July 6, 2021

Development Guidelines

Created by: Angeli Trinidad
Last Update:

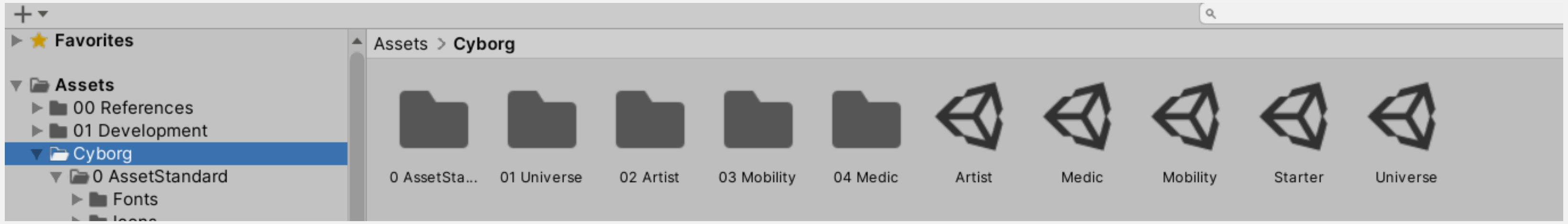
DXC.technology

Agenda

1. General Setup
2. Folder Organization
3. Naming Standards
4. Integration Strategy
5. Sprint Timeline
6. General Scripts
7. DO’s & DON’T’s

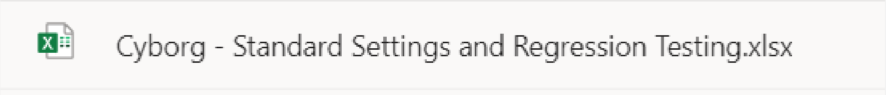
I. General Setup

- **There will be 5 scenes:**
  - Starter (where all persistent objects will be located including the final MRTK)
  - Universe
  - Artist
  - Mobility
  - Medic



- **Each scene will have a MixedRealityToolkit and MixedRealityPlaySpace ONLY for development. The MRTK will be in the *Starter* scene in the integrated version.**

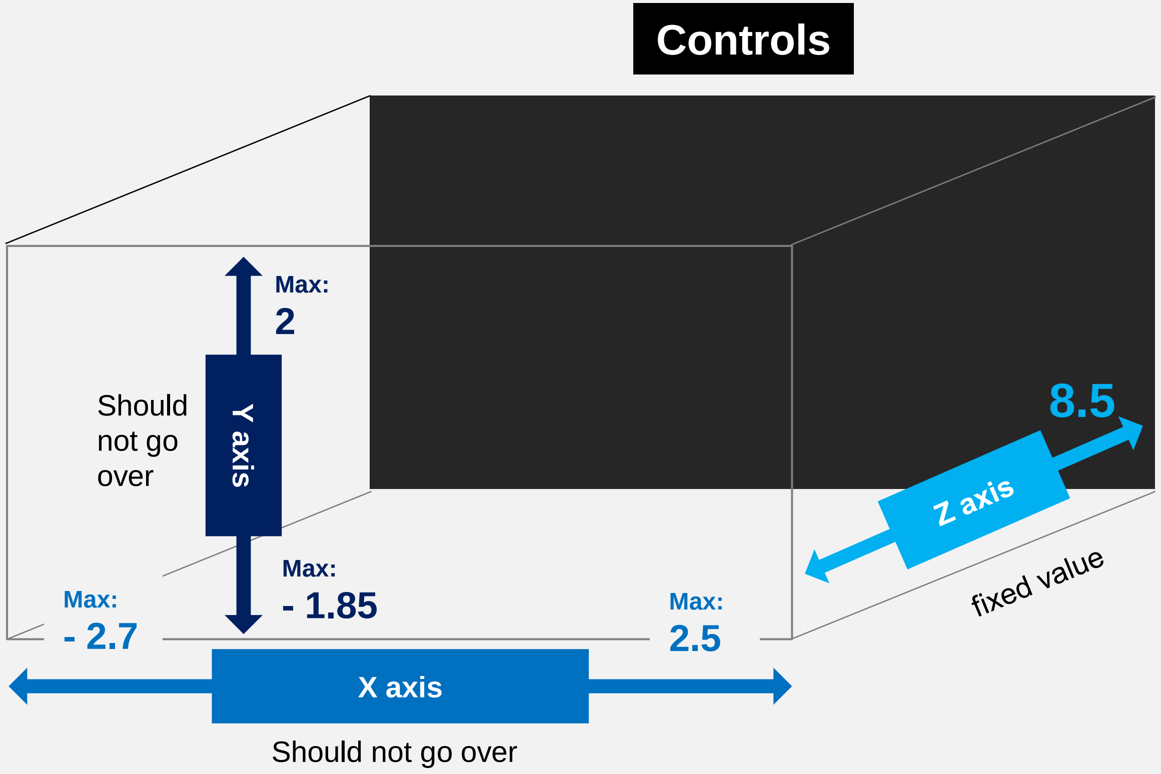
- **Ensure common setup in all scenes by following the checklist in**

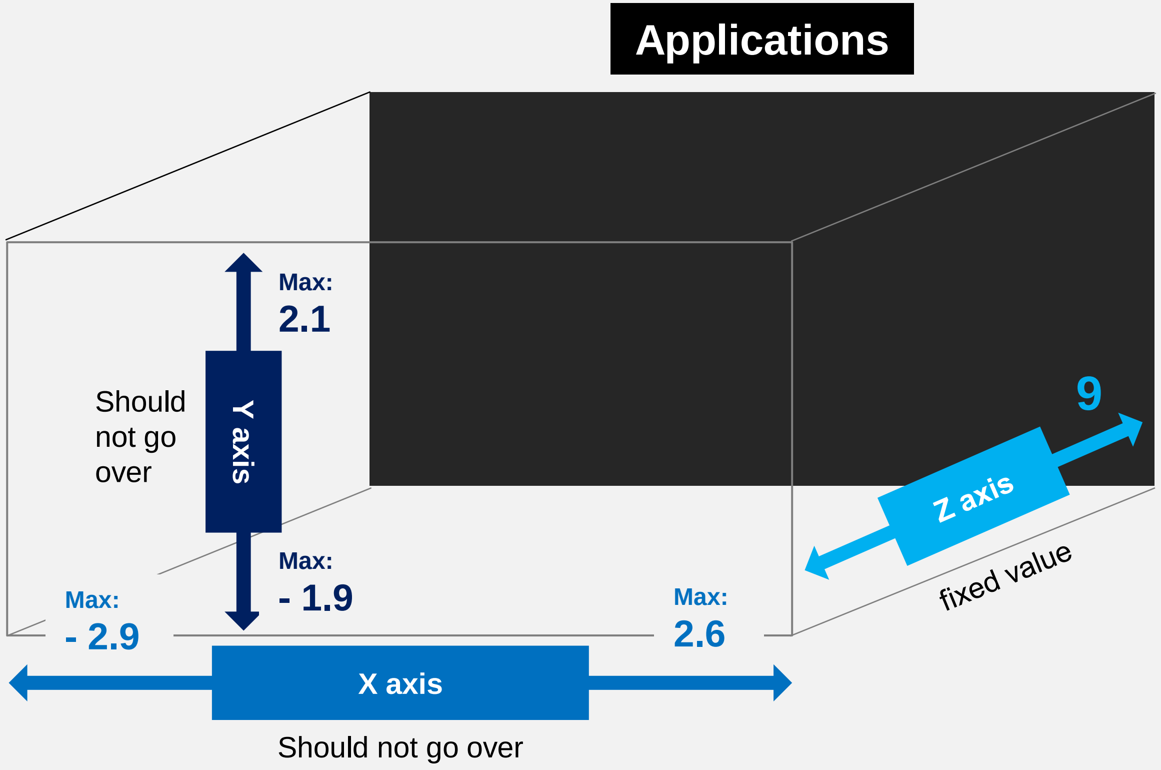


Cyborg - Standard Settings and Regression Testing.xlsx

I. General Setup

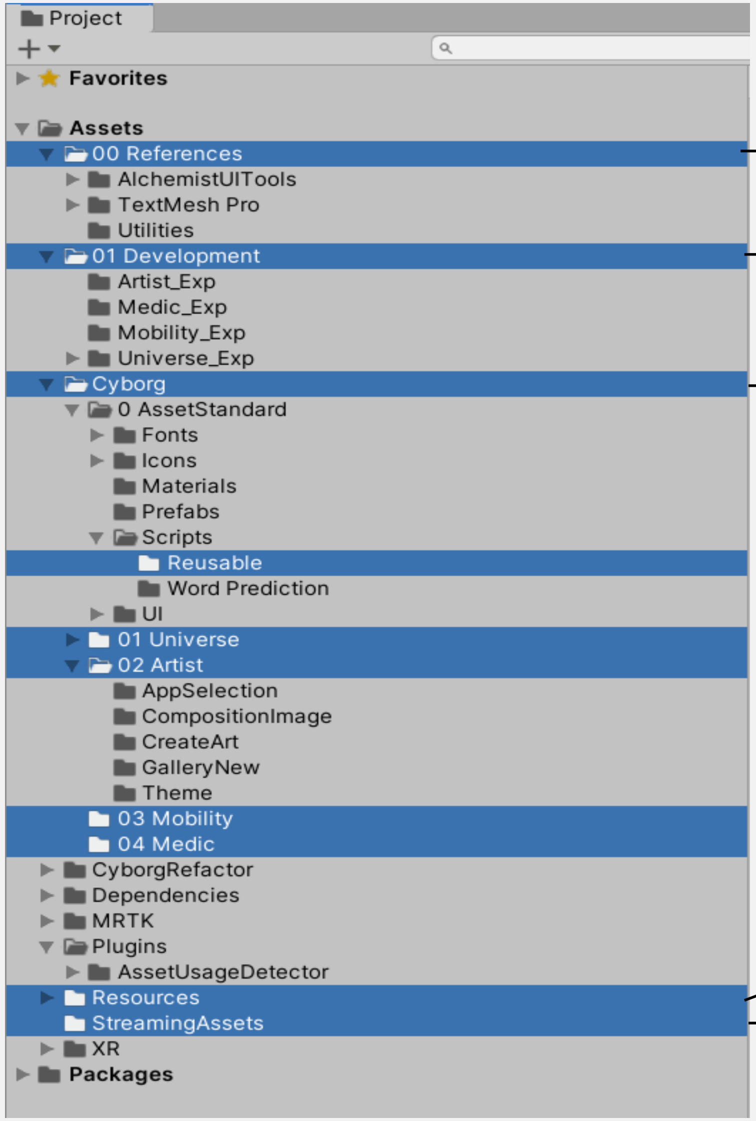
- Standard Distance from **Camera (0,0,0)**





Note: Eyegaze raycast is maximum (10)

II. Folder Organization



```
Project
▶ Favorites

▼ Assets
  ▼ 00 References
    ▶ AlchemistUITools
    ▶ TextMesh Pro
      Utilities
  ▼ 01 Development
      Artist_Exp
      Medic_Exp
      Mobility_Exp
    ▶ Universe_Exp
  ▼ Cyborg
    ▼ 0 AssetStandard
      ▶ Fonts
      ▶ Icons
        Materials
        Prefabs
      ▼ Scripts
          Reusable
          Word Prediction
      ▶ UI
    ▶ 01 Universe
    ▼ 02 Artist
        AppSelection
        CompositionImage
        CreateArt
        GalleryNew
        Theme
      03 Mobility
      04 Medic
  ▶ CyborgRefactor
  ▶ Dependencies
  ▶ MRTK
  ▼ Plugins
    ▶ AssetUsageDetector
  ▶ Resources
    StreamingAssets
  ▶ XR
▶ Packages
```

00 References - upload here all imported **third party references**

01 Development **-** all **experiments** must be located here segregated per project folder.

Cyborg - all **working functionalities and used assets (1 folder = 1 feature)** must be located here segregated per project folder.

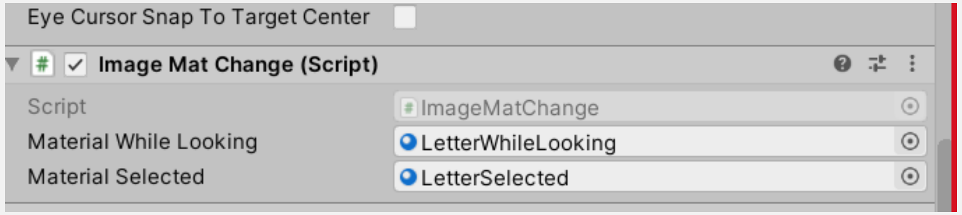
Cyborg/0 AssetsStandard- all **general assets** across the Cyborg projects shall be placed here

Cyborg/0 AssetsStandard/Scripts/Reusable – where **reusable scripts** can be found (instantiation, material change, minimize, etc.)

Resources – runtime assets. Do not store assets here if not needed because it bloats the build size

StreamingAssets – JSON files are located here (app config file, etc.)

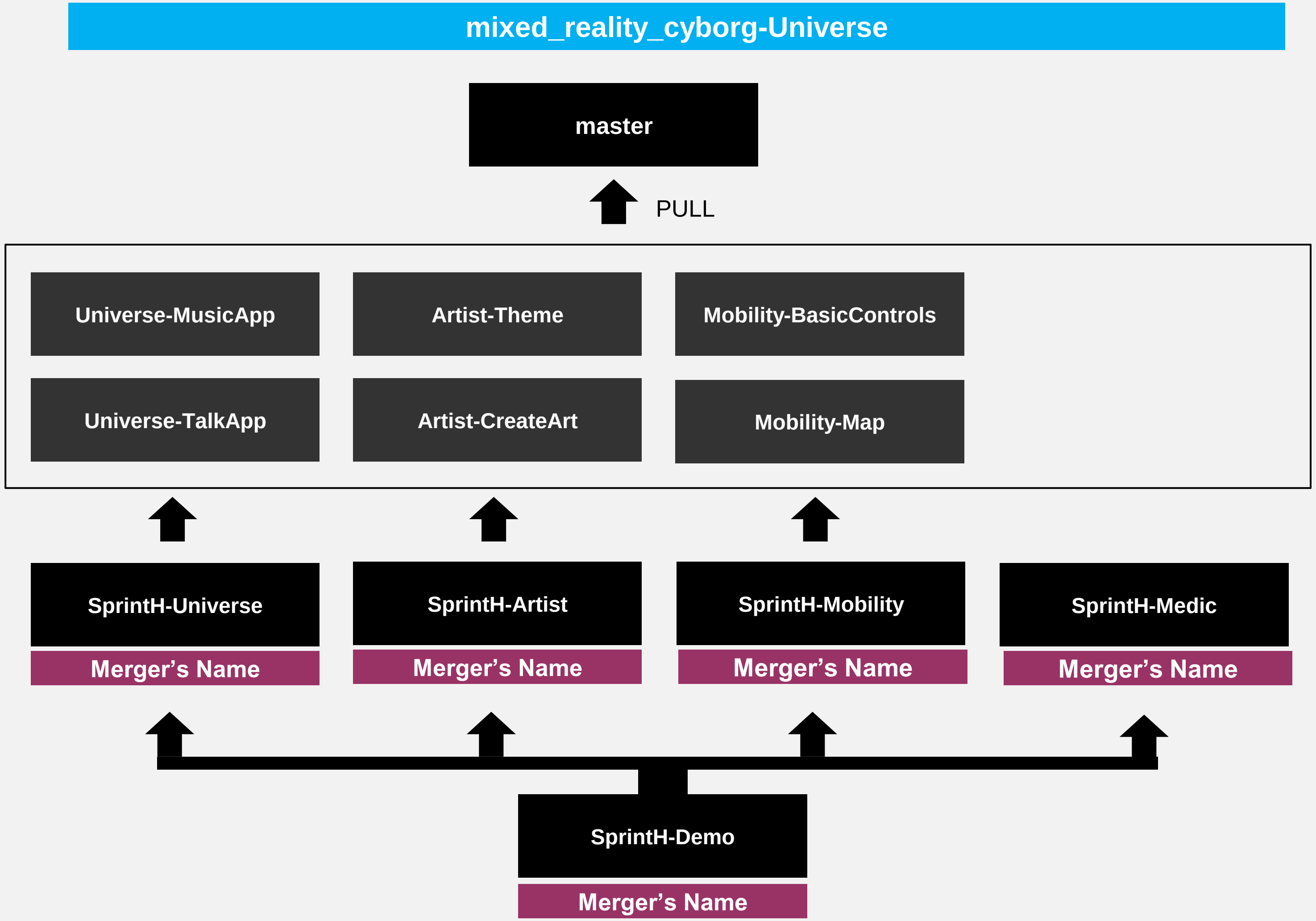
III. Naming Standards

	Format	***Example***
Branch	Project-Feature_summaryOfTask; or Project-Feature	- Universe-MusicApp_loadBar - Mobility-BasicControls - Artist-ThemeSelection
Script (verb + noun)	- UpperCamelCase - Be concise and easy to understand **(verb + noun)** - Don’t use one word only - All general reusable scripts are labeled with *ScriptName_**General***	InstantiateObject.cs
Variable	- camelCase - Be clear and easy to understand - Mention the object type if needed to easily know which object to attach (ex. Material)	materialWhileLooking objectToInstantiate  Eye Cursor Snap To Target Center ☐ ▼ # ☑ **Image Mat Change (Script)** Script: ImageMatChange Material While Looking: ● LetterWhileLooking Material Selected: ● LetterSelected

IV. Integration Strategy

- In general, more than one person should not work on the same scene or prefab because it causes conflict in GitHub.
- **Approach:**

1. Each person should work on a prefab
2. Combine prefabs into the designated scene
3. Move all final used assets from ***01 Development*** folder to ***Cyborg*** folder
4. Each team should have an assigned person responsible for integration
5. Merge changes per project
6. Merge all projects



V. Work Timeline

MON	TUES	WED	THURS	FRI
	DEMO			INTERNAL DEMO
			SCENE INTEGRATION & MAIN INTEGRATION	INTERNAL DEMO DEPLOYMENT & TESTING
FIXES	DEMO Update master branch as the latest			

IV. General Reusable Assets and Scripts (_General)

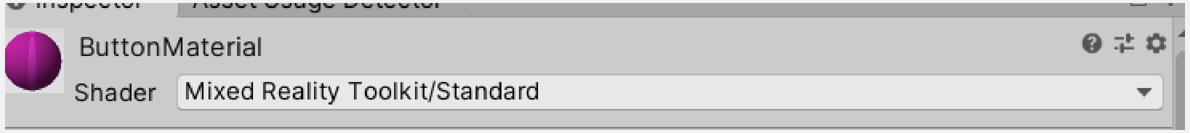
1. **Instantiate Prefab**
2. **Close** (destroys the prefab)
3. **MaterialChange** (changes the color)
4. **Pause** (tagging to sub control)
5. **Open Scene**
6. **Minimize**
7. **Back** (within app)
8. **Back to Universe** (Hides the scene)
9. **Confirmation boxes**
10. **Scroll** (prefab)

Frequent Issues

- **MRTK EyeGaze not working –**
  - There is no collider attached
  - GameObject position too far (raycast not hitting the object)
- **NullReferenceField–**
  - Object not attached to Serialized field

DO’s

- **Start using the EyeTrackingTarget right away in implementing scripts instead of pointer**
- **Add comments in your code**
- **Name variables (object type + name)**
- **Name scripts (verb + noun)**
- **In using textures, only use Standard MRTK shaders**



ButtonMaterial

Shader: Mixed Reality Toolkit/Standard

- **If you are importing .png image, change the type to Sprite**
- **Have a consistent design structure** (ex. Same location of back in all pages)
- **Public and serialize fields for easy view in Inspector**
  - Public (if needs to be accessed from other script)
  - Serialized (if private)

DON’Ts

- **Do not include UI and Logic in a single script**
- **Do not use GameObject.Find as much as possible**