



# Data Binding Architectures for Rapid UI Creation in Unity

**Stephan Dilly**

Head of Frontend Engineering  
InnoGames GmbH



CULTURES  
ONLINE







# InnoGames GmbH

- Founded in 2007
- Started with browser games
- Migration to mobile
- Mobile-only strategy now





# Lost Survivor

- RPG/Survival
- 3D
- Launched in Netherlands





6

ELSA ADLER



100%



350 / 330



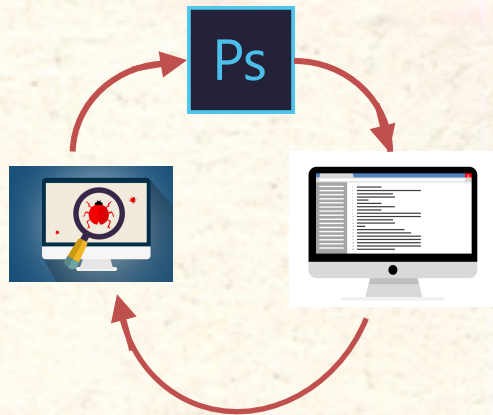






# The Old Approach

- Artists work in Photoshop
- UI Developer applies magic
- QA finds the bugs



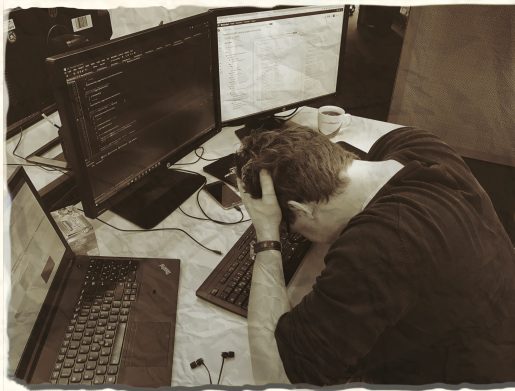




Say Hi to..



Artist Olli



Dev Lars



QA Dennis





# Flaws

- 'Not my problem' attitude
- Developer plays UI artist
- Untested spaghetti code
- Long turnaround times





# Architecture in Lost Survivor

- Artist works directly in Unity
- Dev/Art work decoupled
- Dev/Art work parallel







Everyone is happy





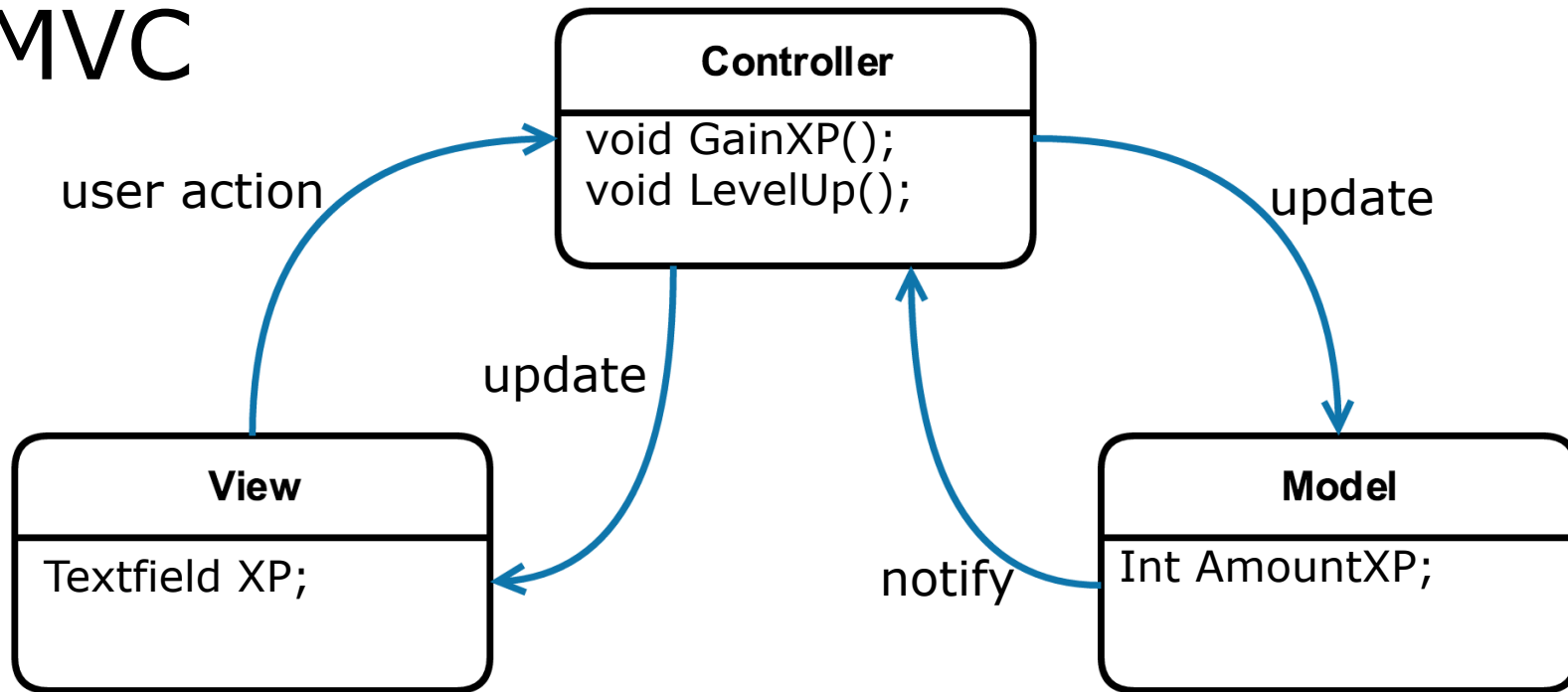
# How did we get there?





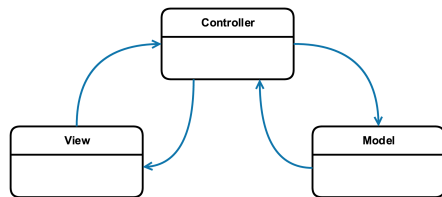


# MVC





# MVC



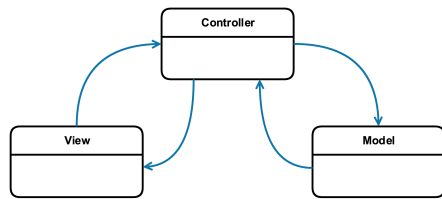
“[Objects] tend to be more reusable, and their interfaces tend to be better defined.”





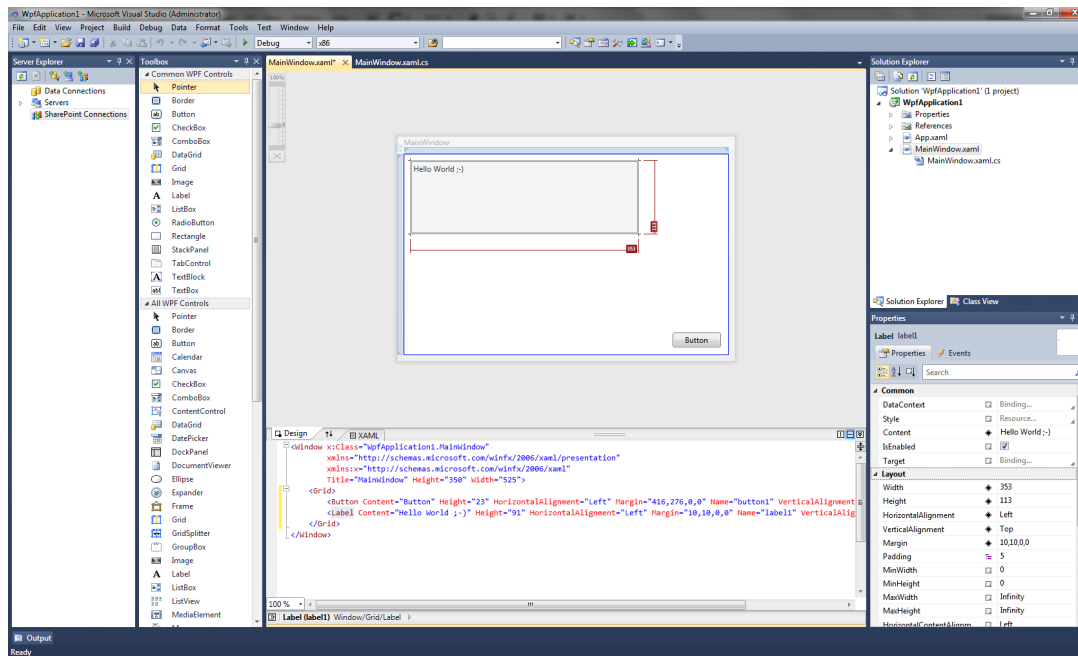
# MVC

- Native iOS SDK support
- Separation of concerns
- Visual designer in XCode



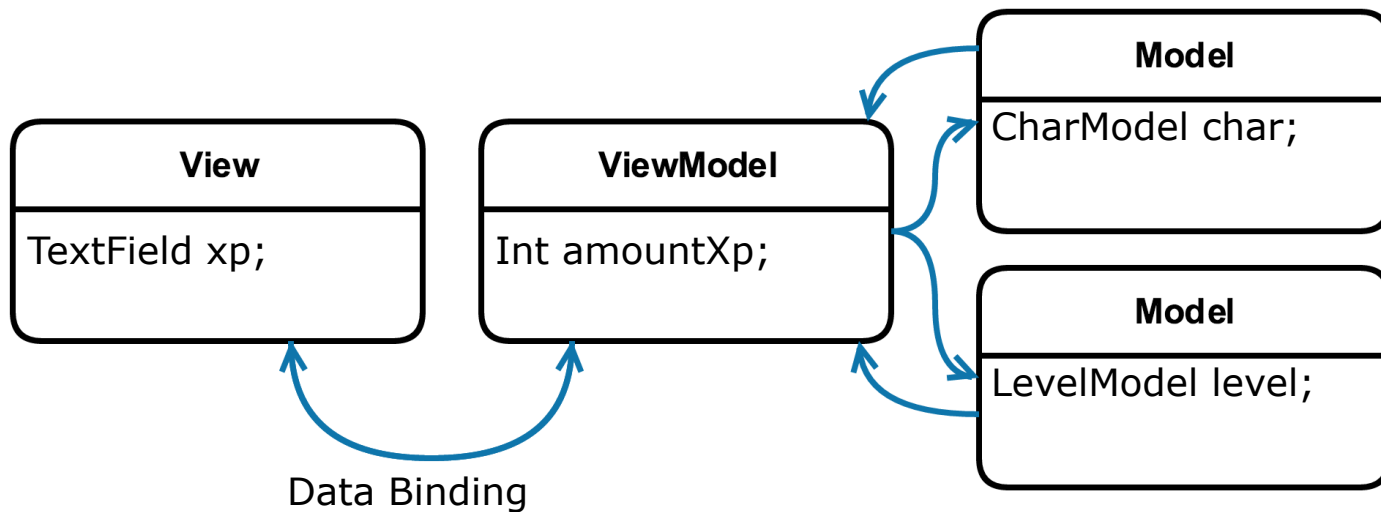


# Windows Presentation Foundation





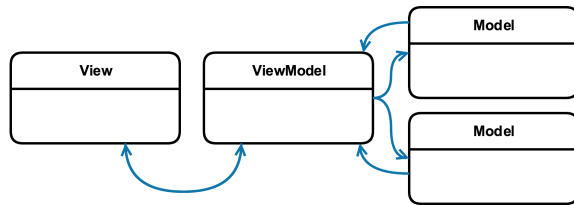
# MVVM





# MVVM

- ViewModel serves the View
- One ViewModel per View
- Based on Data Binding





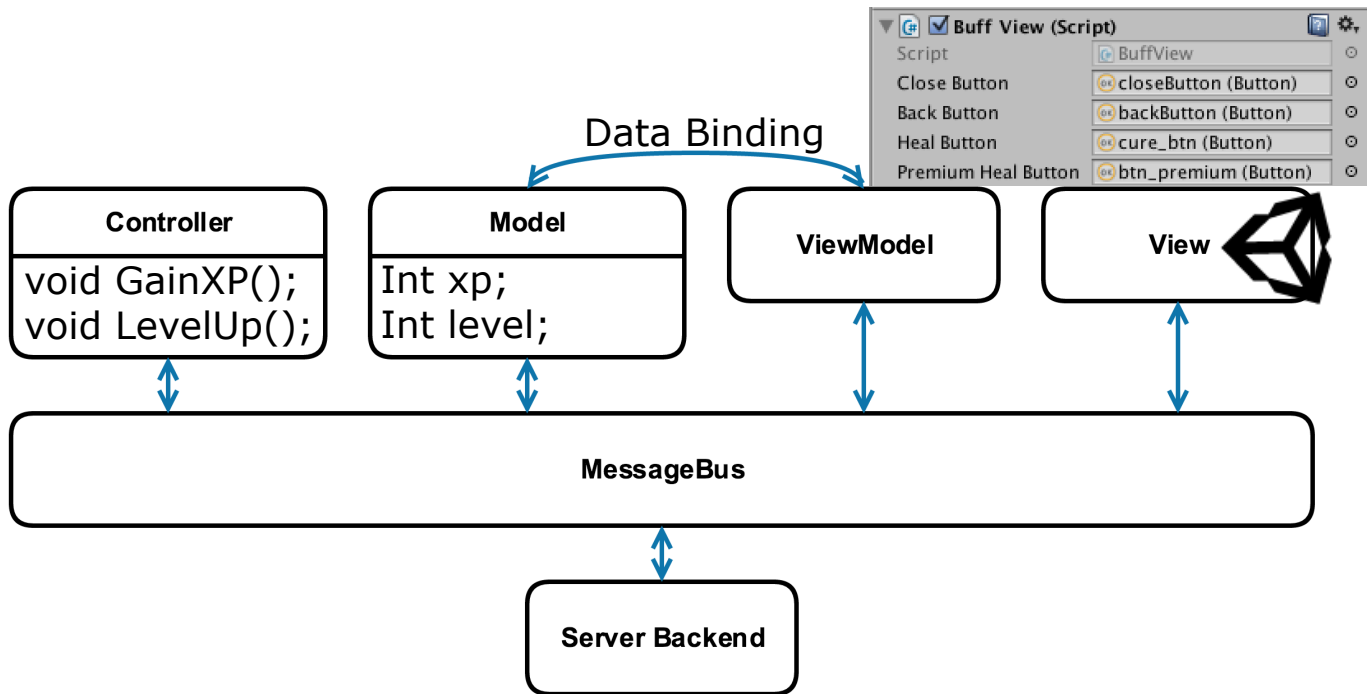


# Our Architecture

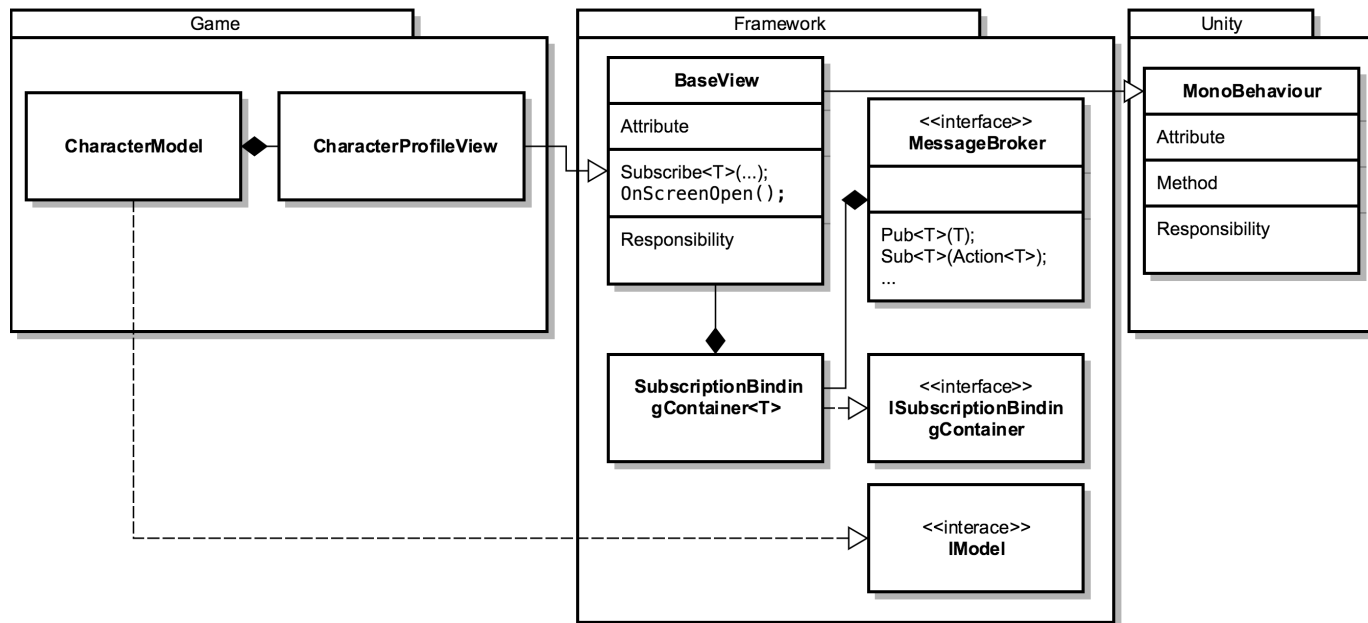




# Architecture in Lost Survivor



# Classes



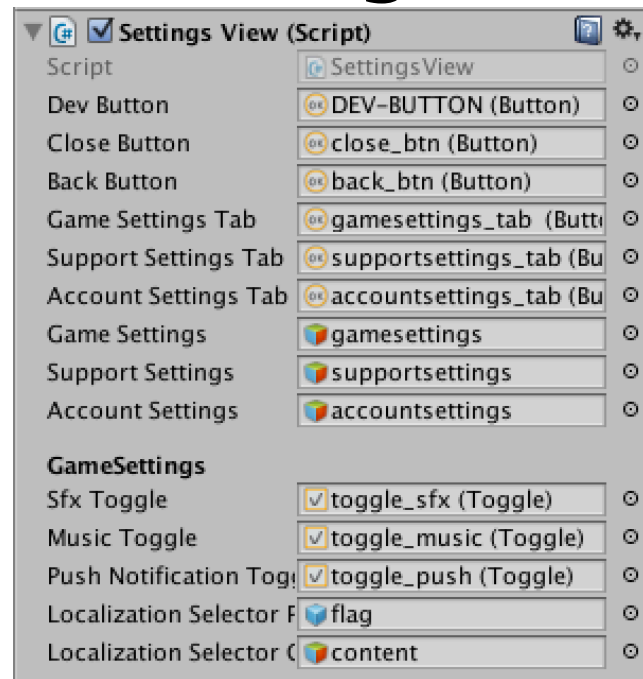
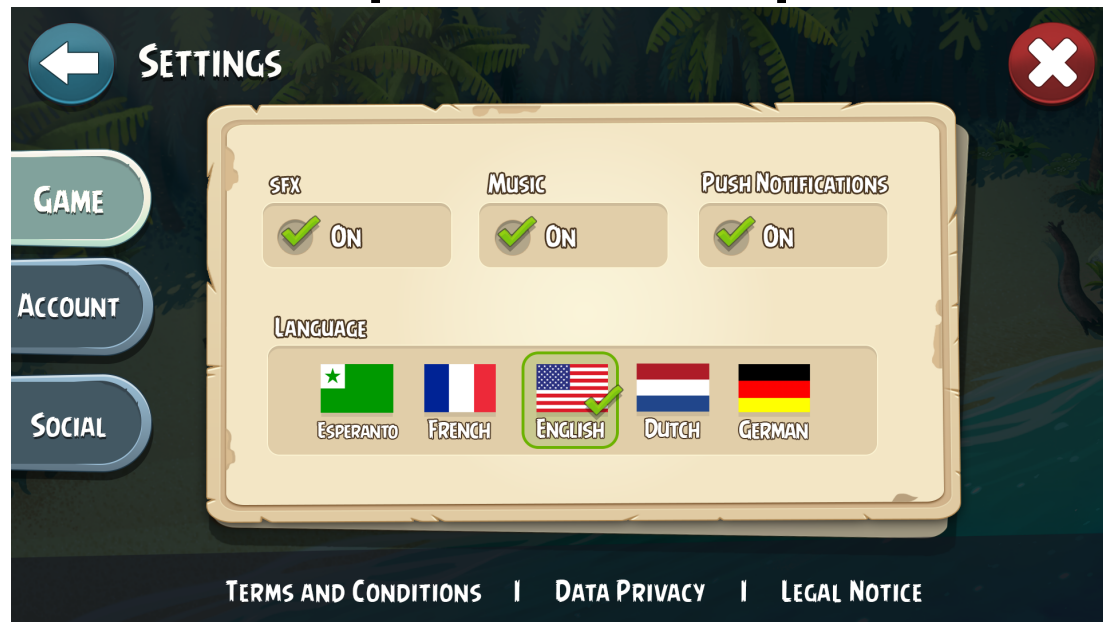


# Examples





# Example: Simple Data Binding





# Example: Simple Data Binding

```
// bind toggles to method calls
Subscribe<SettingsConfigurationModel>()
    .BindToggle(MusicToggle, _audioService.MuteMusic, true)
    .BindToggle(SfxToggle, _audioService.MuteSfx, true)
    .BindToggle(PushNotificationToggle, SetPushEnabled)
```



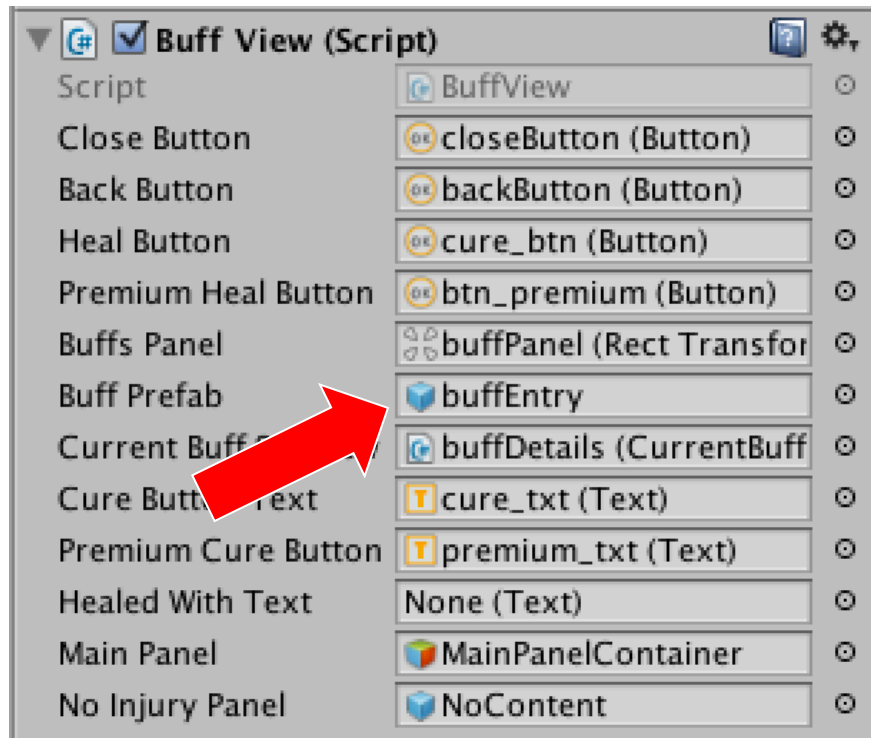




# Example: Custom Data Binding









# Example: Custom Data Binding

```
Subscribe<CharacterModel>()  
    .BindModelChangeAction(UpdateBuffObjects)  
    .BindButton(HealButton, CurrentBuffSubview, (model, script) => ...)  
    ...  
.Finish();
```

```
private void UpdateBuffObjects(CharacterModel model){  
  
    for (int i = 0, count = model.ActiveStageBuffs.Count; i < count; i++)  
    {  
        InstantiateNewBuffGameObject(CurrentBuffSubview, model.ActiveStageBuffs[i]);  
    }  
    ...  
}
```





# Performance Impact

- Based on Reflection
- Startup only
- No Garbage Collector pressure





# Testing The Design

- Artist waits for dev
- Waits for build
- Has to actually play
- Isolation impossible



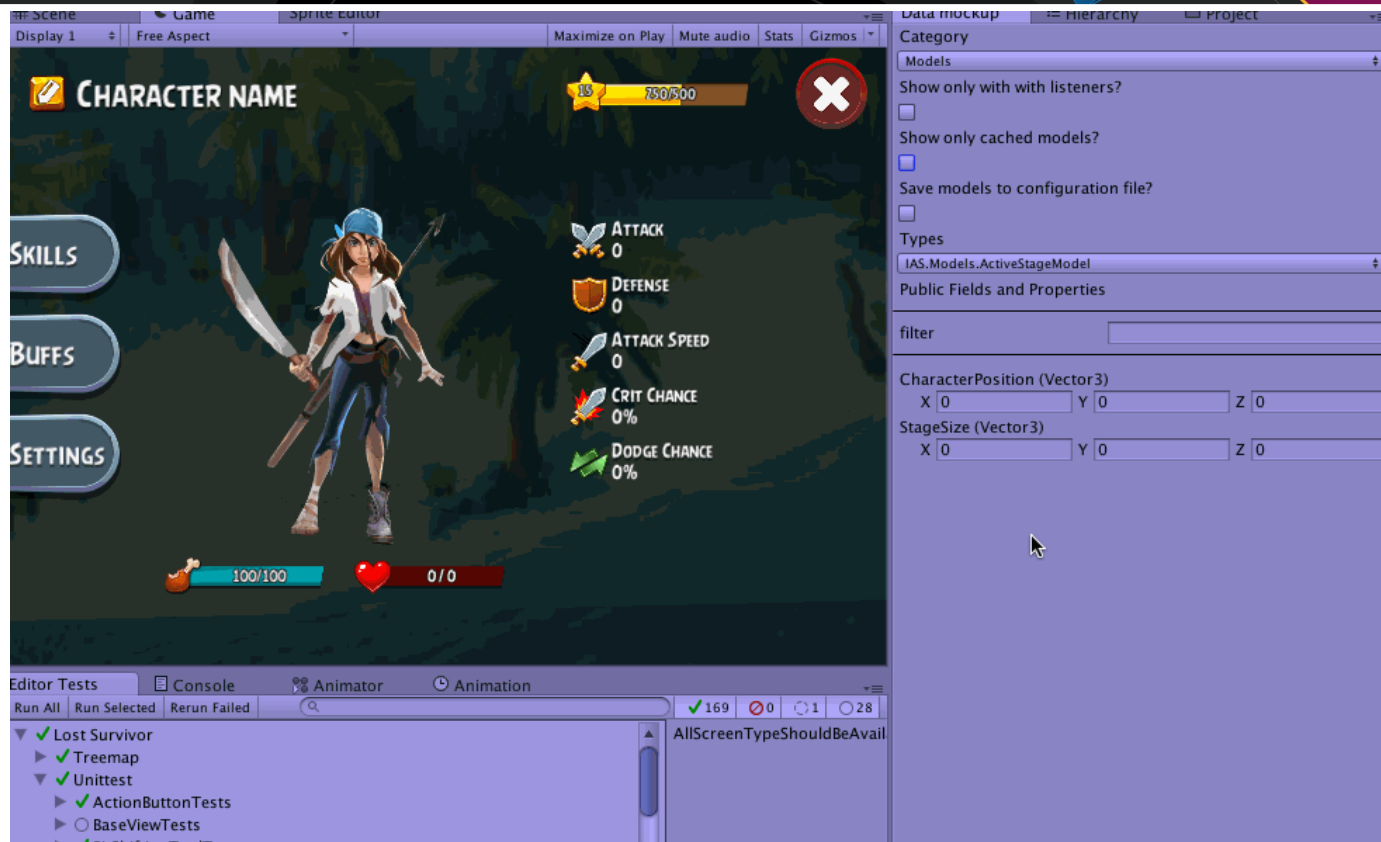




# Mocking For The Rescue

- Isolated testing
- Fast iteration
- Little dependencies
- Everything can be simulated









# Working Together in Unity







# Asset Merging

```
m_IsActive: 1
--- !u!4 401726
Transform:
  m_ObjectHideFlags: 1
  m_PrefabParentObject: {fileID: 0}
  m_PrefabInternal: {fileID: 100100000}
  m_GameObject: {fileID: 106530}
  m_LocalRotation: {x: 0, y: 0, z: 0, w: 1}
  m_LocalPosition: {x: 0, y: 0, z: 0}
  m_LocalScale: {x: 1, y: 1, z: 1}
  m_Children: []
  m_Father: {fileID: 0}
  m_RootOrder: 2
--- !u!4 419642
Transform:
  m_ObjectHideFlags: 0
  m_PrefabParentObject: {fileID: 0}
  m_PrefabInternal: {fileID: 100100000}
  m_GameObject: {fileID: 106530}
  m_LocalRotation: {x: 0, y: 0, z: .707106829, w: .707106709}
  m_LocalPosition: {x: 14.5777721, y: 8.67928696, z: -10}
  m_LocalScale: {x: .100000001, y: .100000001, z: .100000001}
  m_Children: []
  m_Father: {fileID: 0}
  m_RootOrder: 0
--- !u!198 419807452
ParticleSystem:
  m_ObjectHideFlags: 1
  m_PrefabParentObject: {fileID: 0}
  m_PrefabInternal: {fileID: 100100000}
  m_GameObject: {fileID: 106530}
  lengthInSec: .100000001
  startDelay: 0
```

```
m_Name: explode
m_TagString: Untagged
m_Icon: {fileID: 0}
m_NavMeshLayer: 0
m_StaticEditorFlags: 0
m_IsActive: 1
--- !u!4 419642
Transform:
  m_ObjectHideFlags: 1
  m_PrefabParentObject: {fileID: 0}
  m_PrefabInternal: {fileID: 100100000}
  m_GameObject: {fileID: 106530}
  m_LocalRotation: {x: 0, y: 0, z: .707106829, w: .707106709}
  m_LocalPosition: {x: 14.5777721, y: 8.67928696, z: -10}
  m_LocalScale: {x: .100000001, y: .100000001, z: .100000001}
  m_Children: []
  m_Father: {fileID: 0}
  m_RootOrder: 0
--- !u!198 419807452
ParticleSystem:
  m_ObjectHideFlags: 1
  m_PrefabParentObject: {fileID: 0}
  m_PrefabInternal: {fileID: 100100000}
  m_GameObject: {fileID: 106530}
  lengthInSec: .100000001
  startDelay: 0
  speed: 1
  randomSeed: 0
  looping: 0
  prewarm: 0
  playOnAwake: 0
  moveWithTransform: 0
  InitialModule:
    enabled: 1
    startLifetime:
      scalar: 1
```





# Solve Asset Merging

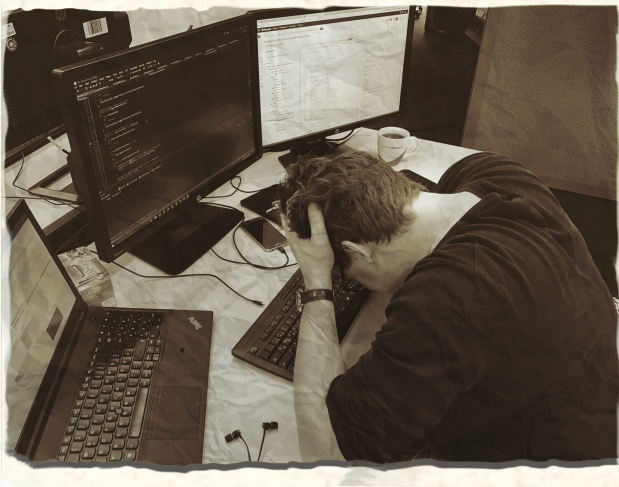
- Not an issue in this architecture
- Artist owns the UI scene
- Dev owns code
- No conflict





# Unit testing

- Untestable code
- No unit tests
- Long turnaround times



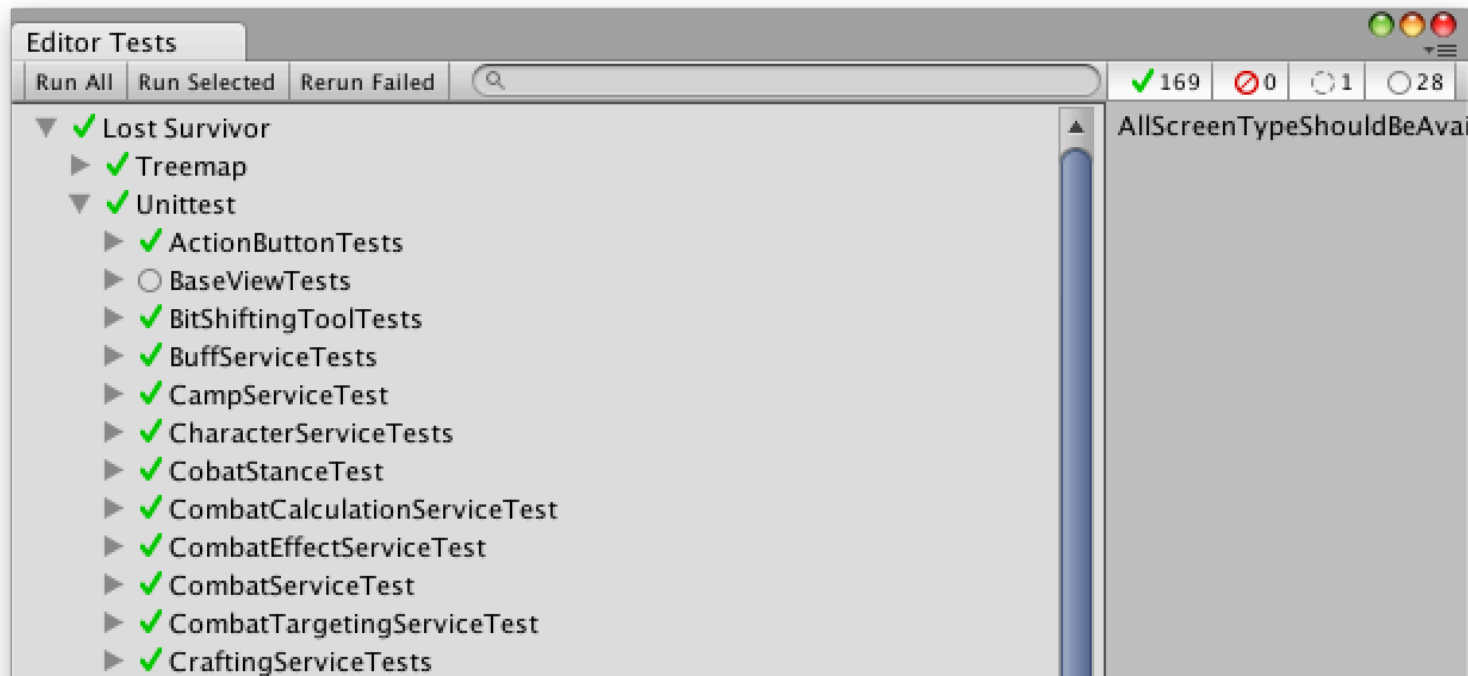


# Unit Testing Now





# Unit Testing Now







# Unit Testing Now







# Manual Testing

- Outsourcing costs
- Waterfall model
- Device fragmentation







## PRODUCTION



**Release: 1.94.1**

10 Feb 19:52: Full roll-c



Supported devices: **13039**







# Manual Testing





# Automation Tests



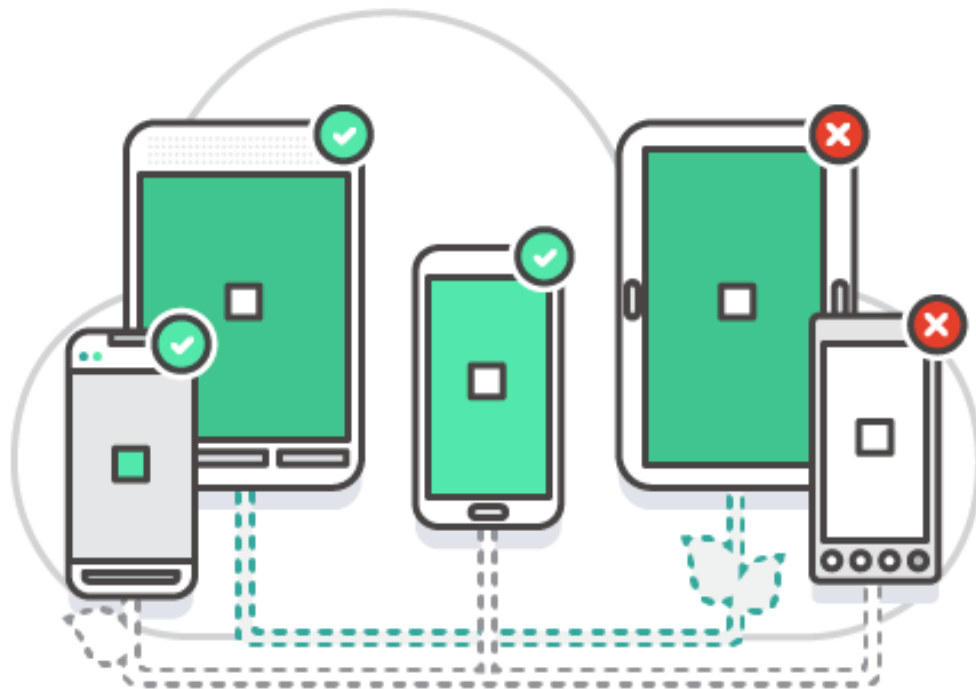






©Facebook





© AWS DeviceFarm







# Automation Tests





# Validation

- What if linking breaks?





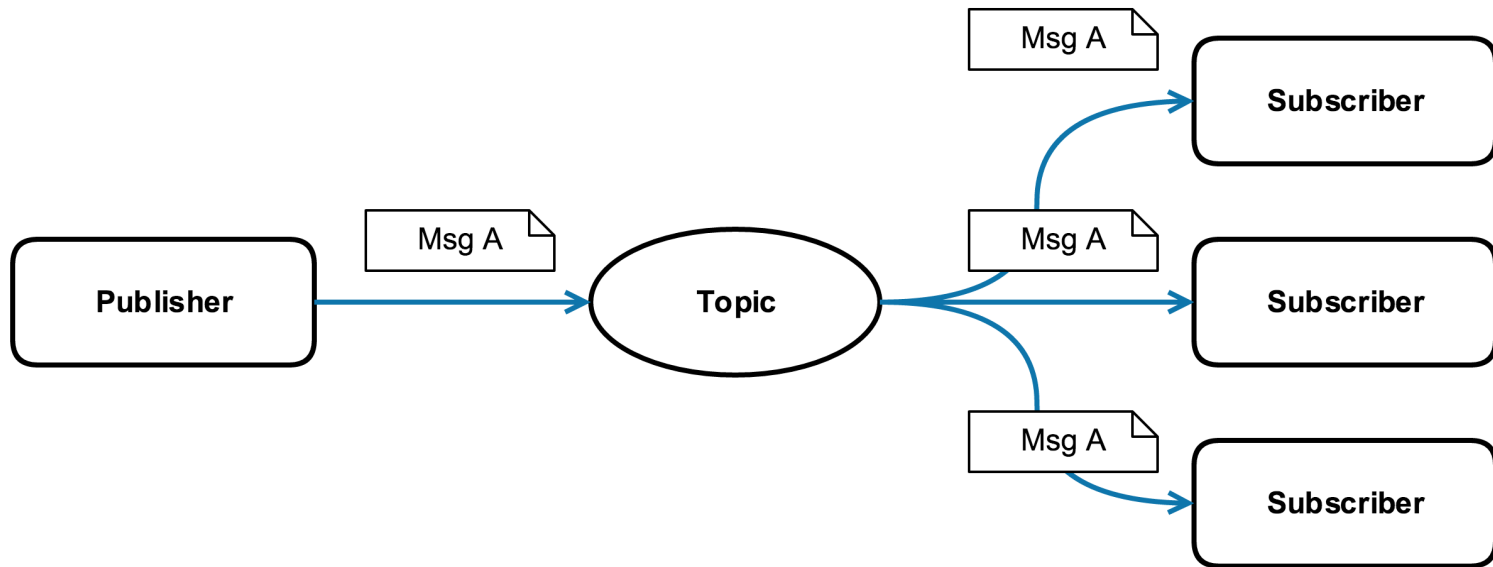
# Validation

```
void OnValidate(){  
  
    // perform the binding  
    OnBind();  
  
    // check if all fields expected from  
    // the binding are present in the model  
    CheckViewToModelRelation();  
  
    // make sure all expected view elements are setup and wired  
    CheckViewsetup();  
  
    UnBind();  
}
```





# MessageBus





# MessageBus - Code

```
// You subscribe to a type not a name  
_messageBroker.Sub<CombatModel>(UpdateState);
```

```
CombatModel combatModel;  
combatModel.Pending.Add(...);  
_messageBroker.Pub(_combatModel);
```





# MessageBus - Performance

- Based on reflection
- Cached reflection objects
- No measurable impact





# Alternatives





cgarciae / karma

Unwatch ▾

9

★ Unstar

19

🍴 Fork

5

&lt;&gt; Code

🔔 Issues 3

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

📶 Pulse

📊 Graphs

An MVC framework for Unity3D

📌 25 commits

🌿 1 branch

📦 0 releases

👤 3 contributors

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



cgarciae committed on GitHub Merge pull request #5 from SacredSkull/master ...

Latest commit d391746 21 days ago

📁 Assets

Updated version stamp to 4.7

21 days ago

📁 ProjectSettings

Updated to Zenject 4.7

22 days ago

📄 .gitattributes

👤 Added .gitattributes &amp; .gitignore files

11 months ago

📄 .gitignore

Init

11 months ago

📄 README.md

Update README.md

9 months ago

📖 README.md

## What is Karma?

MVC Framework for Unity3D







# Alternative: Karma

- Inherit from MVCPresenter
- Annotate with [Element(path)]



Too much ceremony



# Data Bind for Unity

Scripting/GUI

Slash Games

★★★★ (125)

\$25

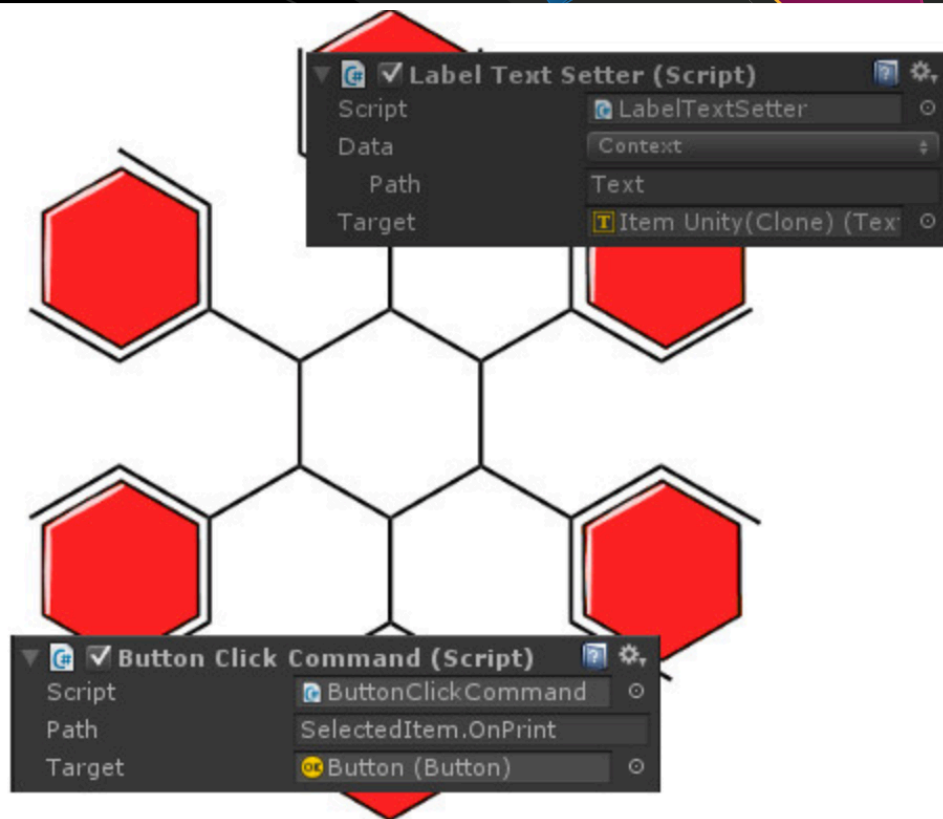
Add to Cart



Requires Unity 5.3.0 or higher.

It's always a good idea to separate your game logic and its presentation. So why not use a well-tested and clean approach to make the connection of the two for you?

**Data Bind for Unity** offers you a well thought through architecture which keeps your UI clean from your logic and the other way around. This goes so far that you won't need





# Alternative: DataBind

- No DI
- Data Binding in the editor



Too Editor Oriented





# Conclusion

- Dev/Art can focus
- Mocking
- Working together in Unity
- Unit testing
- UI Testing





# Future Plans





# Mocking Improvements

- Support interfaces/abstract classes
- Scan the assembly
- Provide DropDown list





# Code Generation

- Reduce manual maintenance
- Artist can auto generate class
- Use partial C# classes







Make it OpenSource





Start Now! 





# Thank you. Questions?

Contact me on Twitter @extrawurst

