

A blue bus is parked on a city street, heavily decorated with graffiti and posters. The bus has a yellow star on the front left and a sign that reads "OPEN PODIUM VOOR TALENT" on the roof. Posters on the side include "NEW SKOOL Moving Arts Project" and "CHILMOBIEL". A sign on the ground in front of the bus says "KOM OF TREDE". The background shows a building with many windows.

Presented by: Edward (E.T.) Garcia

Event Bus (in Unity)

Course Information



- Course Name: Game Programming
- Course Number: CMPS-6180-01
- CRN: 20733
- Instructors: Shihran Joudeh, Ali Raza
- Credit Hours: 3
- Type: Online Course
- Schedule: Mondays
- Time: 06:00PM - 07:30PM
- Dates: 05/20/2024 - 08/09/2024
- Max Seats: 25
- Available Seats: 17
- Campus: Distance Learning
- Part of Term: Special Offering
- Eligibility: Open to Graduate Students
- Wait List: N/A
- University: Tulane University

**The slides and code were prepared for Tulane University's Game Programming course.*

Introduction



- Background: 3 years at Pixeldash Studios (Senior Developer)
 - Education: 6 years of Computer Science Graduate work at UNO
 - Indie game development since 2014
-
- Unity Events
 - Event Bus (one of many implementations)
 - Switch back and forth between slide and demo
 - This information is based on my experience, your mileage may vary!



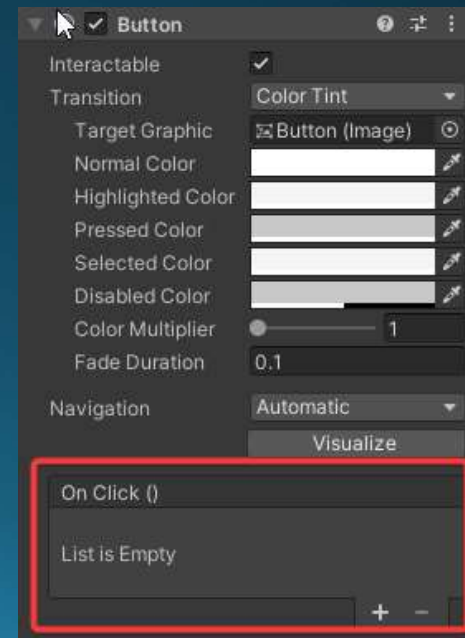
Platformer Setup!

Demo o

Unity Events

AddListener vs AddPersistentListener

- Overall, Persistent event listeners in Unity are event listeners that remain attached to events.
- Provides a visual representation of event connections, making it easier to manage and debug complex interactions.



Collecting coins!

Demo 1

Problem with Demo 1 Implementation

References!!!

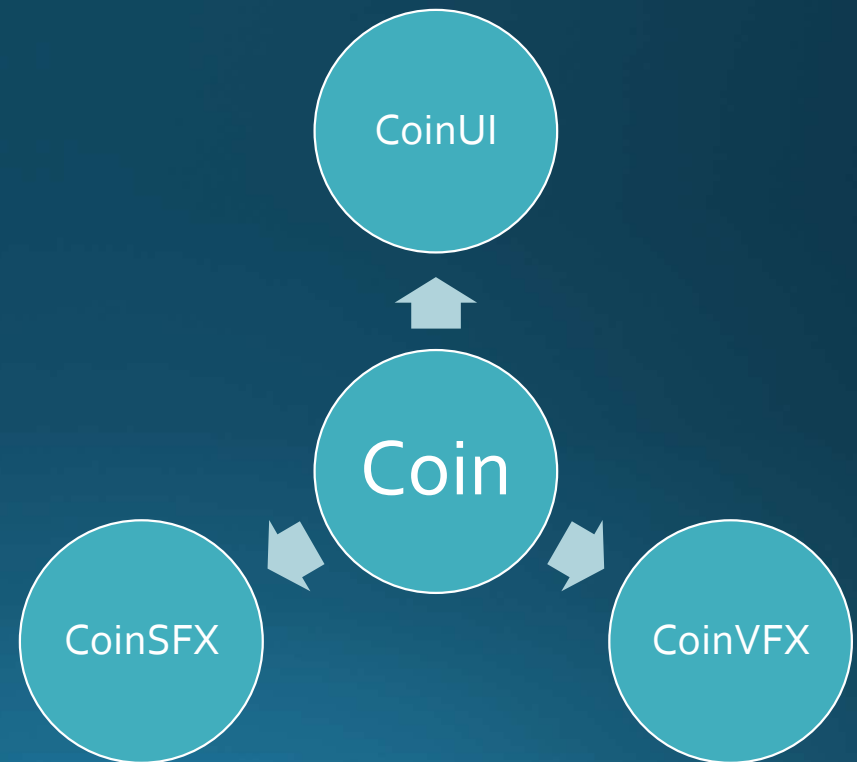
- CoinA → CoinUIA
- CoinB → CoinUIB

VFX

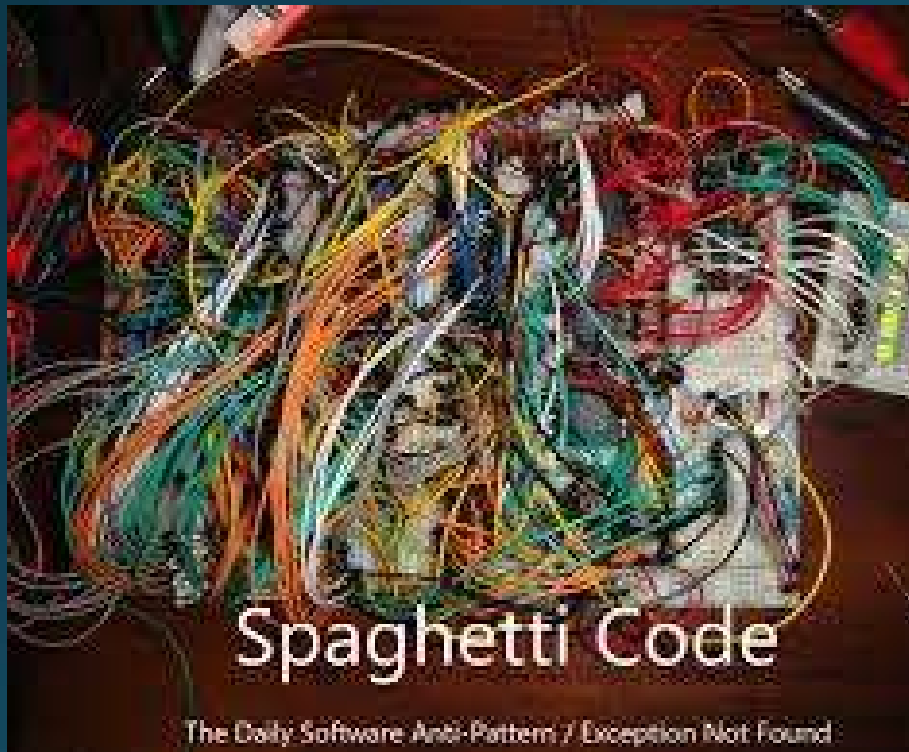
- Coin → CoinVFX...

Audio

- Coin → CoinSFX



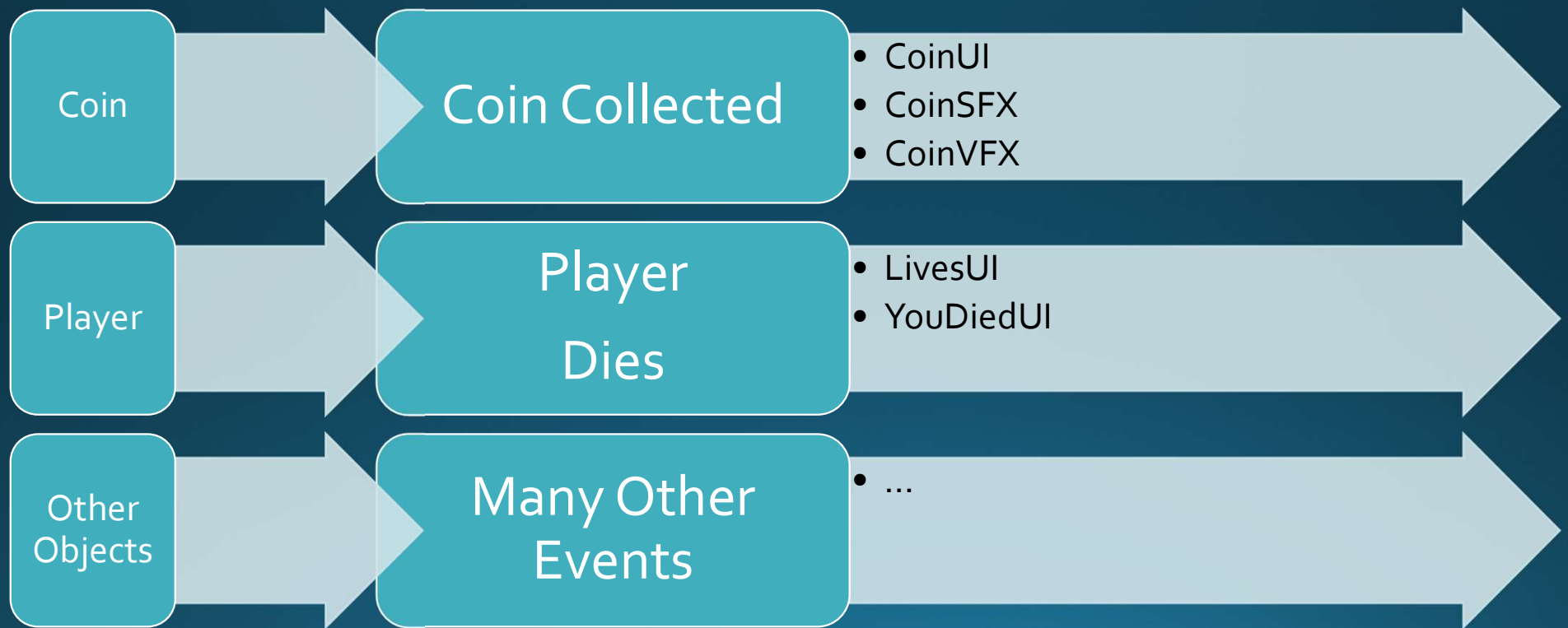
Spaghetti Code



Remove Object from Spaghetti Code...



Global Event List



Global Event List

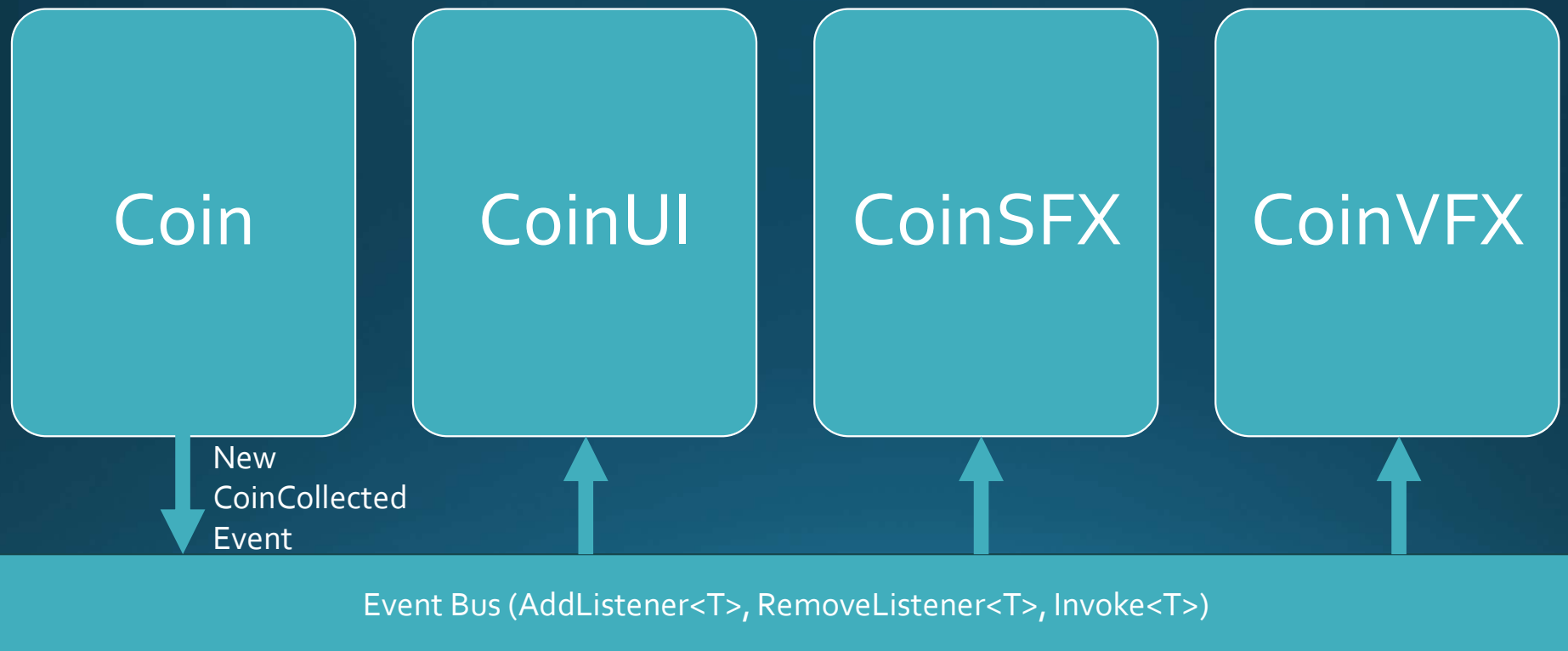
Demo 2

Issue with Demo 2

Maintenance (“Changeability”)

- In Order to Modify the Parameters of an event, it requires changes in 3 places.
 - The Global Event List
 - The Listener
 - The Invoker
- Also, GlobalEventList will be modified often [potential conflicts on teams]

Event Bus (some rando's implementation)_{me}



An Event

- DATA ONLY! Has two responsibilities
 - Identifies what Type of Event has been invoked
 - Contains all the parameters you need to handle the event

```
1  using UnityEngine;
2
3  namespace _3_EventBus
4  {
5      public class CoinCollectedEvent 8: 10 usages ETdoFresh
6      {
7          public Vector3 Position { get; set; } 8: 2 usages
8      }
9  }
```

Event Bus

- A (likely) Global Static or Singleton Class
- After implementation, in any class, anywhere we should be able to do the follow:
 - `EventBus.AddListener<CoinCollected>(OnCoinCollected);`
 - `EventBus.RemoveListener<CoinCollected>(OnCoinCollected);`
 - `EventBus.Invoke(new CoinCollected { Position = transform.position });`

Event Bus!!!

Demo 3

Event Strategies

Direct Events

- Direct connections between event sources and listeners, where events on source invoke specific methods on the listeners.

Global Event List

- A centralized list of events that listeners can subscribe to and sources can publish to, allowing for decoupling but requiring more management.

Event Bus

- A more advanced system that decouples sources and listeners completely, centralizing the event management and improving scalability and maintainability.

Problems you may run into overusing Events

Multiple Sources of Truth

- Many objects keeping state

Orchestration Issues

- Race conditions, everyone listening OnEnable or Awake, no easily readable way to determine order

Missed Events

- Map Power Up? Speed boost? Late player.

Event chains

- Collect Coin → Update Inventory → Update UI → Screen Effect

A Possible Solution

- Remove and replace listeners with a Manager as a Listener!

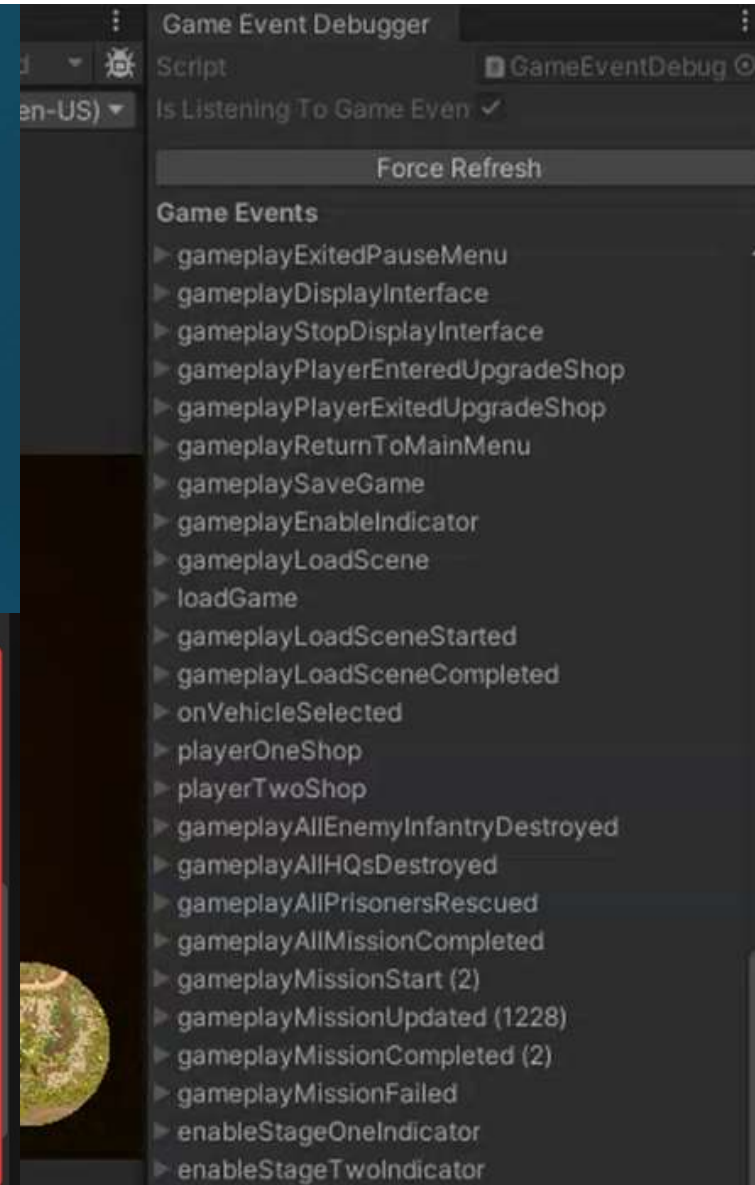
BindableData (+ maybe a DataBus/Database)

- `BindableData<T>` // `Observable<T>`, `TrackChanges<T>`, etc
- `<T>` is value
- Launches events when changed
- Typically implemented where added method invoked immediately on `AddListener`.
- Battles against missed events

Event Debugger using Reflection

- If you are on a project for longer term, making debug tools is quite helpful.
- Shown here is stack trace of event invocations!

```
gameplayAllMissionCompleted
▼ gameplayMissionStart (2)
  ▼ [0.514424] gameplayMissionStart(params: 1)
    UnityEngine.StackTraceUtility:ExtractStackTrace ()
    (wrapper dynamic-method) object:lambda_method (System.Runtime.CompilerServices
    UnityEngine.Events.InvokableCall`1<MissionInstance>:Invoke (MissionInstance)
    UnityEngine.Events.UnityEvent`1<MissionInstance>:Invoke (MissionInstance)
    MissionInstance:OnMissionActivate () (at Assets/Game/MissionSystem/Missions/Missi
    GoToLocationMissionInstance:OnMissionActivate () (at Assets/Game/MissionSystem/M
    MissionInstance/<WaitForMissionActivation>d_45:MoveNext () (at Assets/Game/Miss
    UnityEngine.SetupCoroutine:InvokeMoveNext (System.Collections.IEnumerator,intptr)
  ▼ [0.5414667] gameplayMissionStart(params: 1)
    UnityEngine.StackTraceUtility:ExtractStackTrace ()
    (wrapper dynamic-method) object:lambda_method (System.Runtime.CompilerServices
    UnityEngine.Events.InvokableCall`1<MissionInstance>:Invoke (MissionInstance)
    UnityEngine.Events.UnityEvent`1<MissionInstance>:Invoke (MissionInstance)
    Game.MissionSystem.MissionUI/<HandlePopupQueue>d_19:MoveNext () (at Assets/G
    UnityEngine.SetupCoroutine:InvokeMoveNext (System.Collections.IEnumerator,intptr)
gameplayMissionUpdated (761)
```



The floor is open!

Q&A Session