

# UI Engineering patterns from 'Marvel's Midnight Suns'

Joel Gonzales  
UI Engineering Lead  
Turtle Rock Studios

# Outline

- Introduction
- The Problem
- UICS - User Interface Component System
- Building Around Components
- Best Practices
- Summary

# About Me



# About Me



# About Me



# Marvel Midnight Suns

- Launched December 2022
- 5 year Development Cycle
- Starting from a Clean Slate





CTU: 8  
Total: 9

GARAGE **STORE** DEPOT MISSIONS **CAMPAIGN** SERVICE RECORD TECH TREE BARRACKS STRONGHOLD

### ITEMS FOR BONDS

Equipment

Directives



Venting System  
5,000



Innovative Loading System  
5,000



Stabilizing Equipment System  
5,000



Wear-Resistant Gun Laying Drive  
5,000



Experimental Optics  
4,000



Extended Spare Parts Kit  
3,000



W4

General



CTU: 8  
Total: 9

GARAGE **STORE** DEPOT MISSIONS CAMPAIGN SERVICE RECORD TECH TREE BARRACKS STRONGHOLD

### ITEMS FOR BONDS

Equipment

Directives



Venting System  
5,000



Innovative Loading System  
5,000



Wear-Resistant Gun Laying Drive  
5,000



Experimental Optics  
4,000



GAMES

SHOP



ALL GAMES

FAVORITES



Shop Overwatch

Overwatch 2 Website

Patch Notes

**OVERWATCH 2**

**LIVE NOW**

twitch

Watch Overwatch 2 on Twitch to earn the Razor Sharp Kiriko spray and Donut weapon charm

September 30th, 2022

Initializing systems! Updating Competitive play for Overwatch 2

September 28th, 2022

GAME VERSION

Overwatch 2

Playing Now

Region: Americas Version: 2.1.1.0.10617



Online



★ Favorites - 2/5

Friends - 18/75



Chats and Groups

CTF: 8  
Total: 9

GARAGE **STORE** DEPOT MISSIONS CAMPAIGN SERVICE RECORD TECH TREE BARRACKS STRONGHOLD

### ITEMS FOR BONDS

Equipment

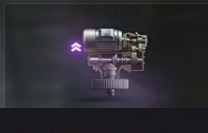
Directives



Venting System  
5,000 sp



Innovative Loading System  
5,000 sp



GAMES

SHOP

...

**FRIENDS** GROUPS RECENT PLAYERS

# SOCIAL FRIENDS

0 PLAYING OVERWATCH / 20 FRIENDS ONLINE

ADD FRIEND

ONLINE




	OFFLINE FOR 1 YEAR	
	OFFLINE FOR 11 YEARS	
	OFFLINE FOR 5 MONTHS	
	OFFLINE FOR 4 MONTHS	
	OFFLINE FOR 3 MONTHS	

	OFFLINE	
	OFFLINE FOR 7 MONTHS	
	OFFLINE	
	OFFLINE FOR 5 MONTHS	

< 2/6 >

CHAT

ESC BACK



Online



★ Favorites - 2/5

Friends - 18/75



Chats and Groups



CTU 8  
Total: 9

GARAGE STORE DEPOT MISSIONS CAMPAIGN SERVICE RECORD

TECH TREE BARRACKS STRONGHOLD

CHAMPIONS

SKINS

EMOTES

RUNES

SPELLS

ITEMS

ICONS

WARDS

CHROMAS

## ITEMS FOR BONDS

Equipment

Directives



Venting System  
5,000



Innovative Loading System  
5,000

531

TOTAL SKINS  
OWNED

3 1 23 38

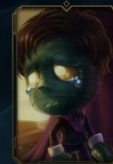
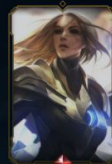
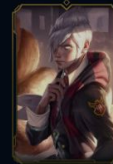
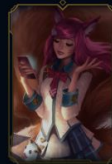
197 6

Search

My Collection

Alphabetical

A



FRIENDS GROUPS RECENT PLAYERS

# SOCIAL FRIENDS

0 PLAYING OVERWATCH / 20 FRIENDS ONLINE

	IN APP

	OFFLINE FOR 1 YEAR
	OFFLINE FOR 11 YEARS
	OFFLINE FOR 5 MONTHS
	OFFLINE FOR 4 MONTHS
	OFFLINE FOR 3 MONTHS

	OFFLINE
	OFFLINE FOR 7 MONTHS
	OFFLINE
	OFFLINE FOR 5 MONTHS

< 2/6 >

CHAT

ESC BACK



Online



★ Favorites - 2/5

Friends - 18/75



Chats and Groups



CTU 8  
Total: 9

GARAGE STORE DEPOT MISSIONS CAMPAIGN SERVICE RECORD

TECH TREE BARRACKS STRONGHOLD

CHAMPIONS

SKINS

EMOTES

RUNES

SPELLS

ITEMS

ICONS

WARDS

CHROMAS

## ITEMS FOR BONDS

Equipment

Directives



Venting System  
5,000



Innovative Loading System  
5,000

531

TOTAL SKINS  
OWNED

3 1 23 38

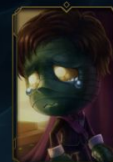
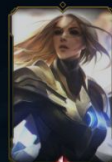
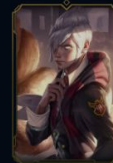
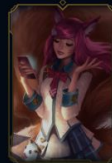
197 6

Search

My Collection

Alphabetical

A



FRIENDS GROUPS RECENT PLAYERS

# SOCIAL FRIENDS

0 PLAYING OVERWATCH / 20 FRIENDS ONLINE

	IN APP

	OFFLINE FOR 1 YEAR
	OFFLINE FOR 11 YEARS
	OFFLINE FOR 5 MONTHS
	OFFLINE FOR 4 MONTHS
	OFFLINE FOR 3 MONTHS

	OFFLINE
	OFFLINE FOR 7 MONTHS
	OFFLINE
	OFFLINE FOR 5 MONTHS

< 2/6 >

CHAT

ESC BACK



Online



★ Favorites - 2/5

Friends - 18/75



Chats and Groups

# What is a Frontend?

# What is a Frontend?

- User Interface that exists outside of the core game loop.

# What is a Frontend?



# What is a Frontend?

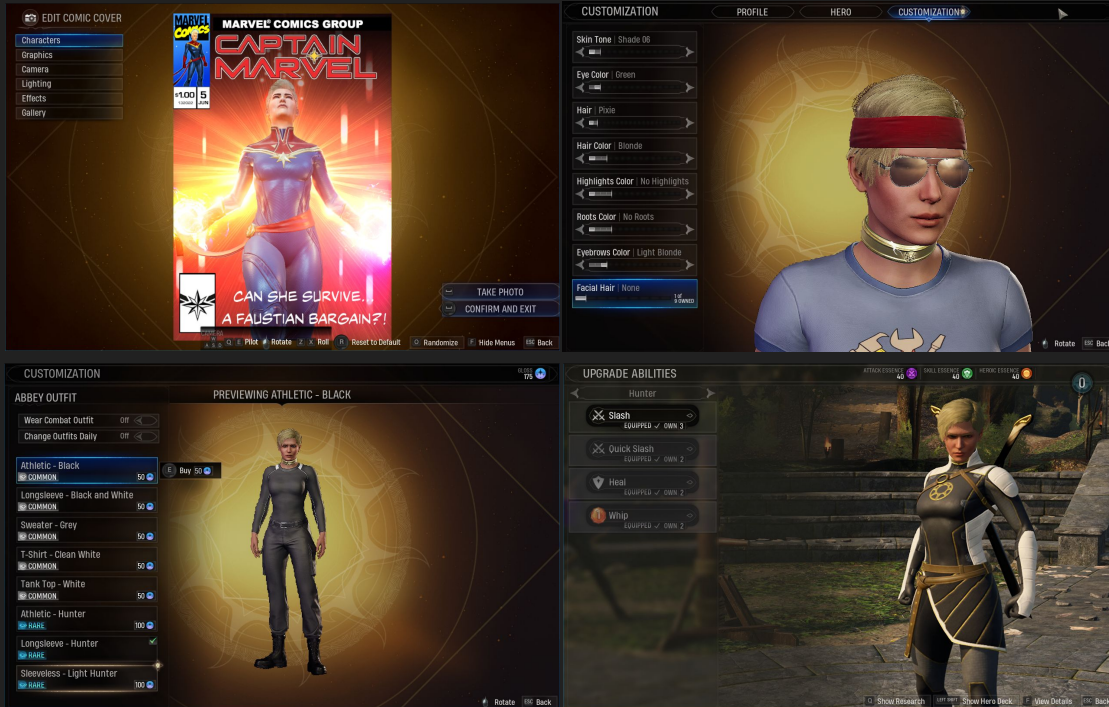


Core Game Loop



# What is a Frontend?

## Core Game Loop

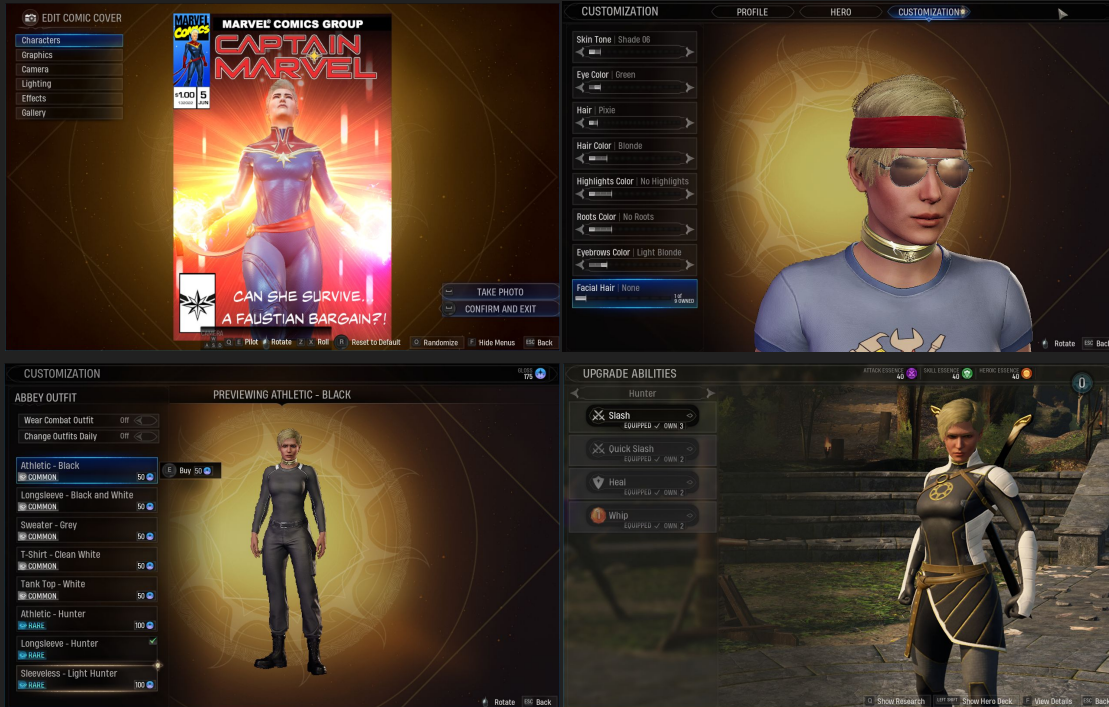
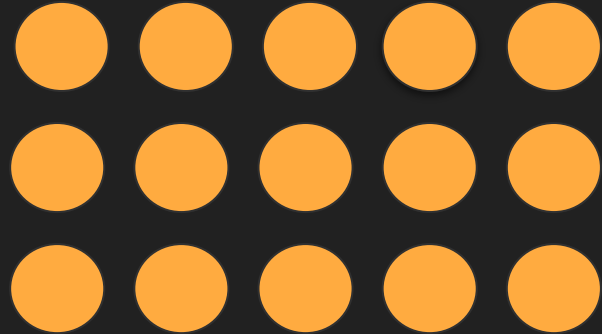


# What is a FrontEnd?

Core Game Loop



Front-End



# Terminology

# Terminology

- Control - Visual that the user can interact with

# Terminology

- Control - Visual that the user can interact with
- Screen - Base of our UI system



# About the Internal Midnight Suns System

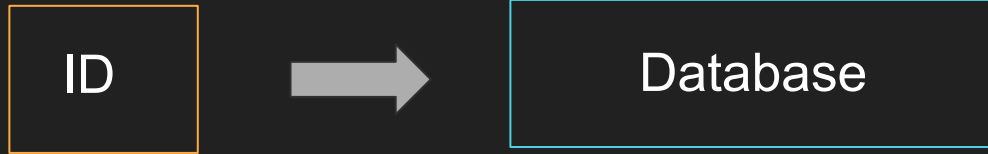
How do we get data?



ID

# About the Internal Midnight Suns System

How do we get data?



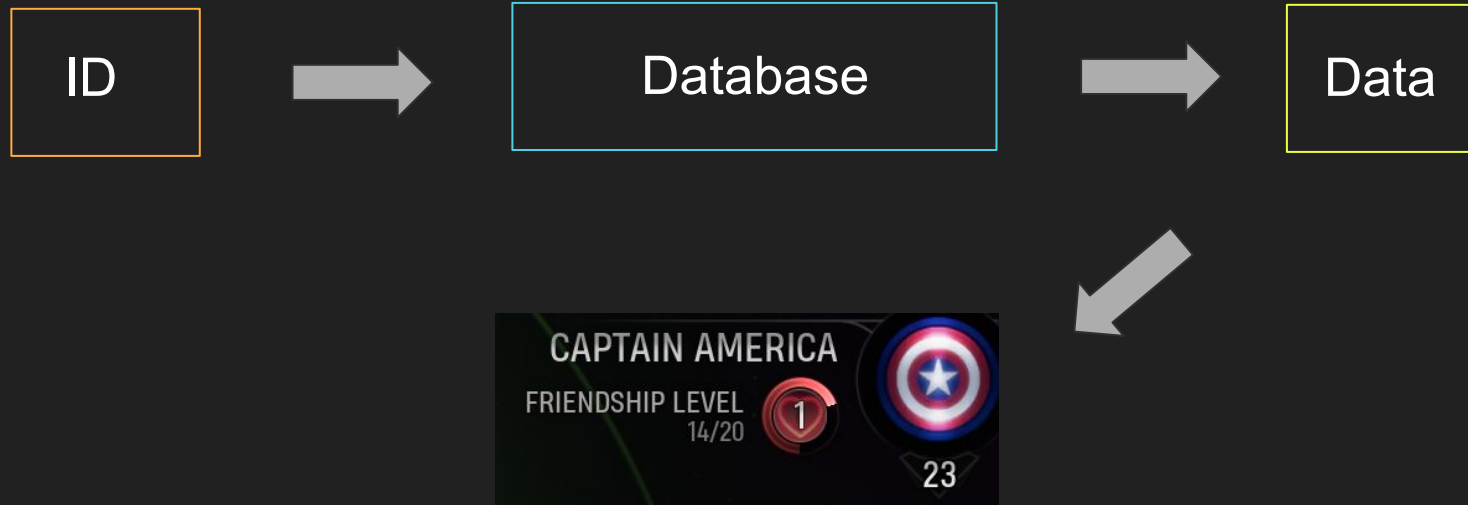
# About the Internal Midnight Suns System

How do we get data?



# About the Internal Midnight Suns System

We have a Global Database class that uses Identifiers to get specific data



- Introduction
- **The Problem**
- UICS - User Interface Component System
- Building Around Components
- Best Practices
- Summary

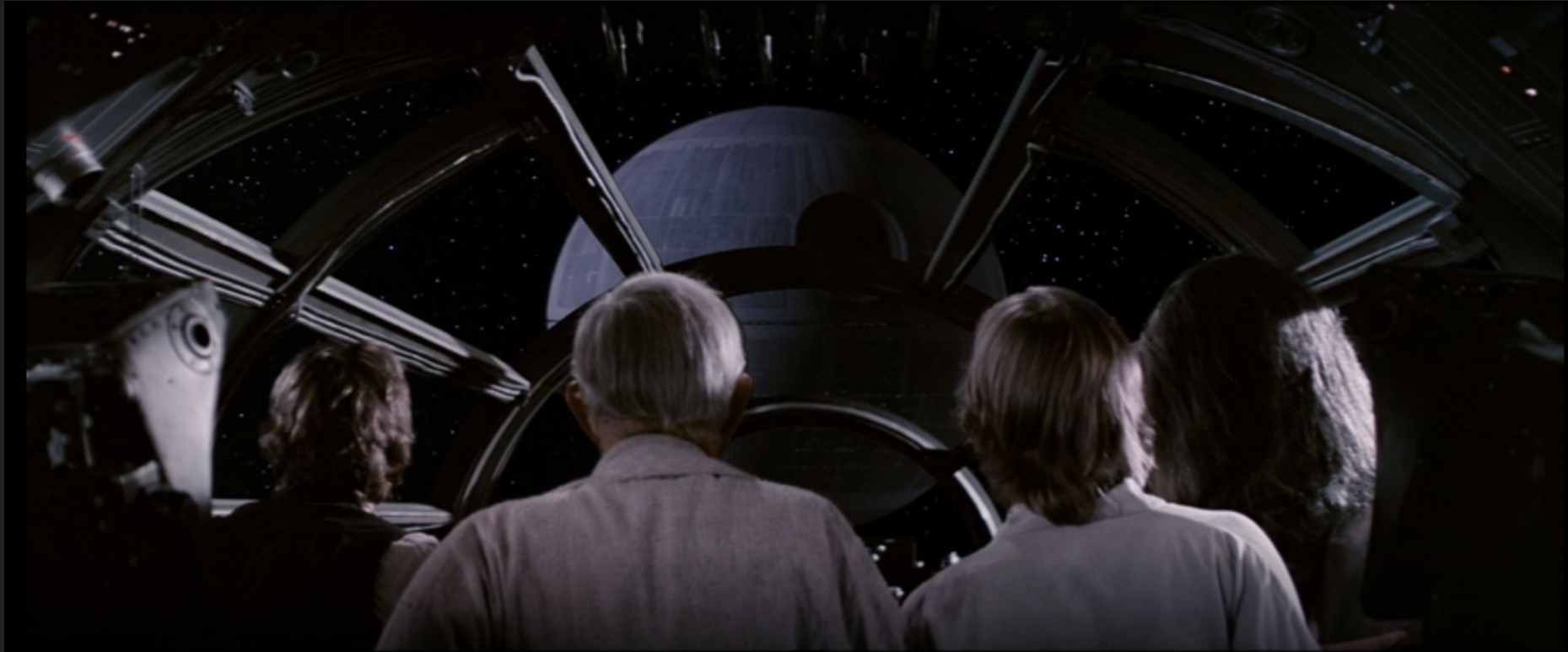
Milestone: First Playable

# Milestone: First Playable



Milestone: Vertical Slice

## Milestone: Vertical Slice



# The Challenge

- Scale of UI Development
- 150 screens, 90% of them in the frontend

Needed a Plan

# Needed a Plan

- Architecture that could scale

# Needed a Plan

- Architecture that could scale
- Help

- Introduction
- The Problem
- **UICS - User Interface Component System**
- Building Around Components
- Best Practices
- Summary

# UICS - User Interface Component System

# UICS - User Interface Component System

UI Component as a term is overloaded

# UICS - User Interface Component System

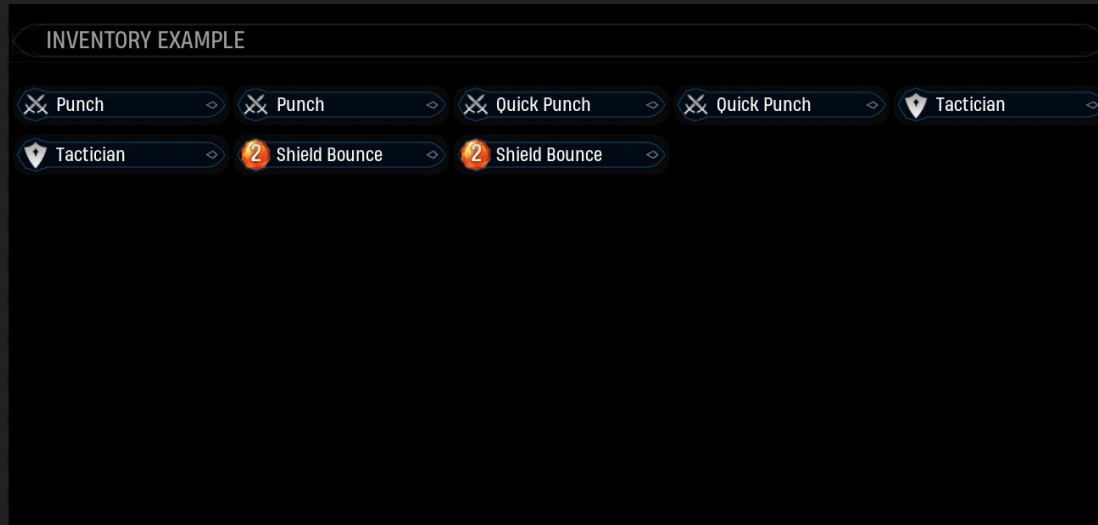
UI Component as a term is overloaded

A UI Component is something that provides functionality to a UI screen.

# UICS - User Interface Component System

UI Component as a term is overloaded

A UI Component is something that provides functionality to a UI screen.



# UICS - User Interface Component System



Screen

# UICS - User Interface Component System

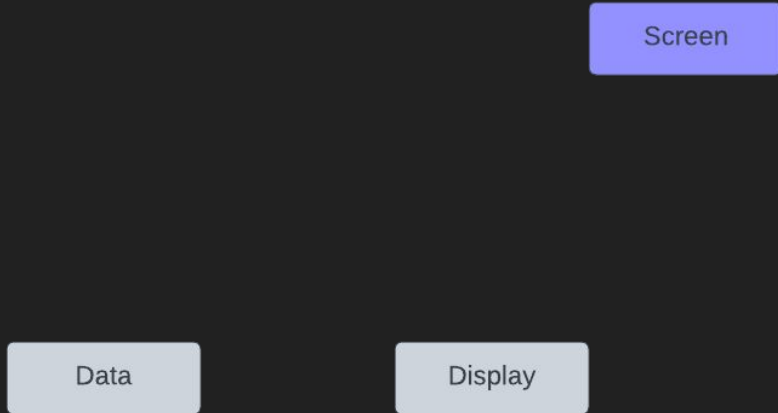
Screen

```
graph TD; Screen[Screen] --- Data[Data];
```

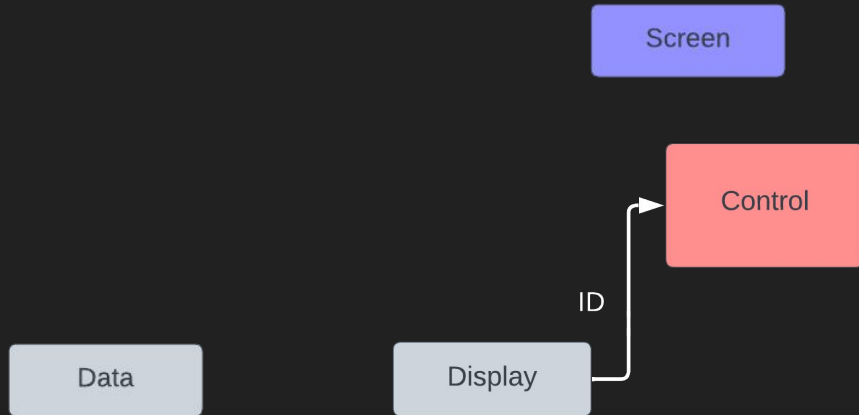
The diagram illustrates a simple system architecture. A central purple box labeled 'Screen' is connected to a light blue box labeled 'Data' located at the bottom left. The connection is represented by a single line.

Data

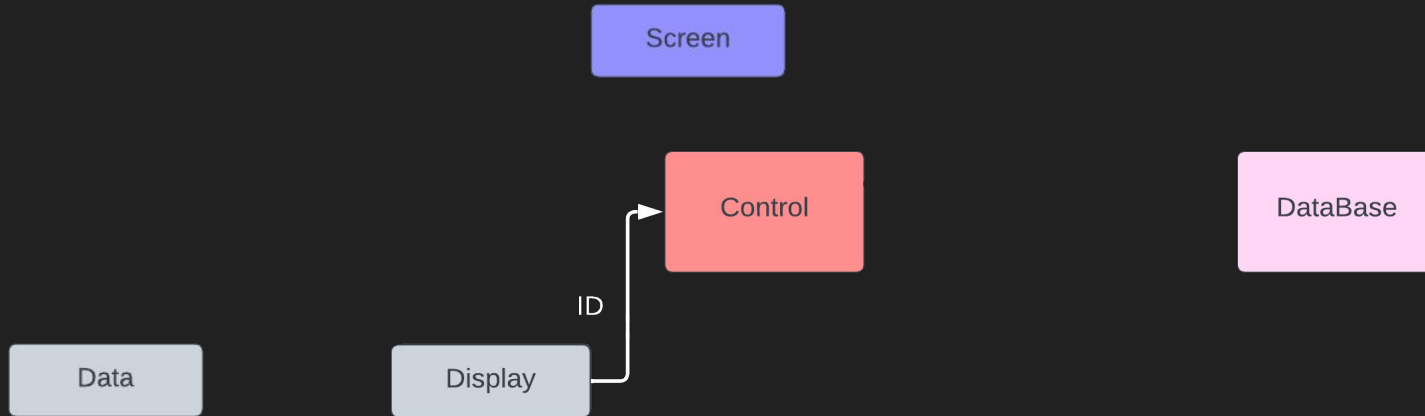
# UICS - User Interface Component System



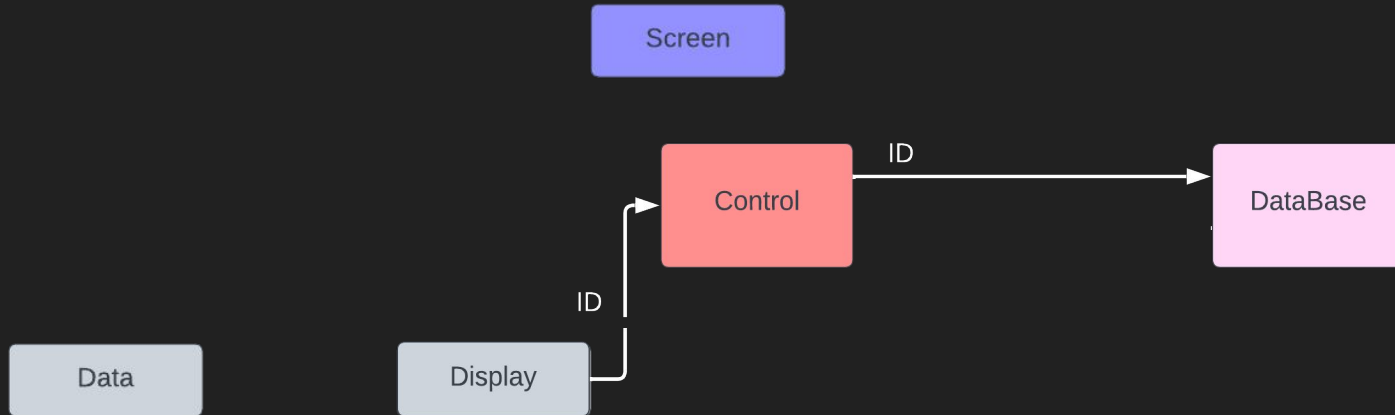
# UICS - User Interface Component System



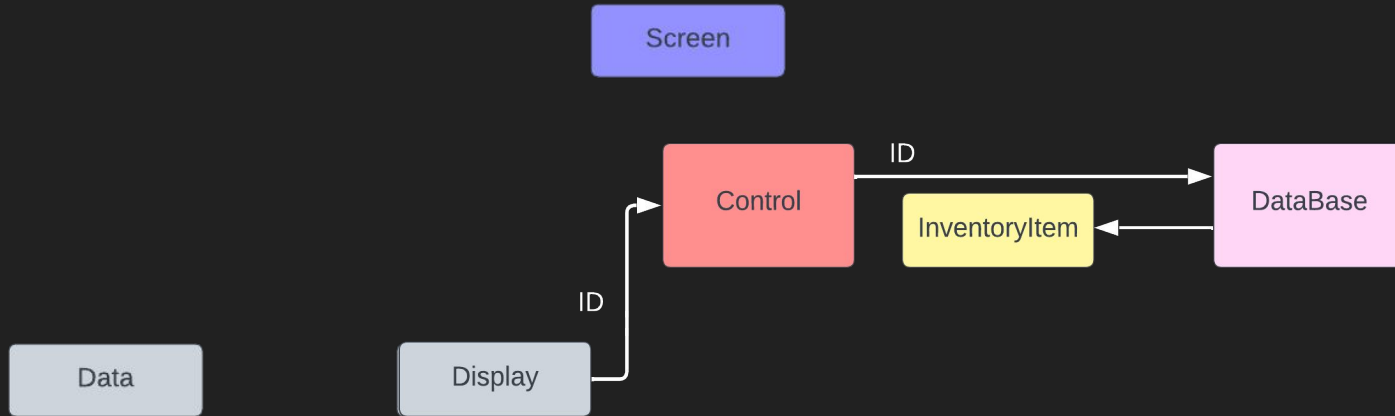
# UICS - User Interface Component System



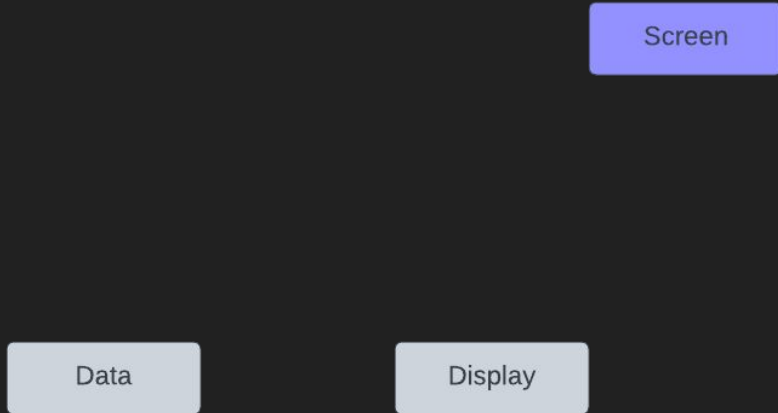
# UICS - User Interface Component System



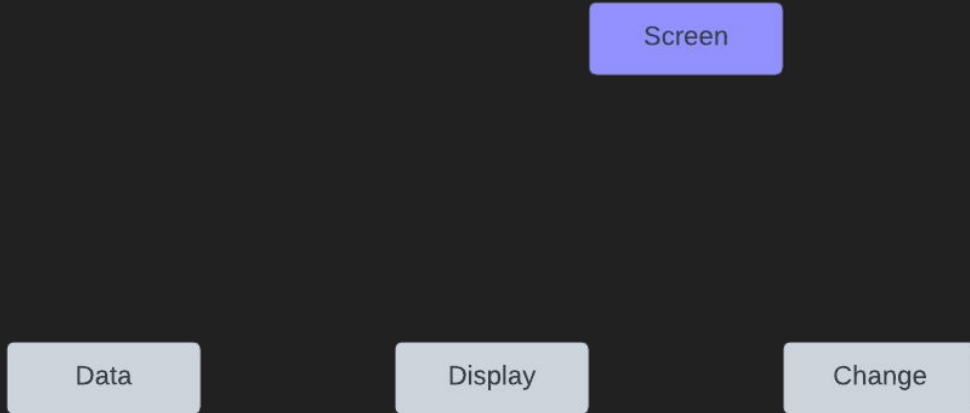
# UICS - User Interface Component System



# UICS - User Interface Component System



# UICS - User Interface Component System



# UICS - User Interface Component System

Screen

```
graph TD; Screen[Screen]; Data[Data]; Display[Display]; Change[Change]; ID[ID]; Screen --- Data; Screen --- Display; Screen --- Change; Screen --- ID;
```

Data

Display

Change

ID

# Example: Inventory Screen

# Example: Inventory Screen

Data

# Example: Inventory Screen

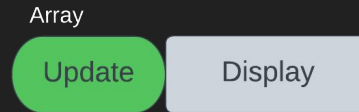
Data

Display

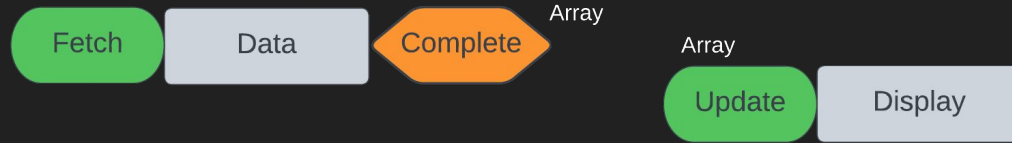
# Example: Inventory Screen



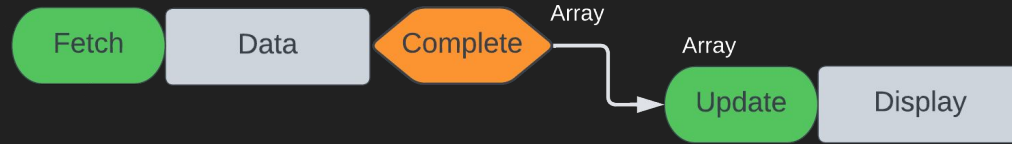
# Example: Inventory Screen



# Example: Inventory Screen



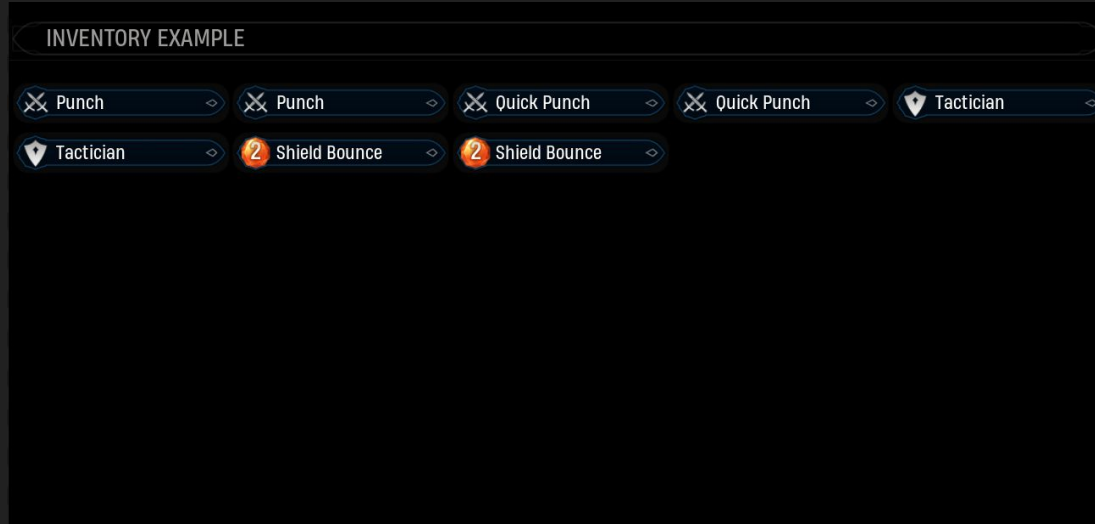
# Example: Inventory Screen



# Example: Inventory Screen



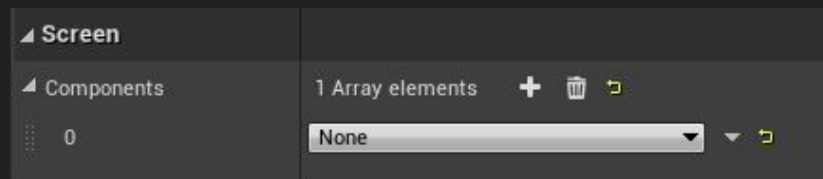
# Example: Inventory Screen



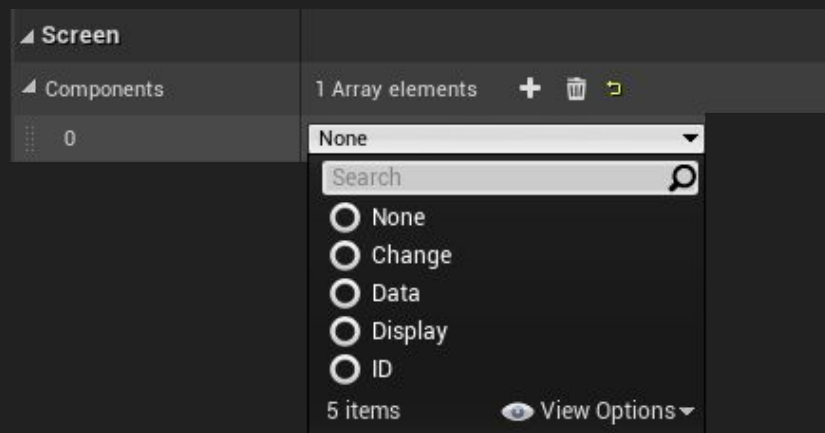
# Editor View

▲ Screen	
Components	0 Array elements + 🗑

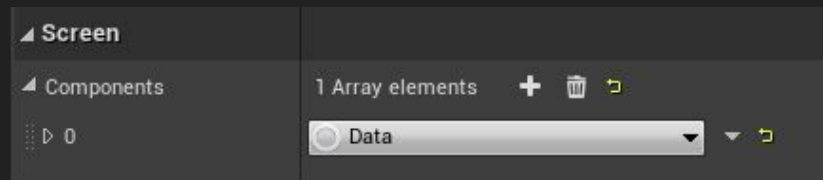
# Editor View



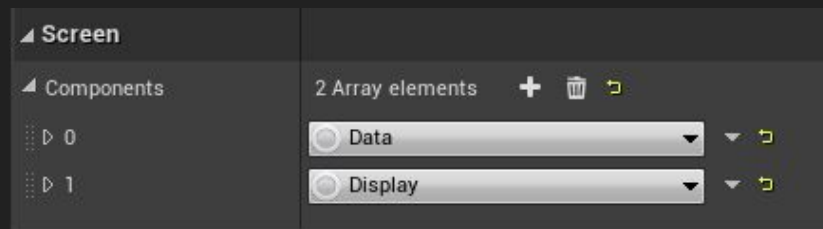
# Editor View



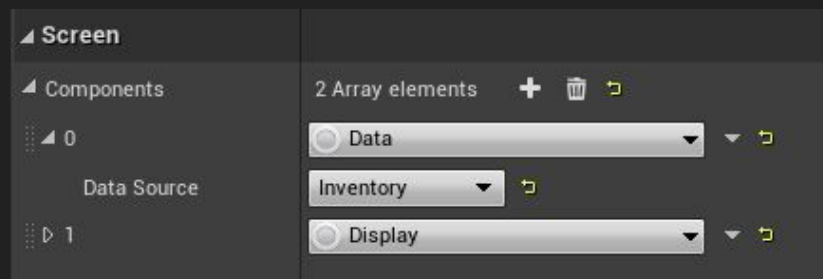
# Editor View



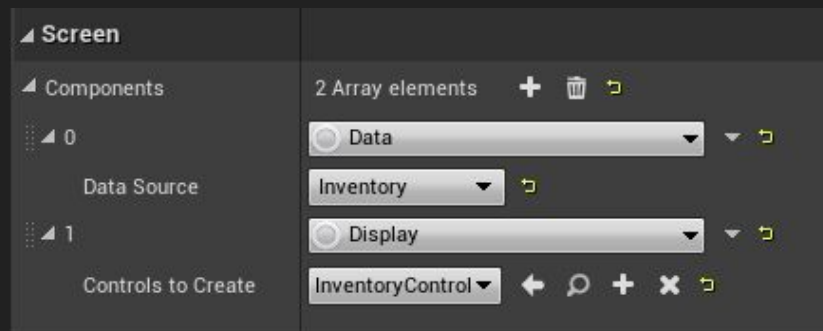
# Editor View



# Editor View

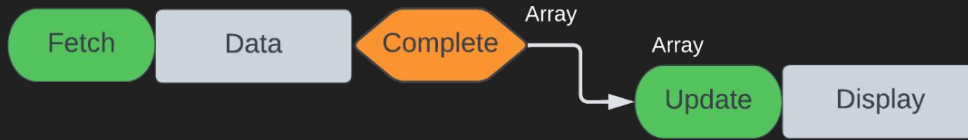


# Editor View

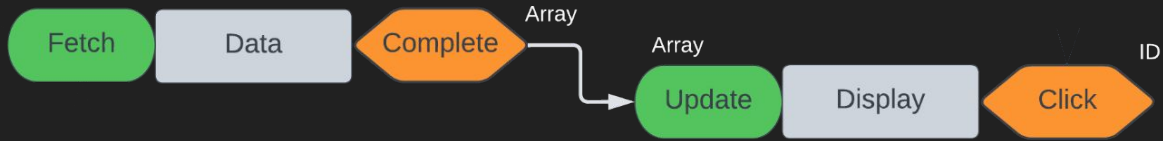


## Example: Inventory Screen With Removal

# Example: Inventory Screen With Removal



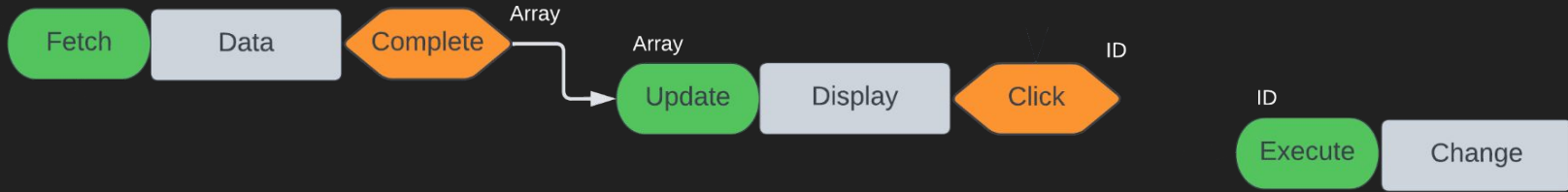
# Example: Inventory Screen With Removal



# Example: Inventory Screen With Removal



# Example: Inventory Screen With Removal



# Example: Inventory Screen With Salvage



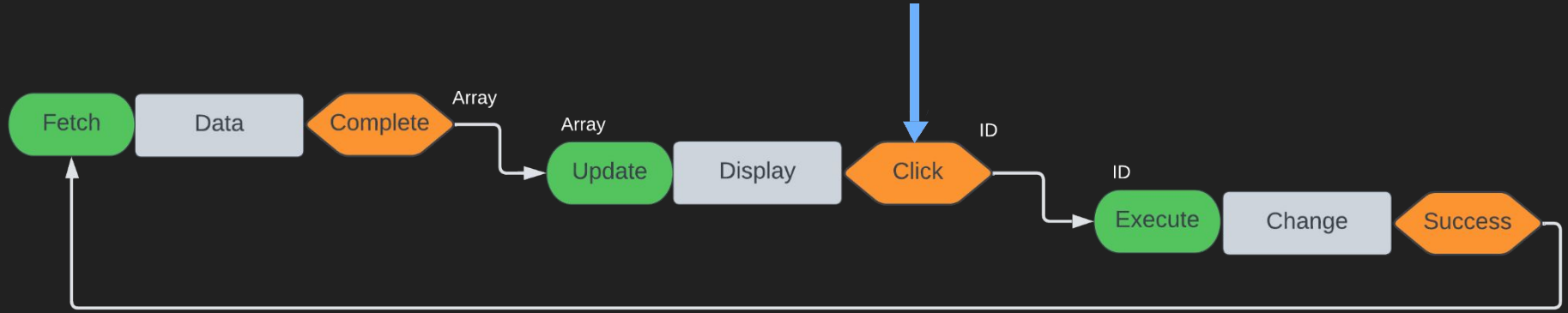
# Example: Inventory Screen With Removal



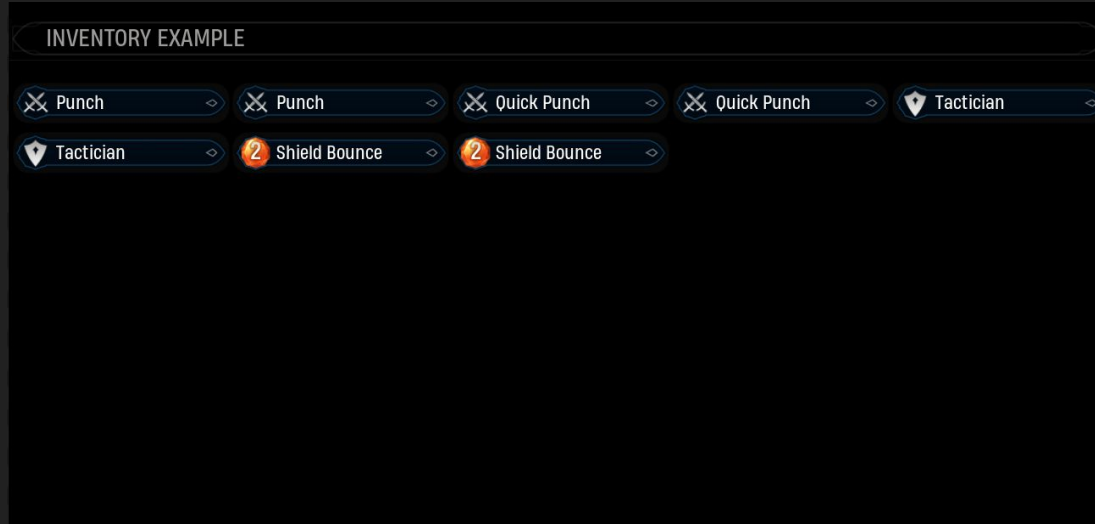
# Example: Inventory Screen With Removal



# Example: Inventory Screen With Removal

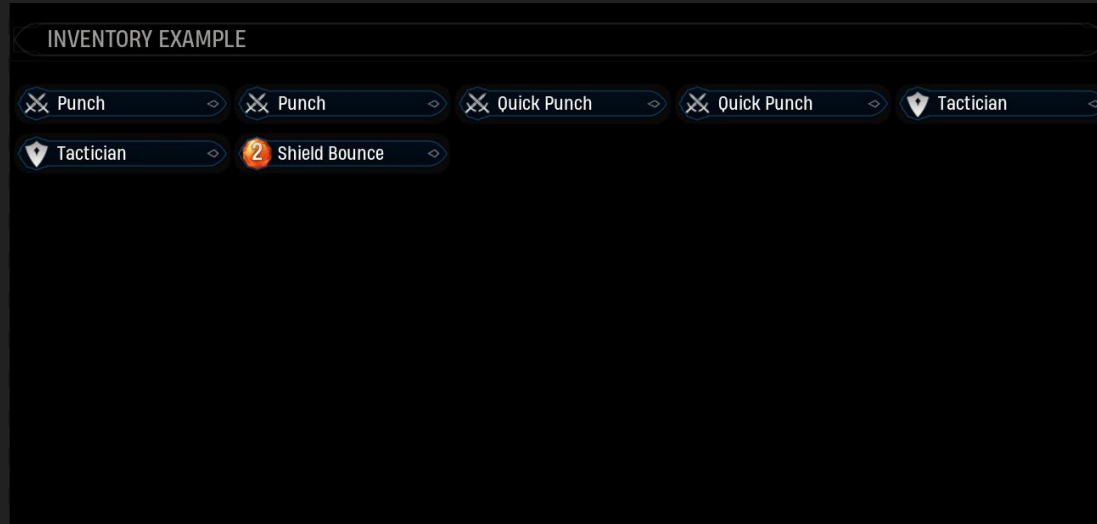


# Example: Inventory Screen With Removal



\* Not Final

# Example: Inventory Screen With Removal



\* Not Final

▲ Screen

▲ Components

⋮ ▷ 0

⋮ ▷ 1

2 Array elements

+

🗑

↺

○

Data

▼

▼

↺

○

Display

▼

▼

↺

▲ Screen

▲ Components

⋮ ▷ 0

⋮ ▷ 1

⋮ ▷ 2

3 Array elements

+

🗑

↺

○

Data

▼

▼

↺

○

Display

▼

▼

↺

○

Change

▼

▼

↺

▲ Screen

▲ Components

⋮ ▷ 0

⋮ ▷ 1

⋮ ▲ 2

Change to Execute

3 Array elements + 🗑️ ↺

☐ Data

▼

▼ ↺

☐ Display

▼

▼ ↺

☐ Change

▼

▼ ↺

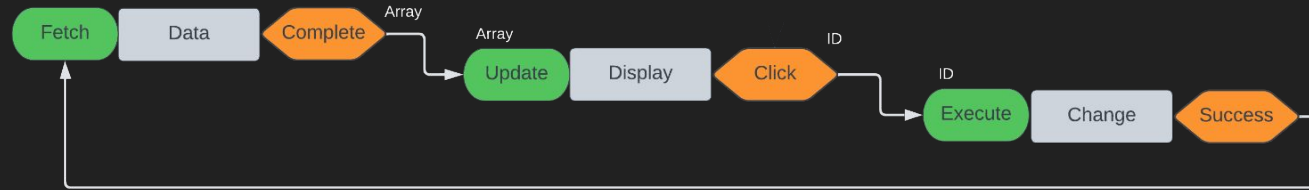
Remove

▼

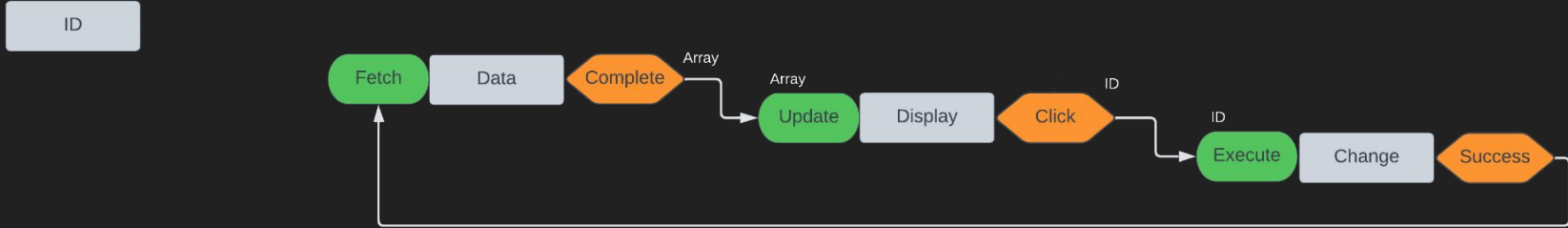
↺

## Example: Inventory Per Character

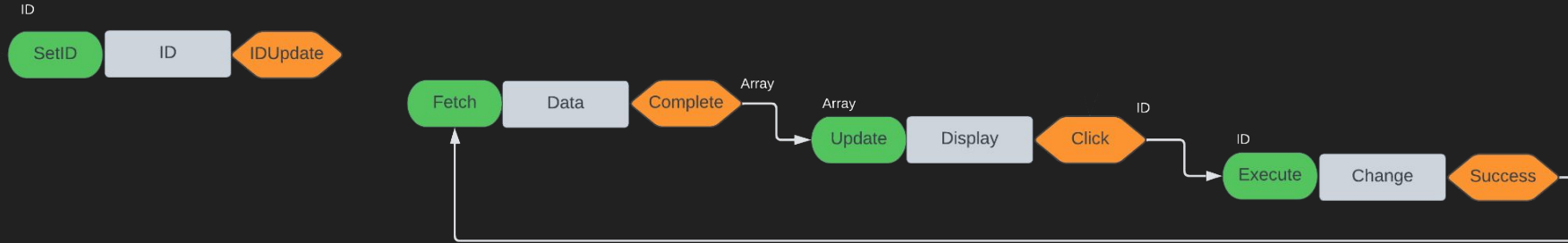
# Example: Inventory Per Character



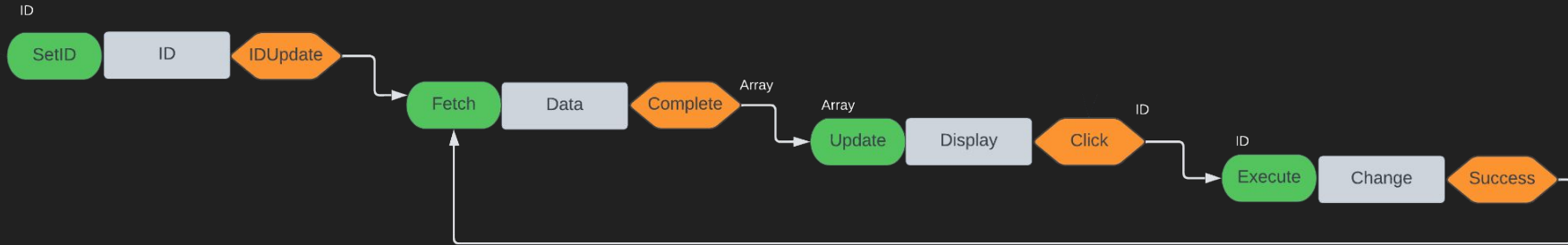
# Example: Inventory Per Character



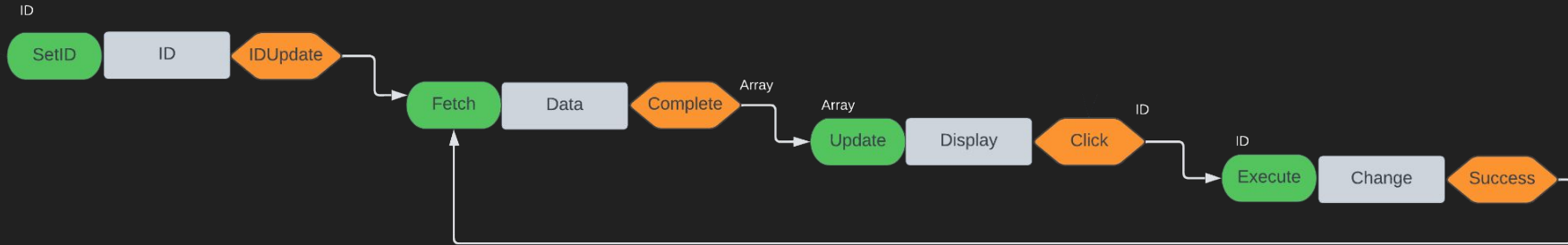
# Example: Inventory Per Character



# Example: Inventory Per Character

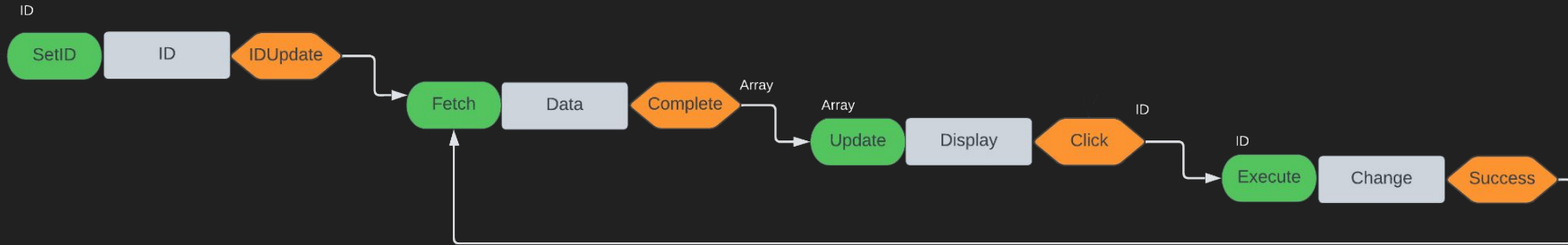


# Example: Inventory Per Character

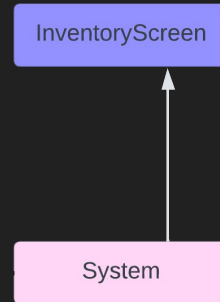


How Do we interface with a UICS?

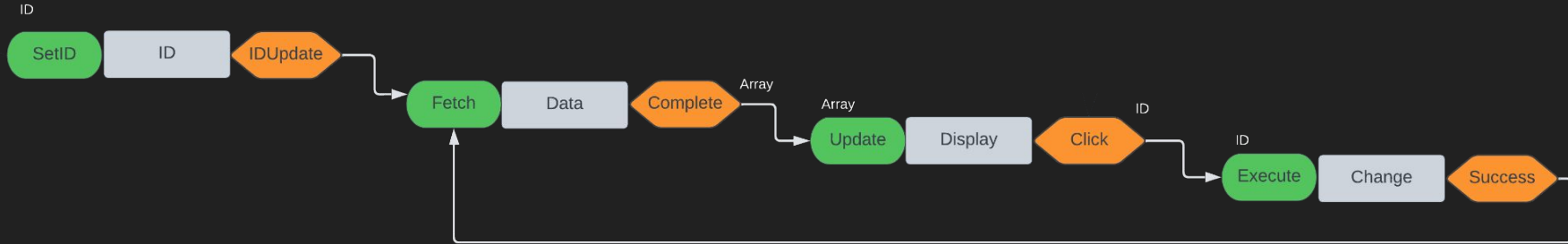
# Example: Inventory Per Character



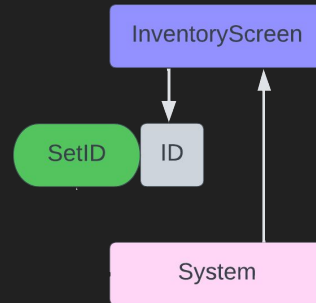
How Do we interface with a UICS?



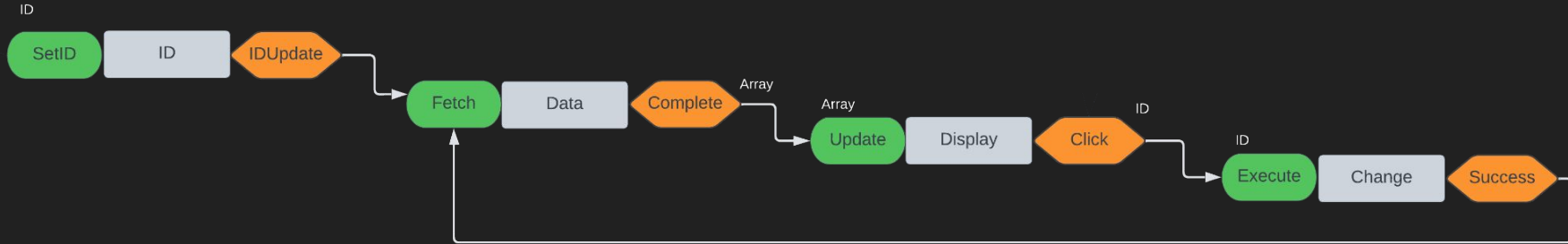
# Example: Inventory Per Character



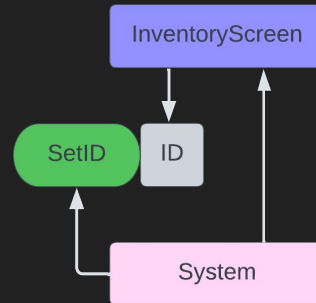
How Do we interface with a UICS?



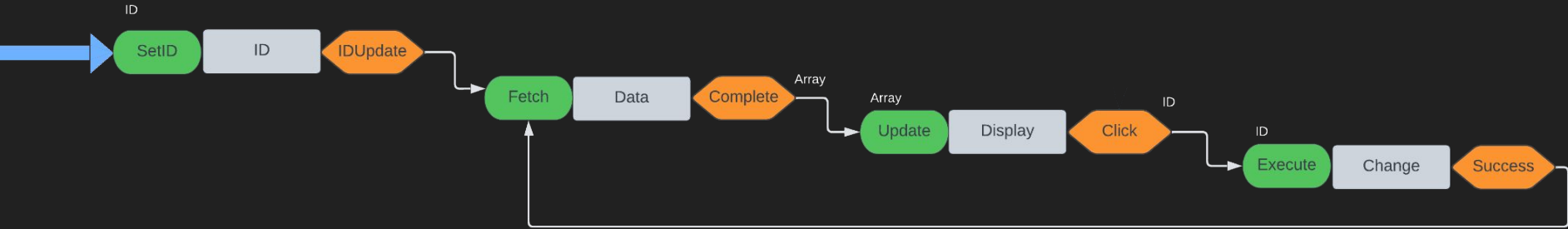
# Example: Inventory Per Character



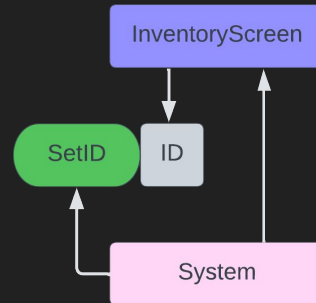
How Do we interface with a UICS?



# Example: Inventory Per Character



How Do we interface with a UICS?



# Example: Inventory Per Character



- Introduction
- The Problem
- UICS - User Interface Component System
- Building Around Components
- Summary

# Simplified Interactions



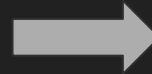
# Simplified Interactions



# Simplified Interactions



# Simplified Interactions



# Simplified Interactions

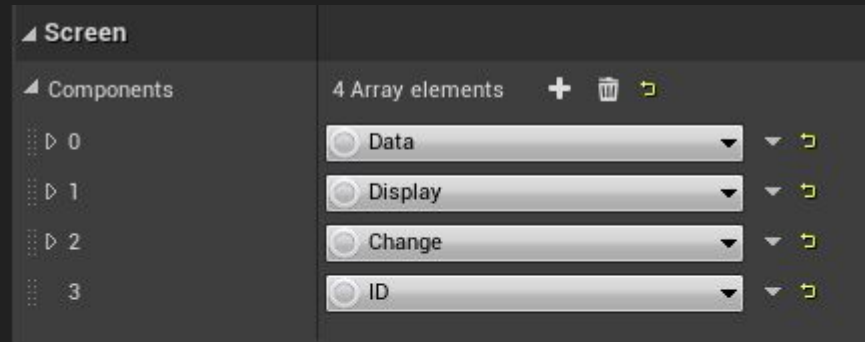


# Selecting From Multiple Components

# Selecting From Multiple Components

Listen To ID	[3] ID	↕
	0 Data	
	1 Display	
	2 Change	
	3 ID	

# Selecting From Multiple Components



# Selecting From Multiple Components

▲ Screen	
▲ Components	4 Array elements + 🗑️ ↺
⋮ ▲ 0	<input checked="" type="radio"/> Data ▼ ↺
Data Source	Inventory ▼ ↺
Change to Listen To	None ▼
ID To Listen To	None ▼
⋮ ▷ 1	<input checked="" type="radio"/> Display ▼ ↺
⋮ ▷ 2	<input checked="" type="radio"/> Change ▼ ↺
⋮ 3	<input checked="" type="radio"/> ID ▼ ↺

# Selecting From Multiple Components

▲ Screen	
▲ Components	4 Array elements + 🗑️ ↺
⋮ ▲ 0	<input type="radio"/> Data ▼ ▼ ↺
Data Source	Inventory ▼ ↺
Change to Listen To	[2] Change ▼ ↺
ID To Listen To	[3] ID ▼ ↺
⋮ ▷ 1	<input type="radio"/> Display ▼ ▼ ↺
⋮ ▷ 2	<input type="radio"/> Change ▼ ▼ ↺
⋮ 3	<input type="radio"/> ID ▼ ▼ ↺

# Selecting From Multiple Components

Screen	
Components	4 Array elements +
0	<input type="radio"/> Data
Data Source	Inventory
Change to Listen To	[2] Change
ID To Listen To	[3] ID
1	<input type="radio"/> Display
Controls to Create	InventoryControl
Data to Listen To	None
2	<input type="radio"/> Change
3	<input type="radio"/> ID

# Selecting From Multiple Components

▲ Screen	
▲ Components	4 Array elements + 🗑️ ↻
⋮ ▲ 0	<input type="radio"/> Data ▼ ↻
Data Source	Inventory ▼ ↻
Change to Listen To	[2] Change ▼ ↻
ID To Listen To	[3] ID ▼ ↻
⋮ ▲ 1	<input type="radio"/> Display ▼ ↻
Controls to Create	InventoryControl ▼ ⬅️ 🔍 + ✕ ↻
Data to Listen To	[0] Data ▼ ↻
⋮ ▷ 2	<input type="radio"/> Change ▼ ↻
⋮ 3	<input type="radio"/> ID ▼ ↻

# Visuals and Components

# Visuals and Components

- Controls can access UI Components on the screen they're attached to

# Visuals and Components

- Controls can access UI Components on the screen they're attached to
- Allows them (and their functionality) to be dropped in and shared easily.

# Control Example: Observer



# Control Example: Observer

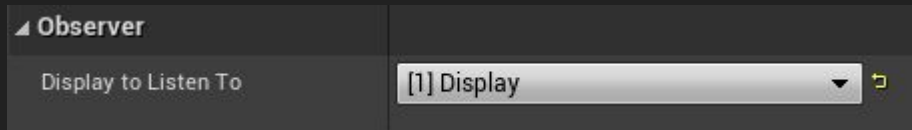


Observer

Display to Listen To

None

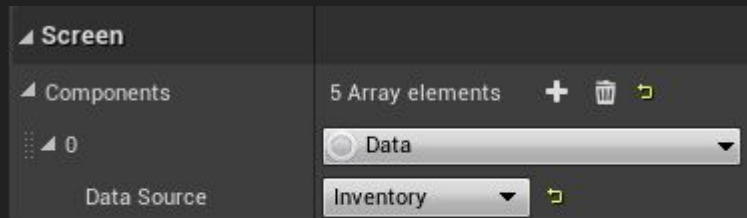
# Control Example: Observer



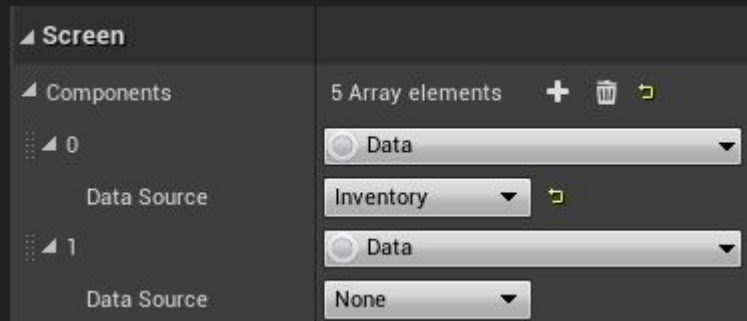
# Control Example: Cyclor



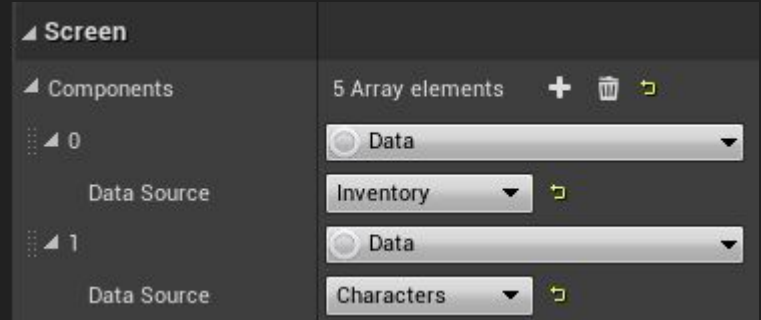
# Control Example: Cyclor



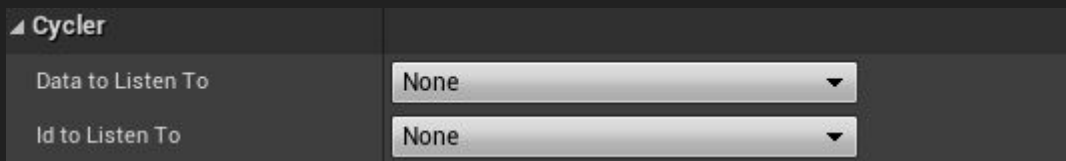
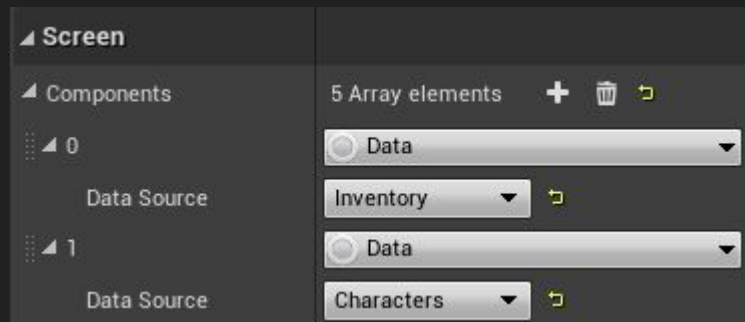
# Control Example: Cyclor



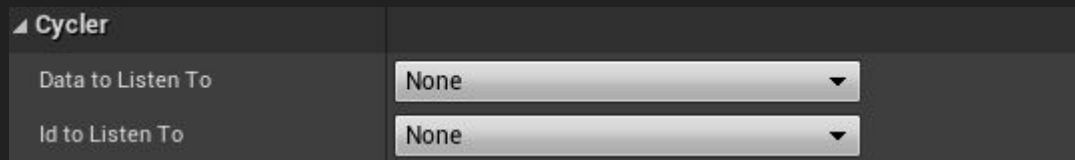
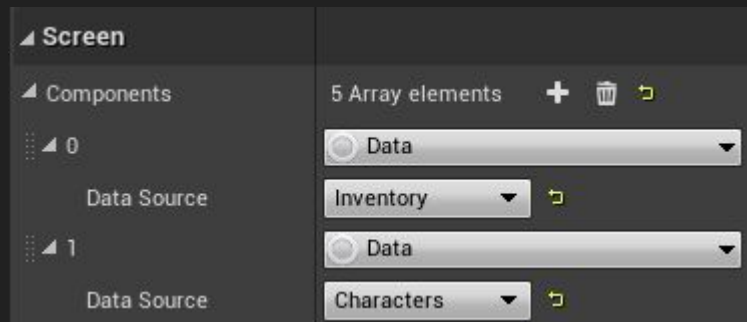
# Control Example: Cyclor



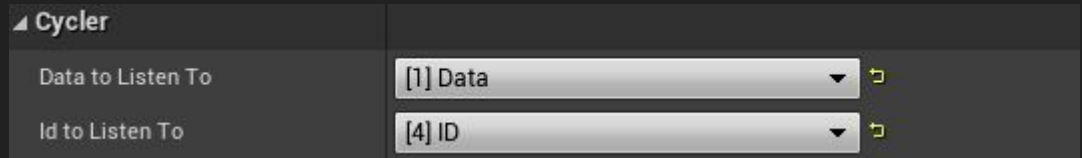
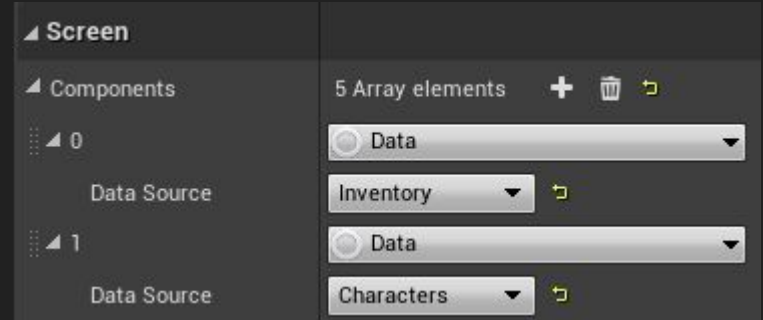
# Control Example: Cyclor



# Control Example: Cyclor



# Control Example: Cyclor



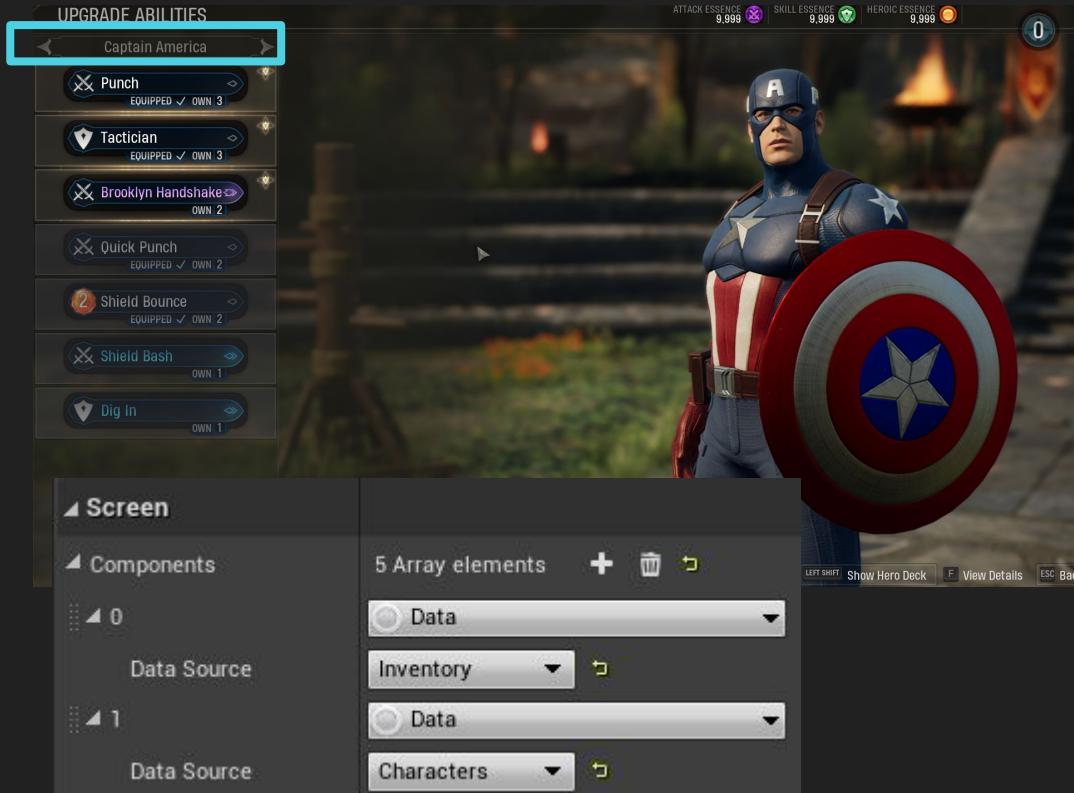
# Control Example: Cyclor



# Control Example: Cyclor



# Control Example: Cyclor



# Control Example: Cyclor

The image shows a screenshot of the Marvel Avengers game interface. The main menu is titled "UPGRADE ABILITIES" and is for the character "Captain America". The menu lists several abilities with their respective levels and ownership status:

- Punch (EQUIPPED ✓ OWN 3)
- Tactician (EQUIPPED ✓ OWN 3)
- Brooklyn Handshake (OWN 2)
- Quick Punch (EQUIPPED ✓ OWN 2)
- Shield Bounce (EQUIPPED ✓ OWN 2)
- Shield Bash (OWN 1)
- Dig In (OWN 1)

At the top right, there are three essence bars: ATTACK ESSENCE (9,999), SKILL ESSENCE (9,999), and HEROIC ESSENCE (9,999). A "0" icon is also visible.

Overlaid on the right side is a control menu for the "Cyclor" character. It features two dropdown menus:

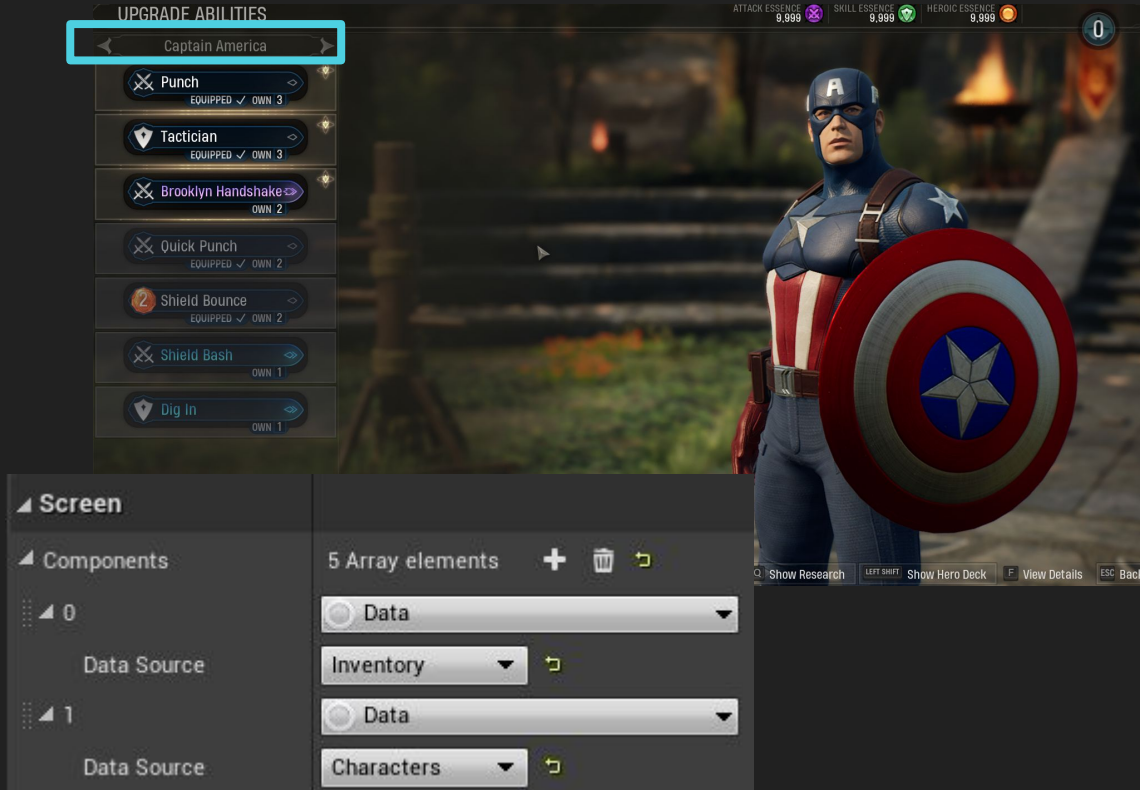
- Data to Listen To: None
- Id to Listen To: None

At the bottom left, there is a "Screen" panel with a "Components" section. It shows a list of components with their data sources:

- Component 0: Data Source (Data)
- Component 1: Data Source (Inventory)
- Component 2: Data Source (Data)
- Component 3: Data Source (Characters)

The background of the game shows Captain America standing in a field with a target symbol on his chest.

# Control Example: Cyclor



# Control Example: Cyclor

The image shows a screenshot of the Marvel Avengers game interface. On the left, the 'UPGRADE ABILITIES' menu for Captain America is open, listing abilities like Punch, Tactician, Brooklyn Handshake, Quick Punch, Shield Bounce, Shield Bash, and Dig In. The 'Cyclor' control panel is visible at the bottom, showing two data sources: 'Data' and 'Characters'. The 'Data' source is selected for both 'Data to Listen To' and 'Id to Listen To'.

**UPGRADE ABILITIES**

Captain America

- Punch (EQUIPPED ✓ OWN: 3)
- Tactician (EQUIPPED ✓ OWN: 3)
- Brooklyn Handshake (OWN: 2)
- Quick Punch (EQUIPPED ✓ OWN: 2)
- Shield Bounce (EQUIPPED ✓ OWN: 2)
- Shield Bash (OWN: 1)
- Dig In (OWN: 1)

**Screen**

**Components**

- 0 Data Source
- 1 Data Source

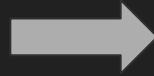
**Cyclor**

- Data to Listen To: [1] Data
- Id to Listen To: [4] ID

# Sharing Functionality between screens



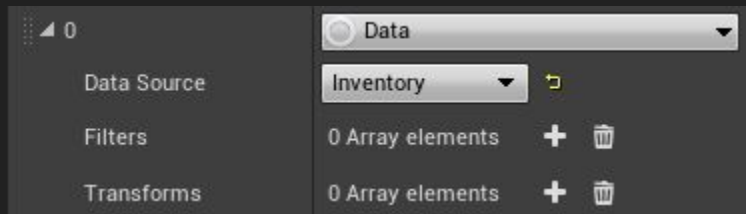
# Sharing Functionality between screens



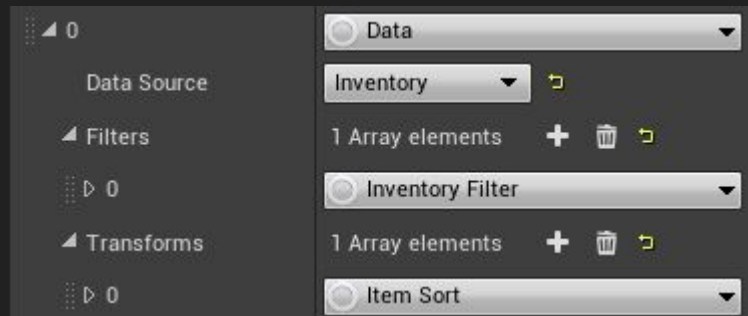
# Outline

- Introduction
- The Problem
- UICS - User Interface Component System
- Building Around Components
- **Best Practices**
- Summary

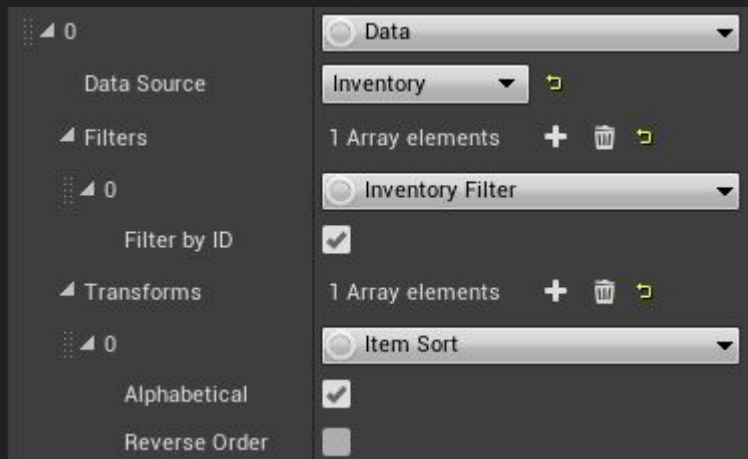
# Filtering Data



# Filtering Data



# Filtering Data



# Stage Component



# Limitations

## Bespoke Functionality



# Building & Designing UI Components

- Keep components well defined from one another
- UICS did not arrive fully formed, but updates can be easy
- Easy to find defects, few side effects

- Introduction
- The Problem
- UICS - User Interface Component System
- Building Around Components
- Summary

What did we learn with UICS?

# What did we learn with UICS?

- Less effort to make changes and fix defects

# What did we learn with UICS?

- Less effort to make changes and fix defects
- Enforced consistency and provided a workflow to guide UI creation

# What did we learn with UICS?

- Less effort to make changes and fix defects
- Enforced consistency and provided a workflow to guide UI creation

# What did we learn with UICS?

- Less effort to make changes and fix defects
- Enforced consistency and provided a workflow to guide UI creation
- An investment in technology and people

# What did we learn with UICS?

- Less effort to make changes and fix defects
- Enforced consistency and provided a workflow to guide UI creation
- An investment in technology and people
- It handles Scale

How did it go for Midnight Suns?

How did it go for Midnight Suns?



# Thank You

Contact:

[joel.gonzales@turtlerockstudios.com](mailto:joel.gonzales@turtlerockstudios.com)

<http://www.lowpolycount.com>

Special Thanks To: Chris Zimmerman, Lauren Gonzales, Firaxis UI Team

