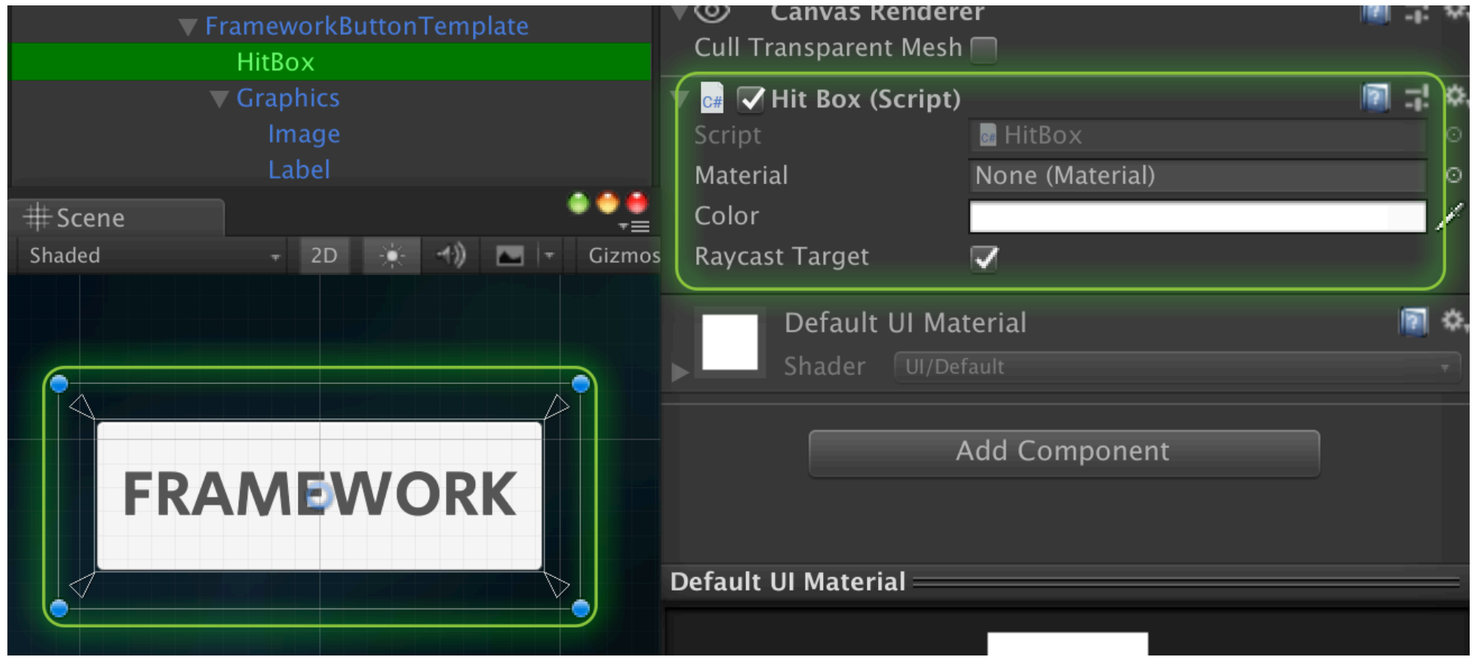
Framework Button

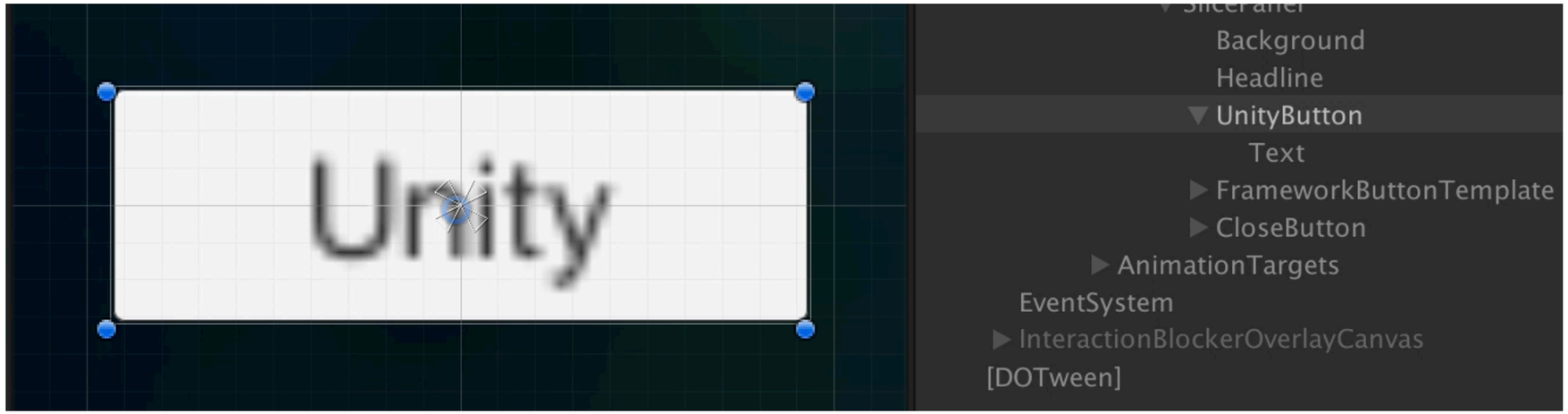


Has a **HitBox** child, that defines the interaction area for the button

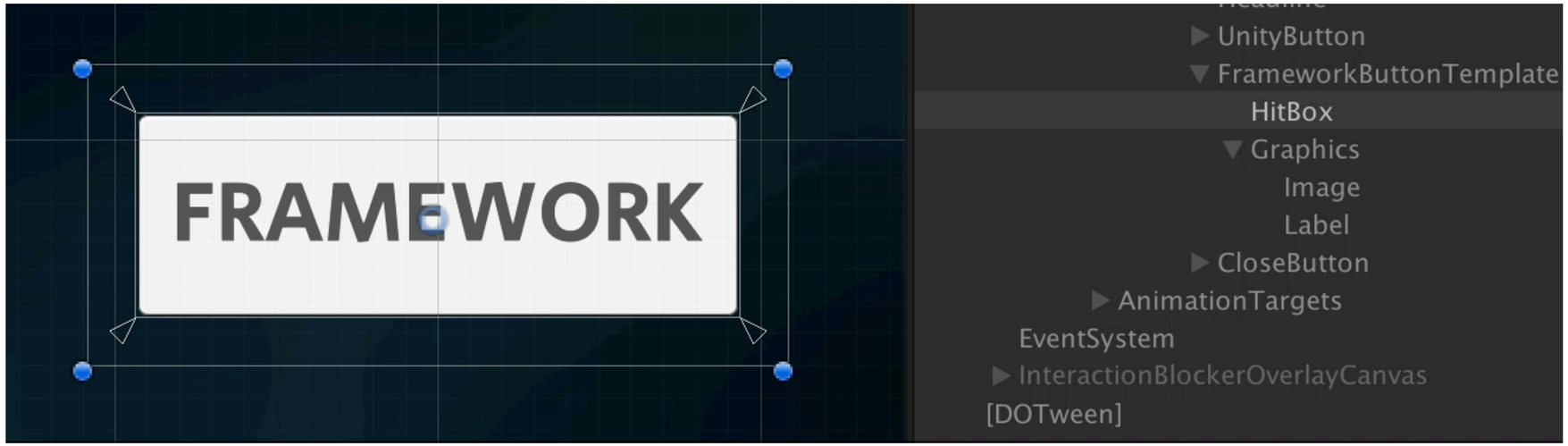
It is placed above the animated part and can be bigger than the components graphics, so it prevents the situation of click not being registered:

- When Interacting with small-sized elements,
- Due to scaling during animation (pressed state)



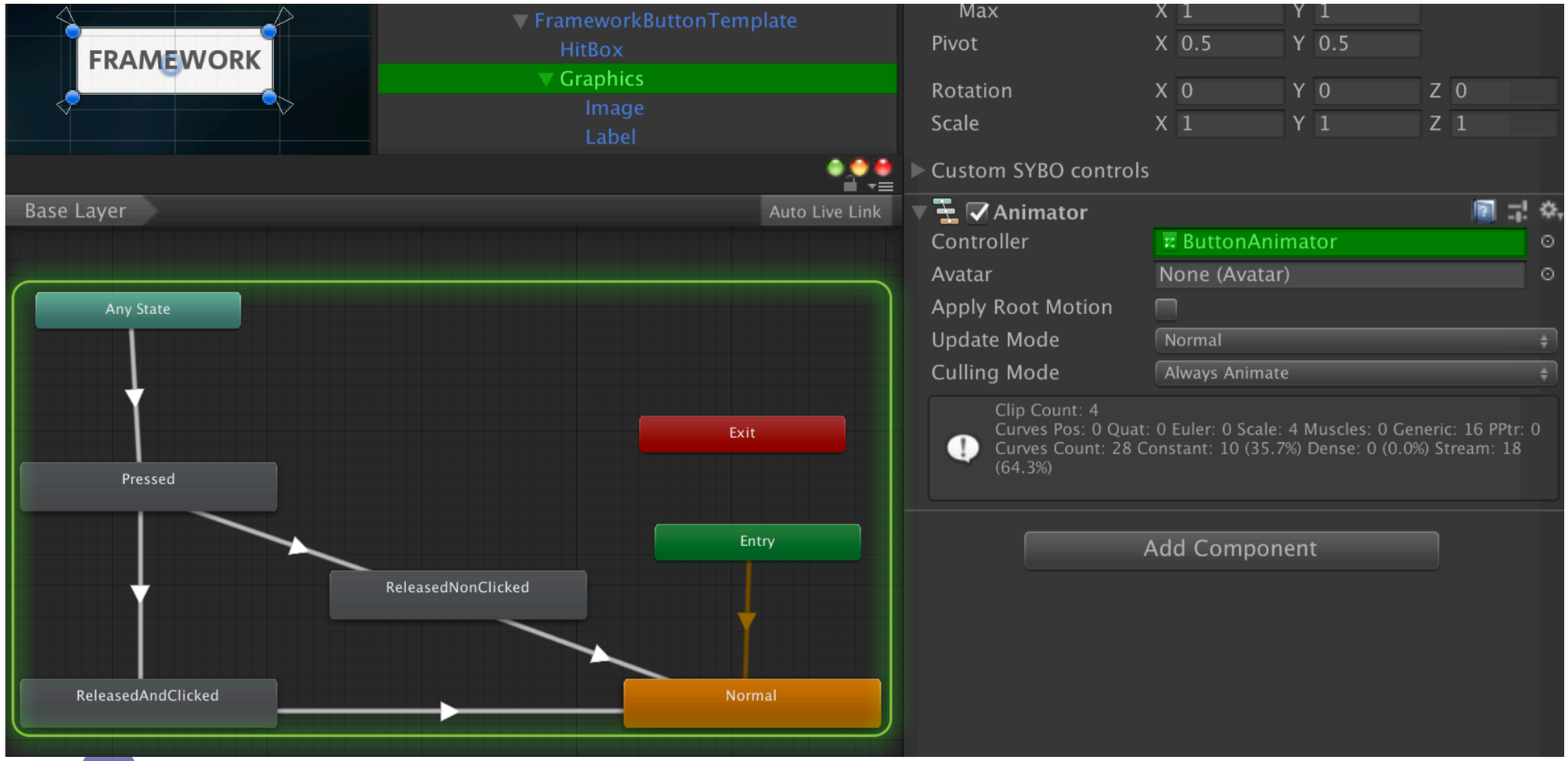


Unity collider scales with the button

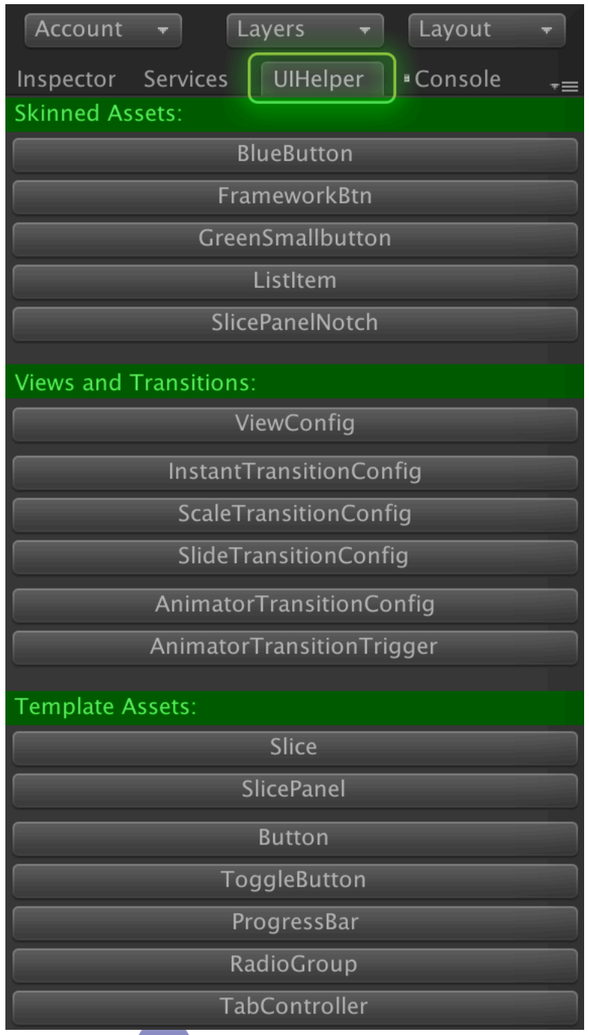


HitBox doesn’t scale with the graphics

- Feedback provided by animators
- Animators can be easily re-used across the buttons.



TEMPLATE AND SKINNED ASSETS



- **Skinned Assets**

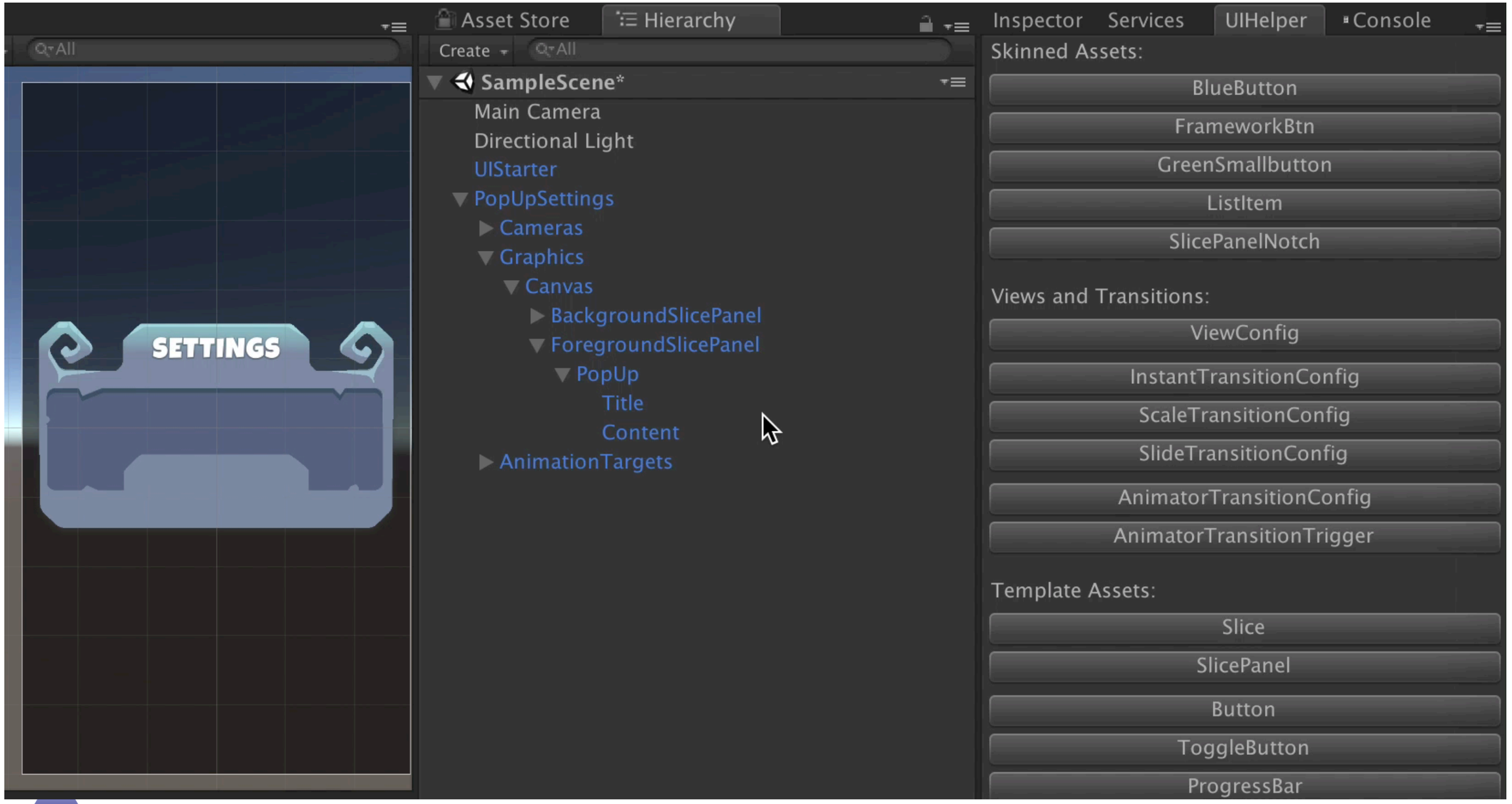
  Project-specific skinned components, prepared for immediate instantiation

- **Views and Transitions**

  Basic transitions

- **Template Assets**

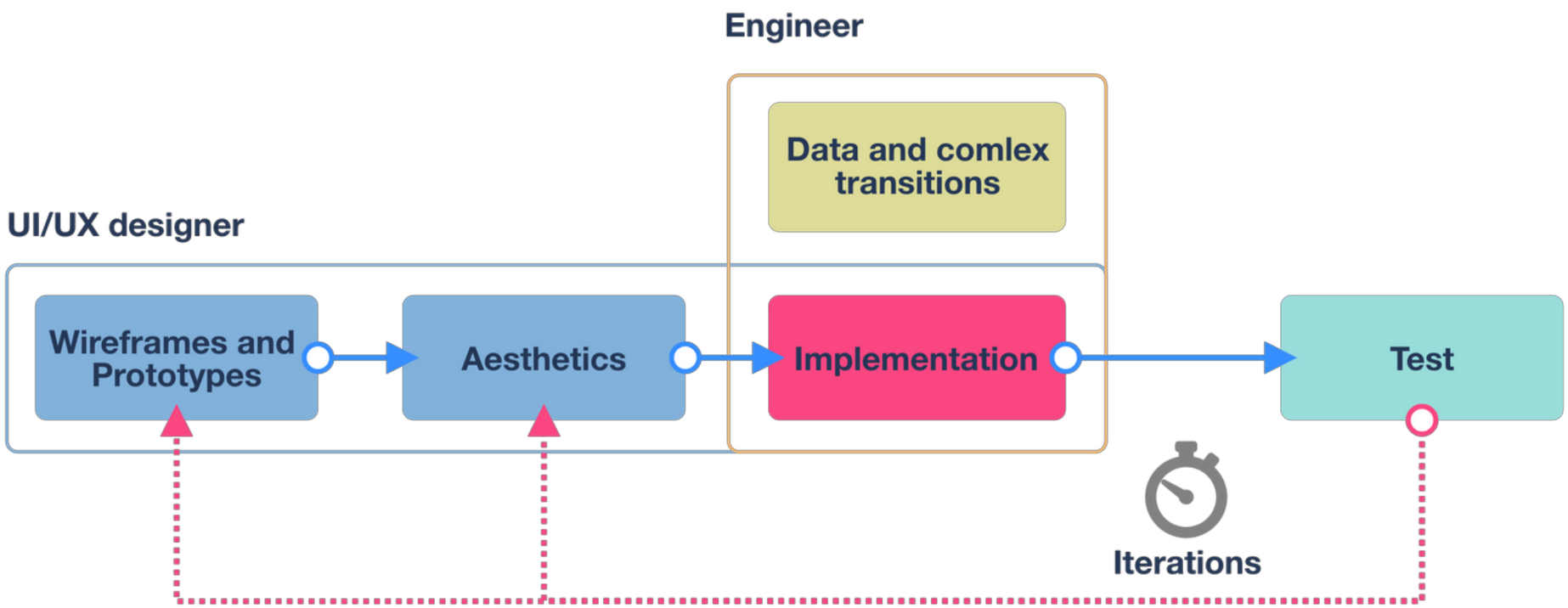
  Templates of basic components provided by UI framework


Asset Store
Hierarchy
Inspector
Services
UIHelper
Console
All
Create
All
SampleScene*
Main Camera
Directional Light
UIStarter
PopUpSettings
Cameras
Graphics
Canvas
BackgroundSlicePanel
ForegroundSlicePanel
PopUp
Title
Content
AnimationTargets
SETTINGS
Skinned Assets:
BlueButton
FrameworkBtn
GreenSmallbutton
ListItem
SlicePanelNotch
Views and Transitions:
ViewConfig
InstantTransitionConfig
ScaleTransitionConfig
SlideTransitionConfig
AnimatorTransitionConfig
AnimatorTransitionTrigger
Template Assets:
Slice
SlicePanel
Button
ToggleButton
ProgressBar

2

PROVIDE FREEDOM FOR NON-ENGINEERS

WORKFLOW WITH UI FRAMEWORK



Freedom in implementation without a big necessity to code

=

Significant reduction of the iteration time between designer and engineers

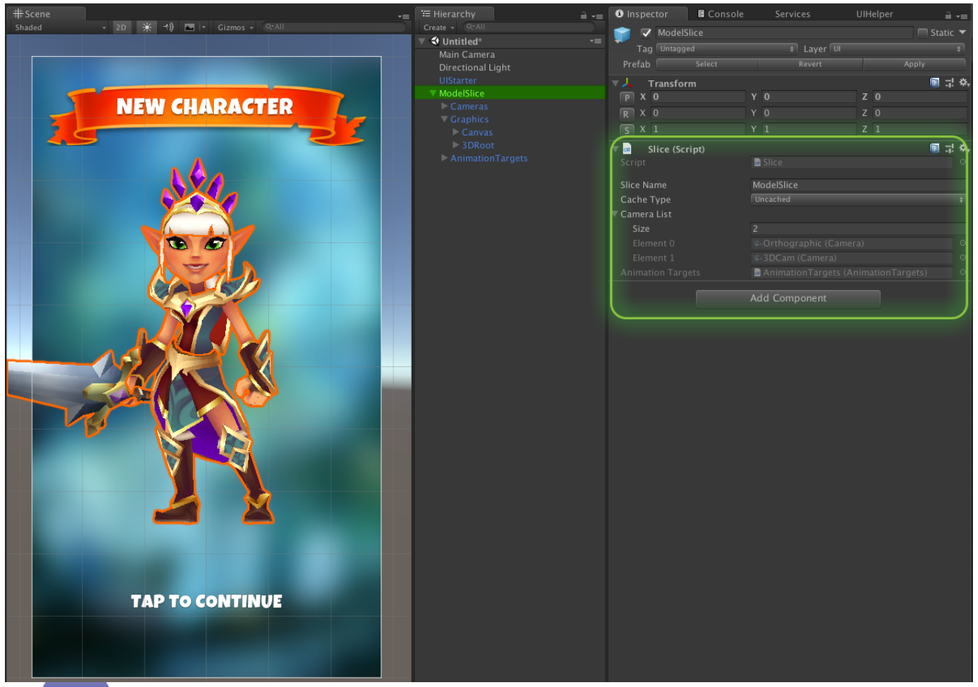
MAIN COMPLEX AREAS:

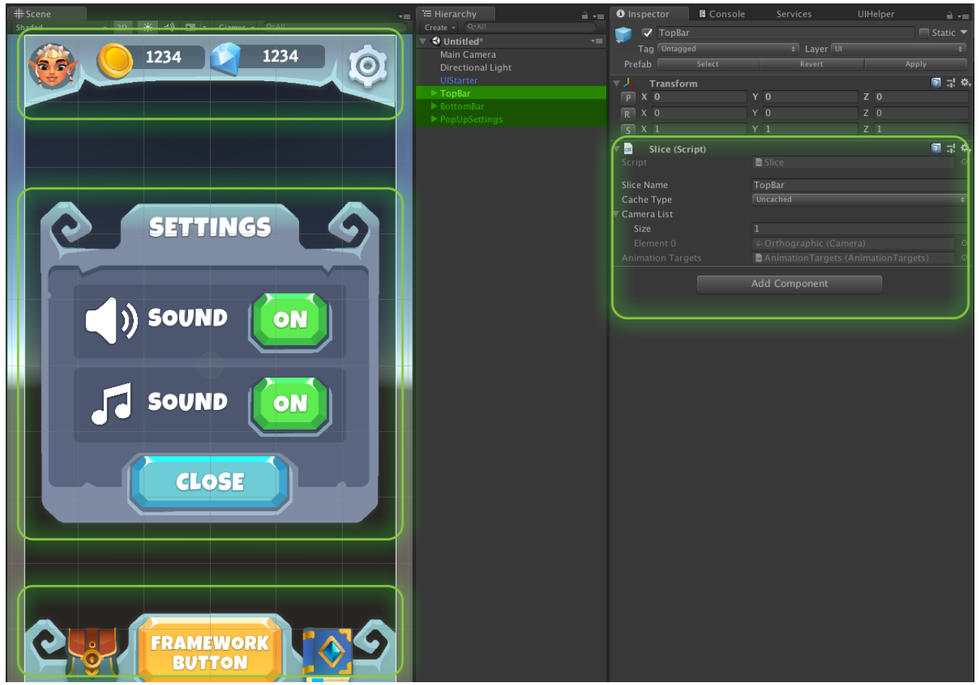
- Navigation and transitions between screens
- Controlling the rendering order of multiple layers, including 3D

SLICE UI COMPONENT

Serve as containers of canvases, graphic assets and cameras

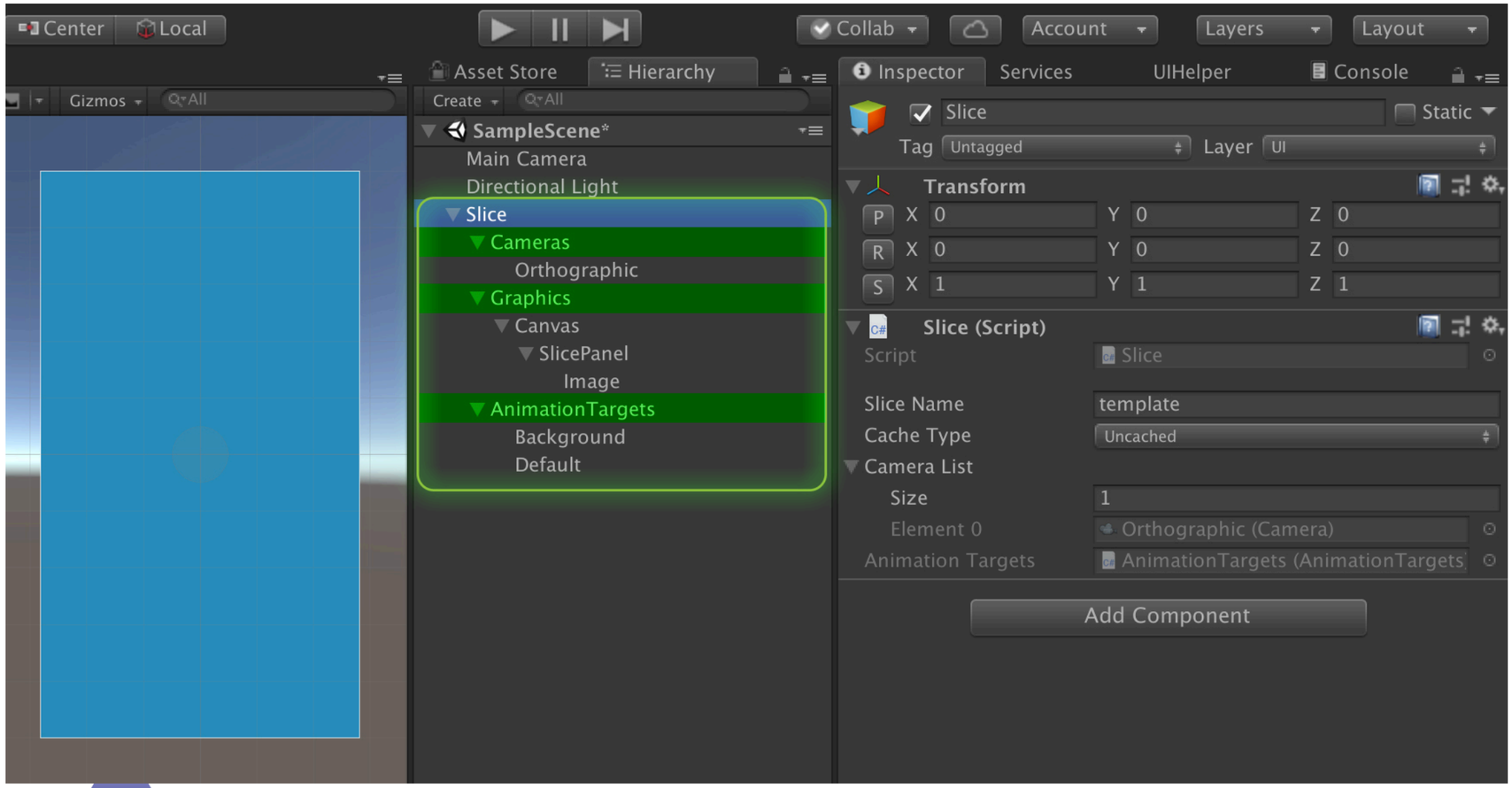
- Can be built as an entire screen (loading screen, 3D model)
- Can be built as a part of the screen (navigation menus, pop-ups, etc.)





FRAMEWORK SLICE TEMPLATE

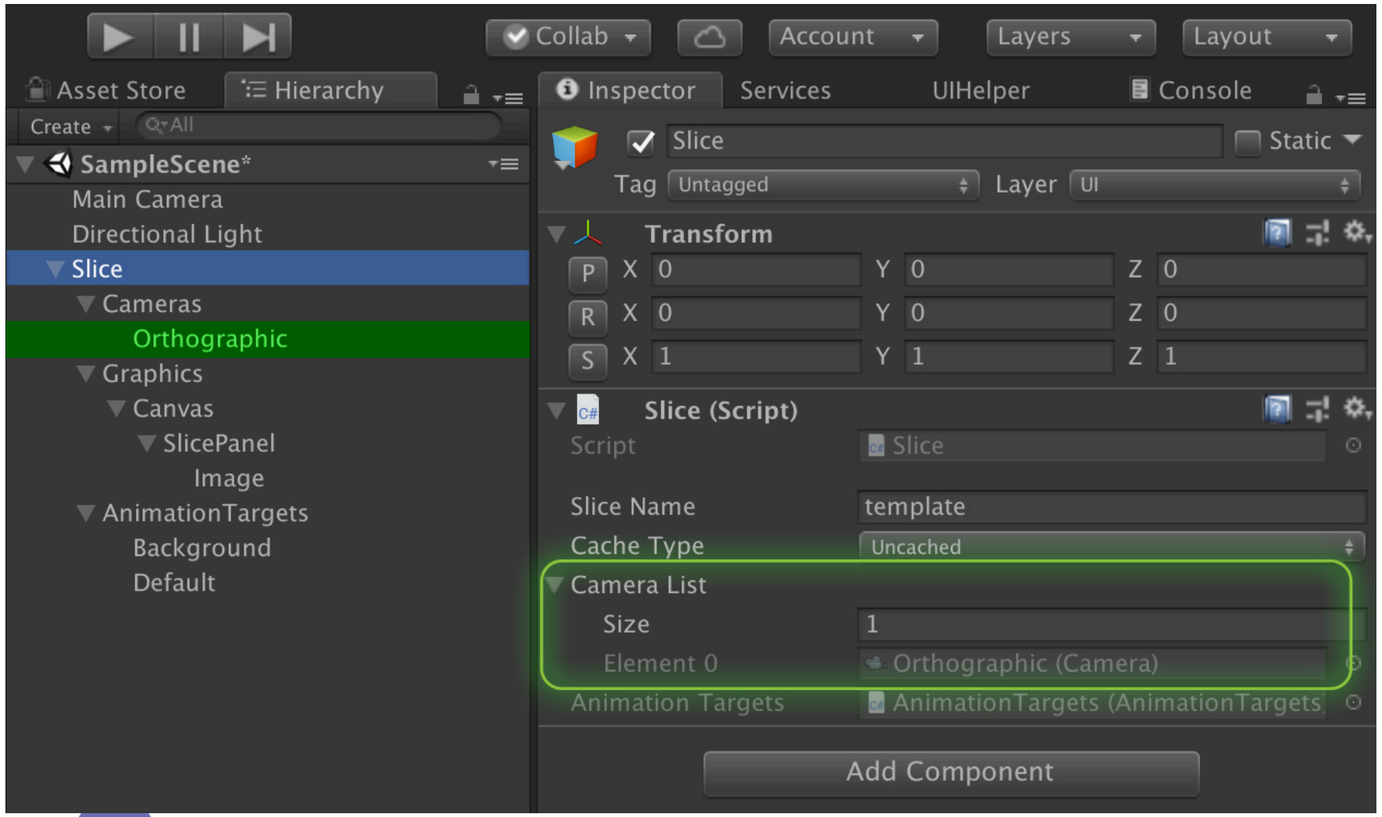
Consists of a root object with 3 main children:



- Cameras
- Graphics
- Animation Targets

CAMERAS

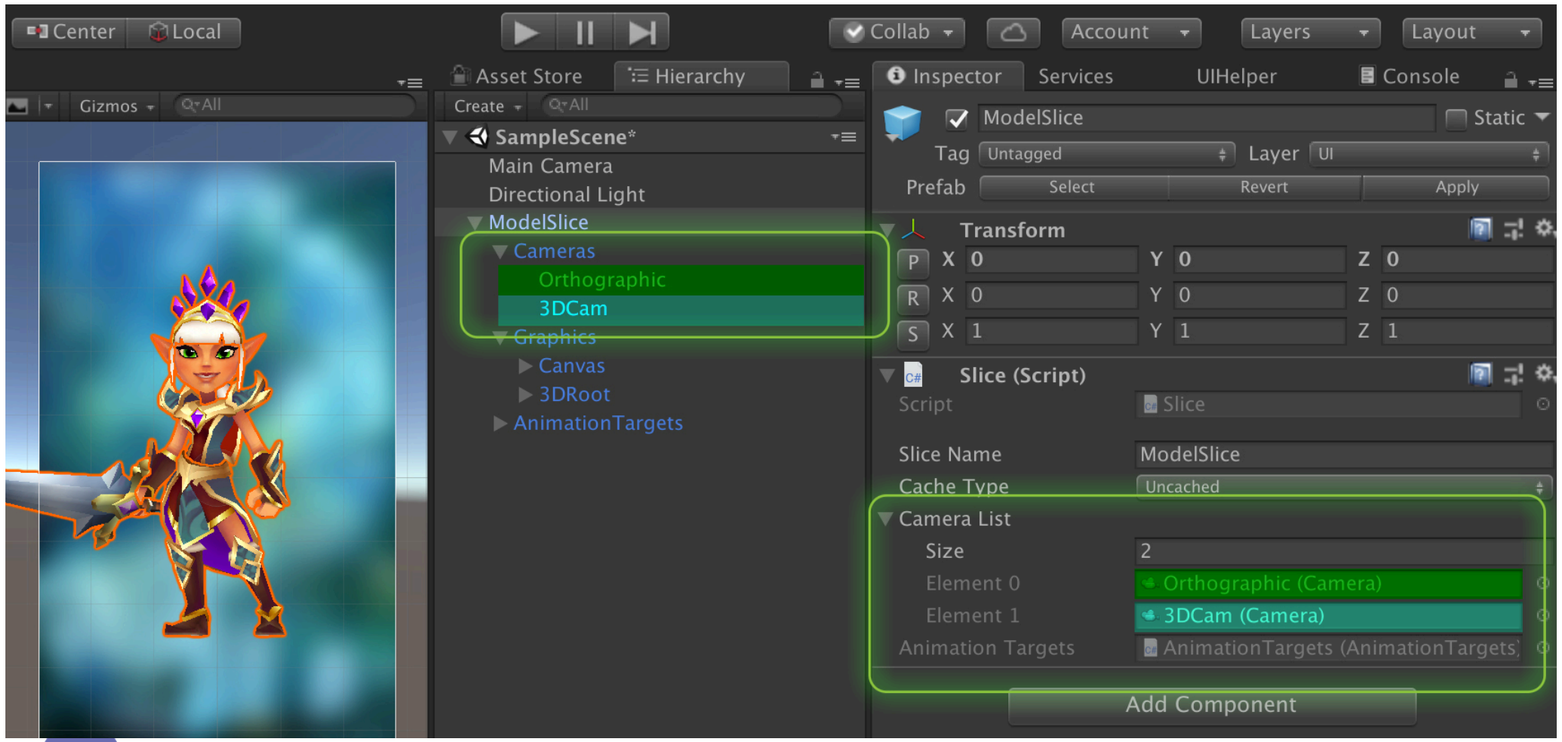
The slice template comes with 1 orthographic camera per default (most commonly used camera for 2D graphics)



Supports as many cameras as needed

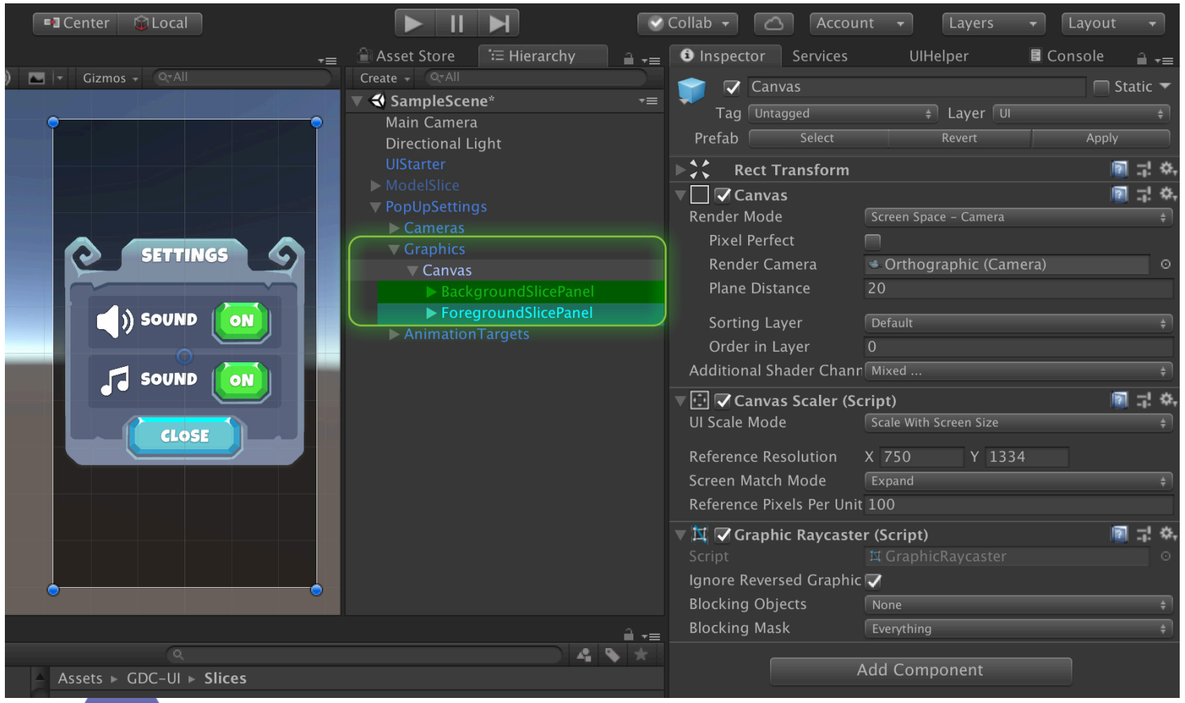
- Orthographic (2D)
- Perspective (3D)

- Added cameras become automatically referenced on the Slice component in the field called Camera List
- The order of the cameras determines the rendering order where last child camera is rendered topmost



GRAPHICS

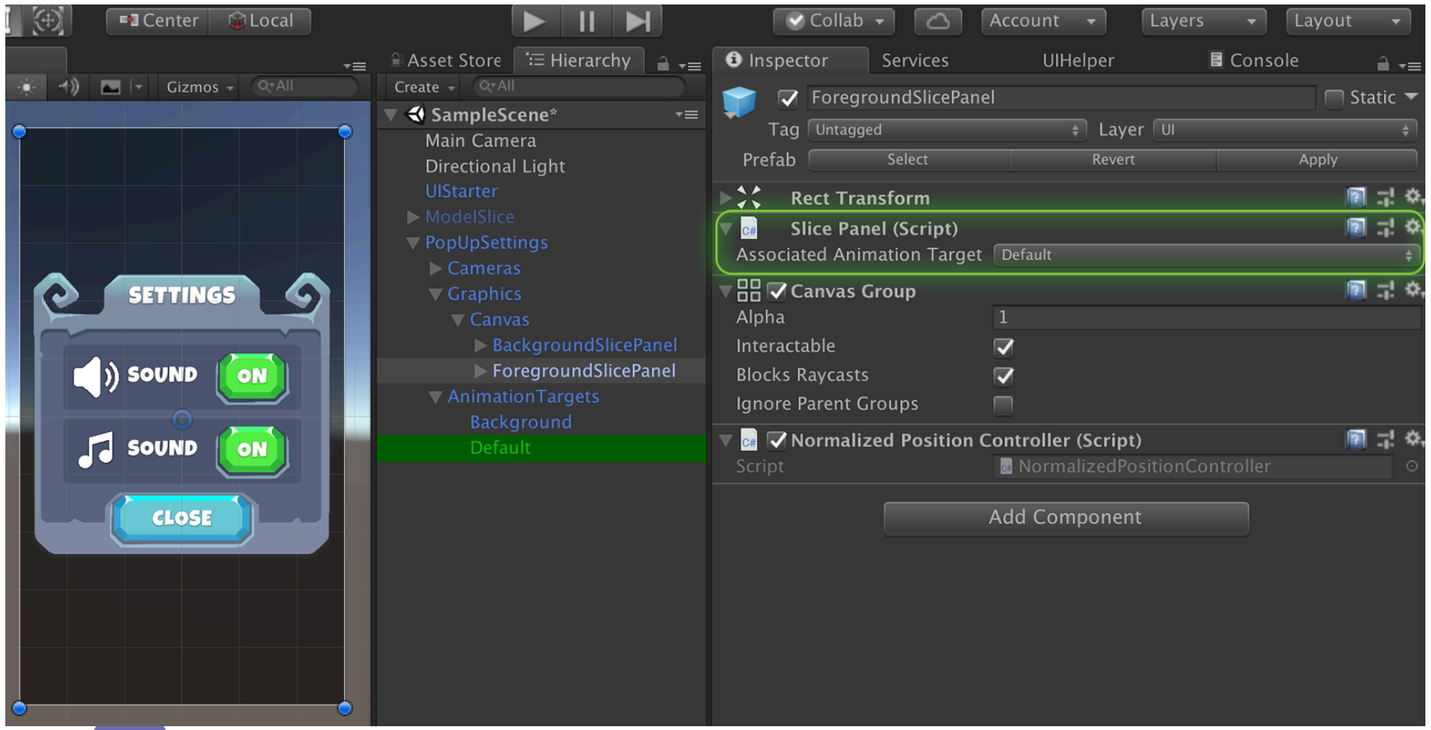
- Acts as a parent for all the graphics contained in the slice.
- Children of the Graphics object are expected to be **canvases** (where 2D gets rendered) with an associated camera for each canvas.



- Children of the canvases are expected to be **Slice Panels**

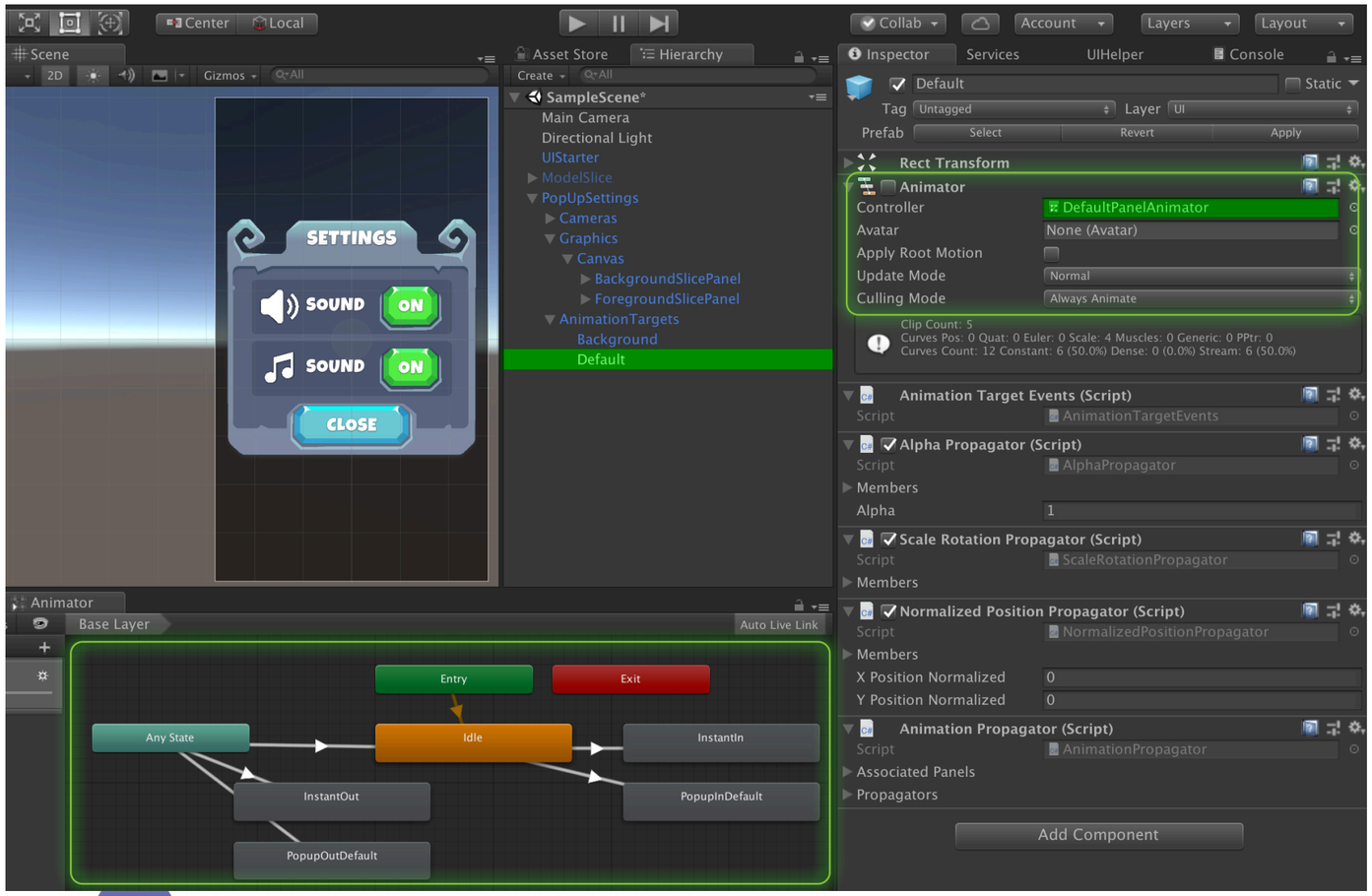
SLICE PANELS

- Parents of graphics assets for the slice.
- Similar to the generic Unity panel, but has an added component that allows to associate this panel with an **Animation Target**



ANIMATION TARGETS

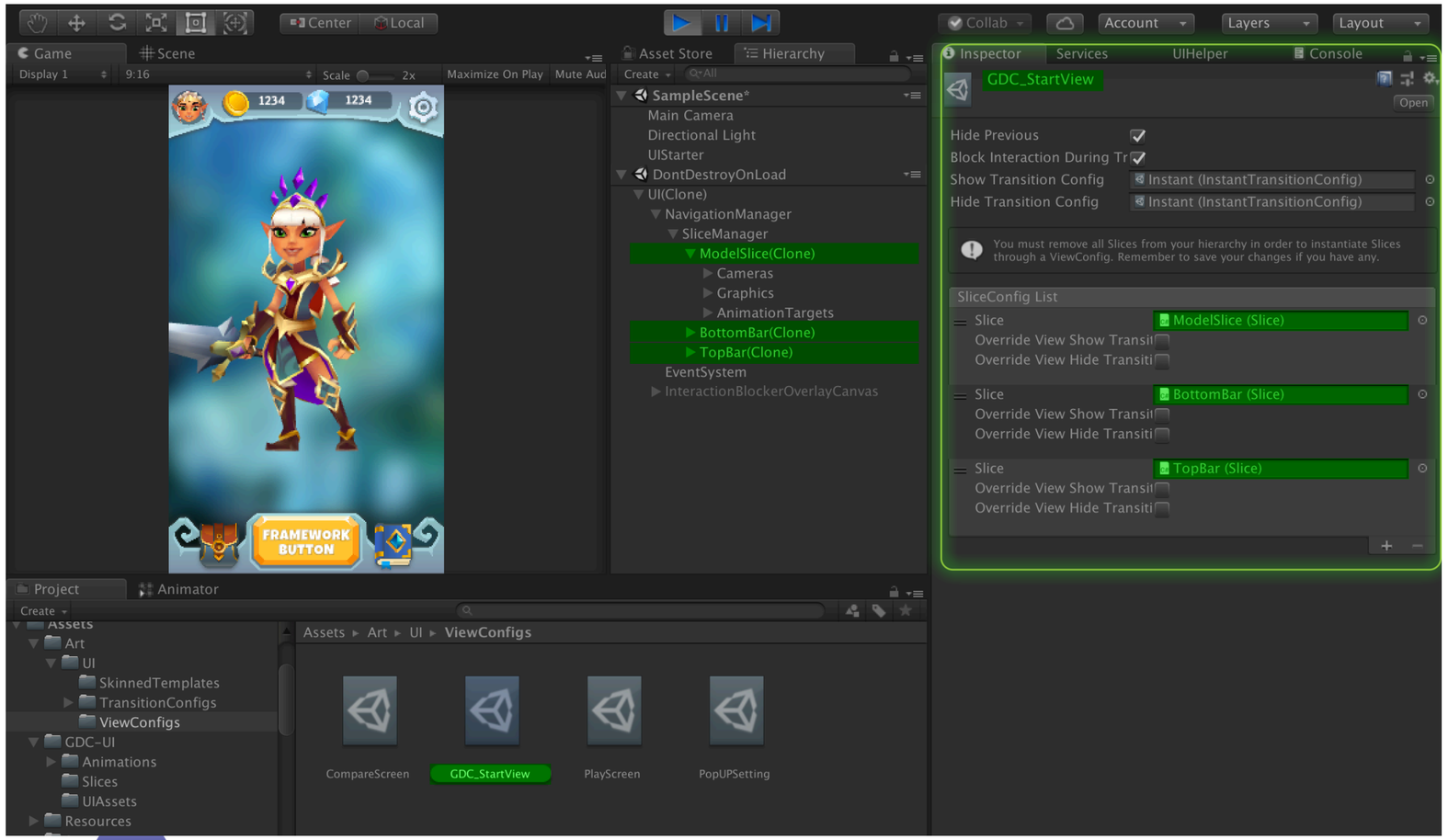
- Propagate animations for our Slice Panels



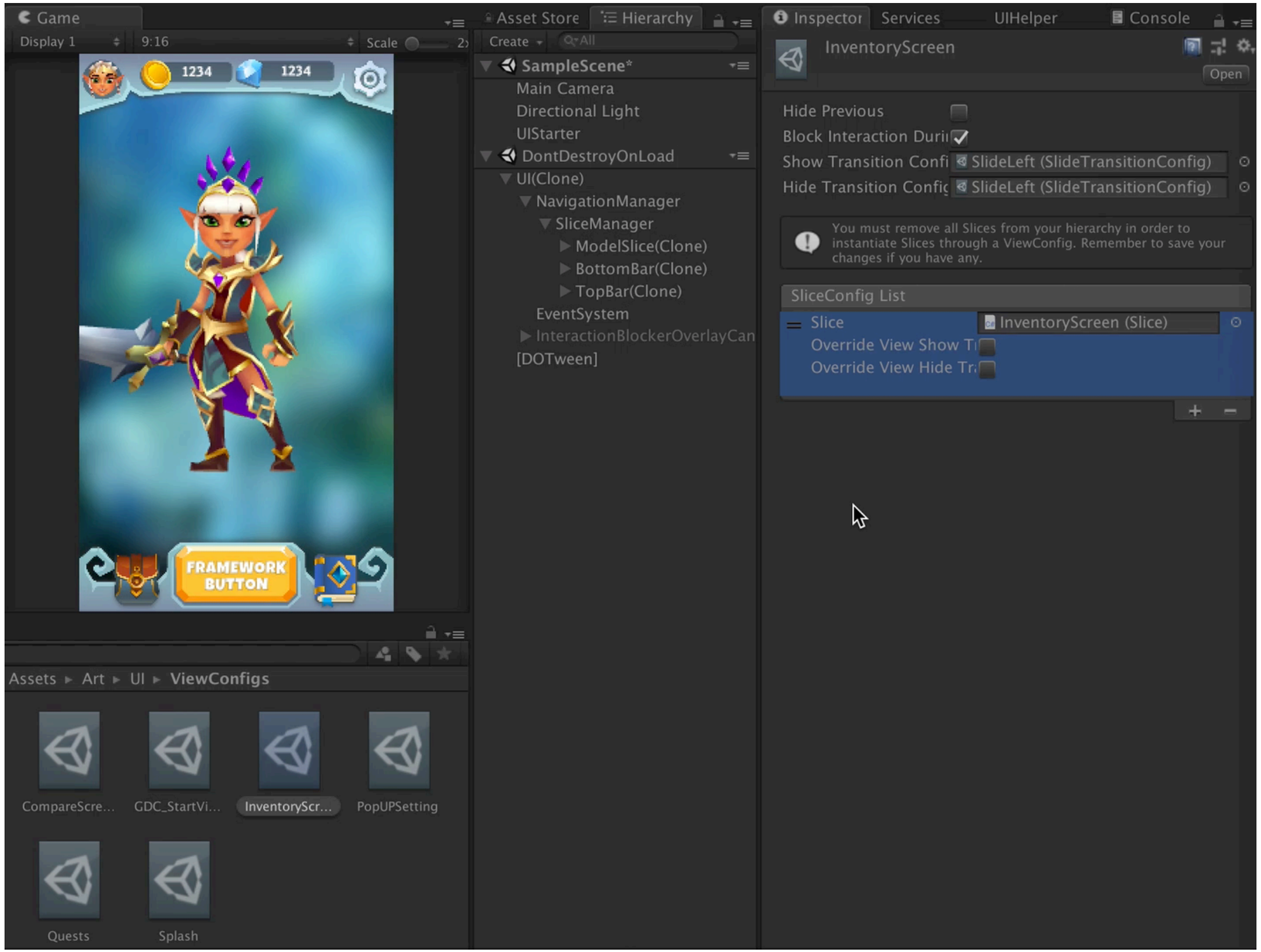
Allow to re-use animations across the slices in our projects, **independent of amount of children and naming**

VIEW CONFIG

A “recipe” for the UI system about how to build and present a view



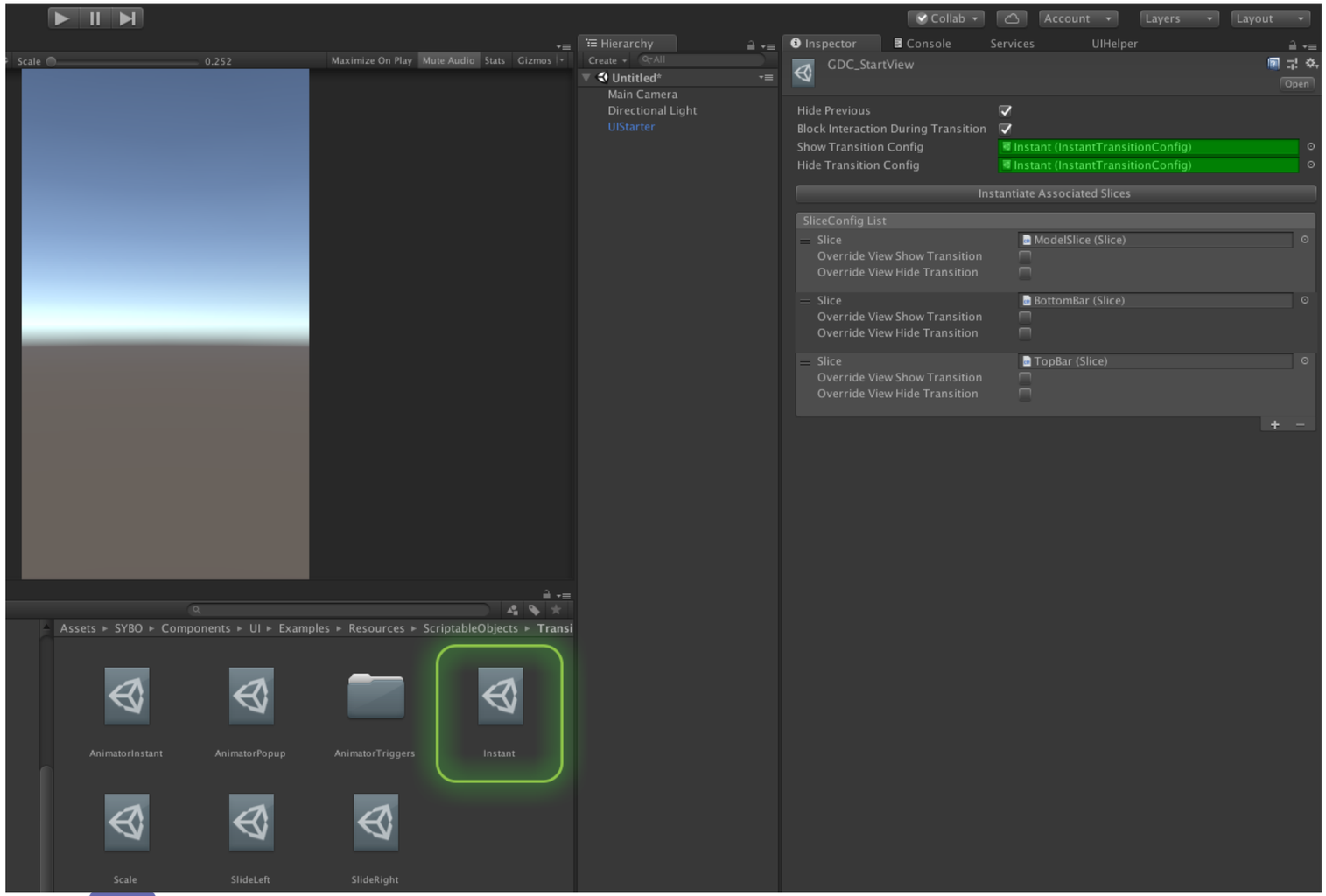
- A **View** can include multiple slices if desired
- **Slices** can also be shared across multiple views
- Easy to add a new slice by assigning it in the Slice field


Game
Display 1
9:16
Scale
1234
1234
FRAMEWORK BUTTON
Asset Store
Hierarchy
Create
SampleScene*
Main Camera
Directional Light
UIStarter
DontDestroyOnLoad
UI(Clone)
NavigationManager
SliceManager
ModelSlice(Clone)
BottomBar(Clone)
TopBar(Clone)
EventSystem
InteractionBlockerOverlayCan
[DOTween]
Inspector
Services
UIHelper
Console
InventoryScreen
Open
Hide Previous
Block Interaction Duri
Show Transition Confi
SlideLeft (SlideTransitionConfig)
Hide Transition Confi
SlideLeft (SlideTransitionConfig)
You must remove all Slices from your hierarchy in order to instantiate Slices through a ViewConfig. Remember to save your changes if you have any.
SliceConfig List
Slice
InventoryScreen (Slice)
Override View Show Tr
Override View Hide Tra
Assets ▸ Art ▸ UI ▸ ViewConfigs
CompareScre...
GDC_StartVi...
InventoryScr...
PopUPSetting
Quests
Splash

BENEFITS OF THE VIEW CONFIG

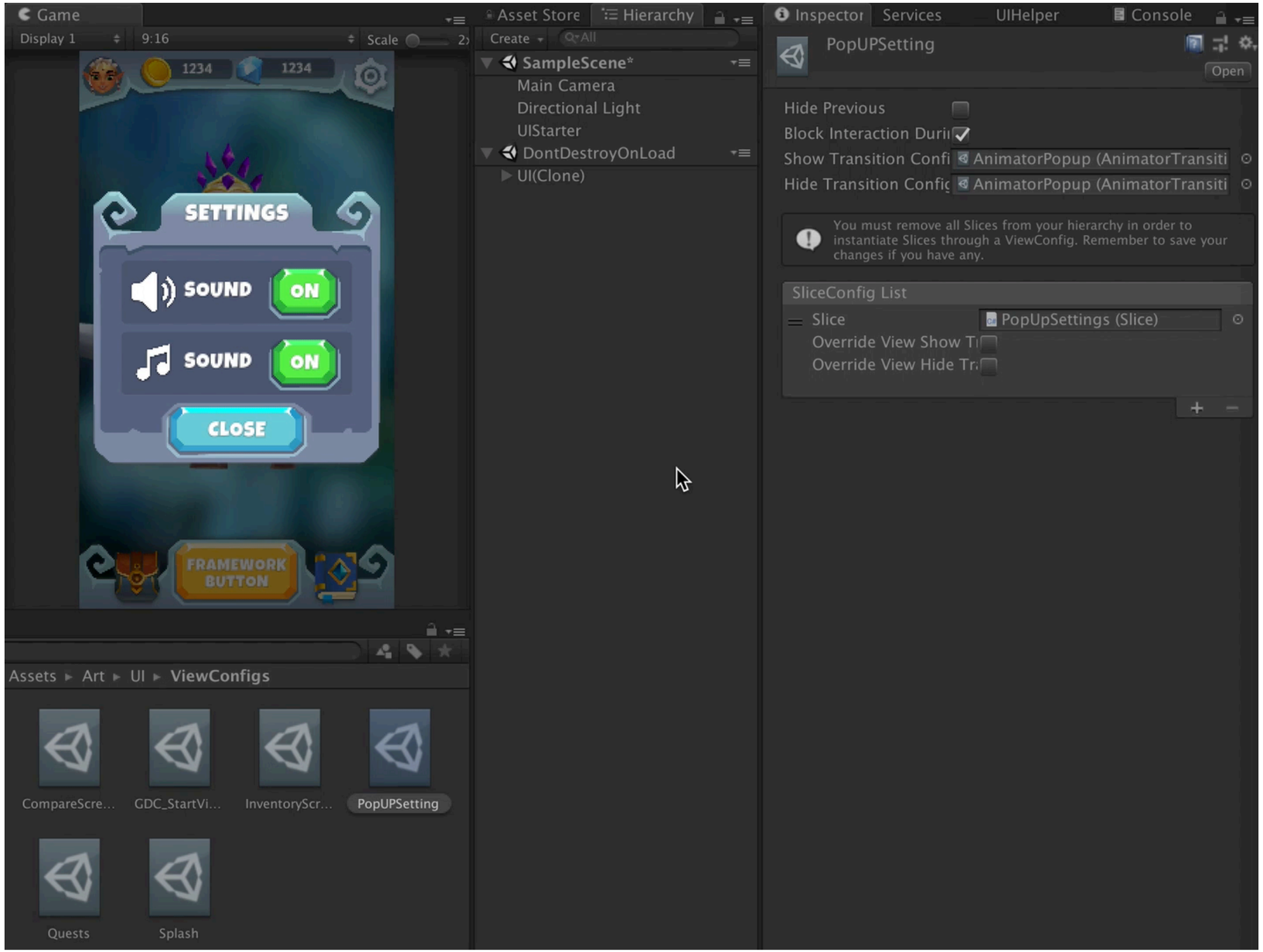
- Re-design and iterations become extremely quick. Combining different view-setup is a matter of seconds
- The order of slices in the View config defines rendering order
- Easy to hide or show previous views

TRANSITION CONFIG



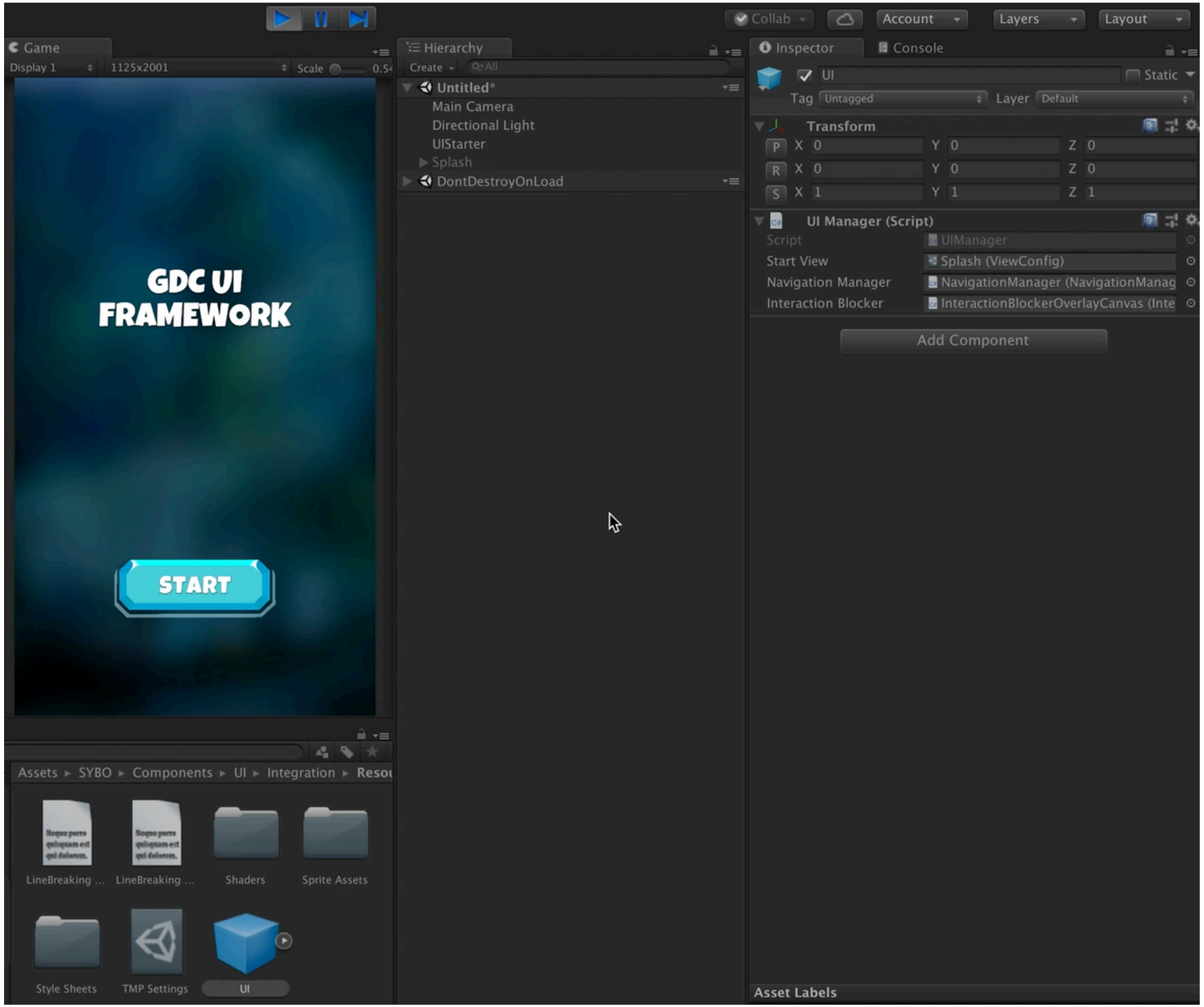
UI framework provides transition examples for different configurations:

- Scale
- Slide
- Instant
- Custom animation


Game
Display 1
9:16
Scale
1234
1234
SETTINGS
SOUND
ON
SOUND
ON
CLOSE
FRAMEWORK BUTTON
Asset Store
Hierarchy
Create
SampleScene*
Main Camera
Directional Light
UIStarter
DontDestroyOnLoad
UI(Clone)
Inspector
Services
UIHelper
Console
PopUPSetting
Open
Hide Previous
Block Interaction Duri
Show Transition Confi
AnimatorPopup (AnimatorTransiti
Hide Transition Confi
AnimatorPopup (AnimatorTransiti
You must remove all Slices from your hierarchy in order to instantiate Slices through a ViewConfig. Remember to save your changes if you have any.
SliceConfig List
Slice
PopUpSettings (Slice)
Override View Show T
Override View Hide Tr
Assets ▸ Art ▸ UI ▸ ViewConfigs
CompareScre...
GDC_StartVi...
InventoryScr...
PopUPSetting
Quests
Splash

BENEFITS OF THE TRANSITION CONFIG

- Operates with the whole slice panel
- Can be re-used across the projects. Can be re-used across views with the similar desirable behaviour, saving developers time and providing consistency
- Allow to create different combinations of animations for multiple slice panels in one slice
- Transitions for any slice in the view can be overridden with a specific transition, without affecting the rest


Collab
Account
Layers
Layout
Game
Display 1
1125x2001
Scale
GDC UI FRAMEWORK
START
Hierarchy
Create
Untitled*
Main Camera
Directional Light
UIStarter
Splash
DontDestroyOnLoad
Inspector
Console
UI
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
UI Manager (Script)
Script
UIManager
Start View
Splash (ViewConfig)
Navigation Manager
NavigationManager (NavigationManag
Interaction Blocker
InteractionBlockerOverlayCanvas (Inte
Add Component
Assets ▸ SYBO ▸ Components ▸ UI ▸ Integration ▸ Resou
LineBreaking ...
LineBreaking ...
Shaders
Sprite Assets
Style Sheets
TMP Settings
UI
Asset Labels

FRAMEWORK CONCEPT RECAP

Slice

Slice Panel 1

Slice Panel 2

Animation Targets

View 1

Slice 1

Slice 2

Slice 3

Navigation transition

View 2

Slice 4

Slice 3

NAVIGATION TRANSITIONS

- Can be built by designer
- Once built, can be easily reused across the views and projects
- Easy to start prototyping inside the project, saving time

TECHNICAL UNIFICATION

- Interaction design principles and basic UI architecture become unified, that helps to maintain holistic design.
- Less time consuming and requires less resources to develop and maintain

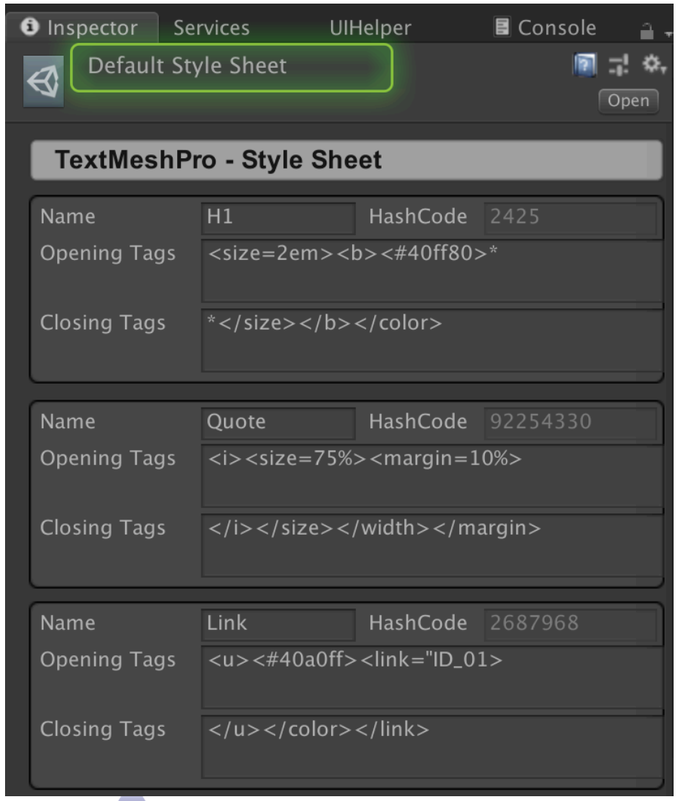
UI INNOVATIONS

- Once a good solution is found and implemented in the UI framework, all the projects get the benefit of using it
- In-house development allows us to adapt it fast for all the technical needs

COMPLIMENTED WITH OTHER TOOLS

TYPOGRAPHY

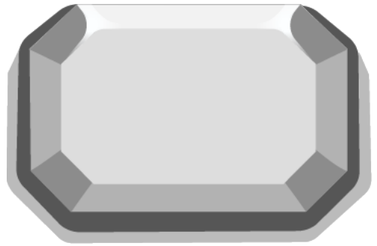
TextMeshPro
Style Sheets



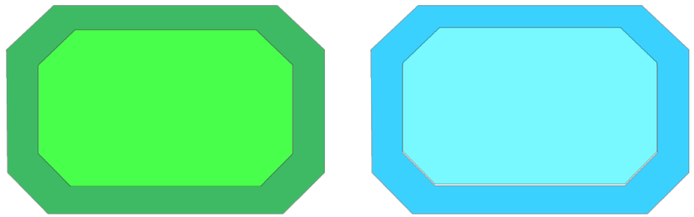
COLORS

Unity
Color libraries

Desaturated assets



Colored in Unity



OTHER TOOLS

Vector assets

UI system in Sketch

Animated prototypes

SYBO UI FRAMEWORK

- The framework allows us to have a living design-system of components and interaction patterns **inside the project** that almost anyone can use.
- The process of redesign - easy and fast
- First of all is a **time-saving measure**, that allows to focus on research, design and testing, rather than implementation

THANK YOU!

Big thanks to Johnny Mikkelsen and SYBO tools team