O3DE UX UI SIG

Getting started and Introductions

June 24th, 2021 11-12pm (PST)

Presenters

Joshua Rainbolt *AMZN_Joshua*
Liv Erickson *Liv [AMZN]*
Yuyi Hsu *AMZN_Yuyi*

Notetaker

Lee Hung Nguyen *AMZN_Lee*

UX / UI charter agenda items

Open floor introductions ---------- 25 minutes

- Name, Your area of expertise, and company (if applicable).
- Bonus detail: What are you looking to get out of the UX charter?

Joshua ---------- 10 minutes

- What you should know first.
- What is the purpose of the UX charter.
- UX Tenets.
- Cross-cutting processes: interactions with the other SIGs.
- More about the UX charter.

Liv ---------- 10 minutes

- How is the UX team transitioning to an open source model?
- Where do all the docs live?
- How should the community start engaging with Git Hub?
- What could community engagement look like?

Yuyi ---------- 10 minutes

- Existing program: What is the UX Library and how do we use it?
- Existing program: Rev the engine.
- How do people contribute to Rev the engine?

Next steps ---------- 5 minutes

When, where, and how often will we meet.

What you should know first

- Nothing is set in stone. We are drafting this as an initial plan of action to get the ball rolling and or until we find a better way of doing things.
- These sessions should be interactive until we define a better process.
- We want your help to formulate which direction we go from here.
- We will be working from a collaborative model with the community.
- This includes design, research, feedback, and implementation.
- Everything is still in flux, so just expect that things will continue to change as we go along. Just know as things change we will keep everyone looped in.

UX Tenets

- **Emphasis on efficiency.** Creative professionals value productivity. We aspire to optimize workflows and reduce inflexibility so the creators can fully immerse in their creative flow.

- **Empower creativity**. Creative freedom is important to unleash creator's potential. We prepare the environment with building blocks that enable users to freely explore, prototype and test their ideas without imposing unnecessary limitations.

- **Foster learning.** We strive to provide an error tolerant and trustworthy environment that encourages users to master the tools with confidence, and use progressive disclosure to promote good decision making and reduce cognitive load.

- **Encourage contribution while maintaining consistency.** We aim to provide an open environment with the design system where people can build consistent user experience with ease, yet have flexibility to fit the board spectrum of needs.

What is the purpose of the UX charter?

- Deliver consistency for our user interactions and hold a quality bar for the engine.
- Establish and maintain a UX framework, patterns, interactions, and UX libraries that our customers will use.
- Raise the awareness of user experience in O3DE by gathering customer feedback and bringing customer insights back to the community
- Set future UX direction.

Cross-cutting Processes: interactions with other SIGs.

- Collaborate with other SIG teams on feature development.
- Provide verification and approval of interface design to ensure consistency.
- Define and implement a process for review and approval the UI/UX work across all SIGs.
- Work closely with SIG chairs to ensure compliance.
- Maintain involvement across other SIGs to represents the UI/UX SIG. *(Say something if you see something wrong.)*

More about the UX charter

- The charter will meet on a Bi-weekly bases and review any emails, tickets, patterns, or updates being requested.
- We also might assign delegated members to review tickets activities with the group and determine any follow up. Based on your speciality and or background.
- This charter will hold an open office hours weekly.
- During our meetings follow up on activities for a given submission. We may review problems with the current design system in conjunction with QA an the submitter.
- Determination will be made about the submissions and reply to the email/ticket with the given determination.

Liv’s updates

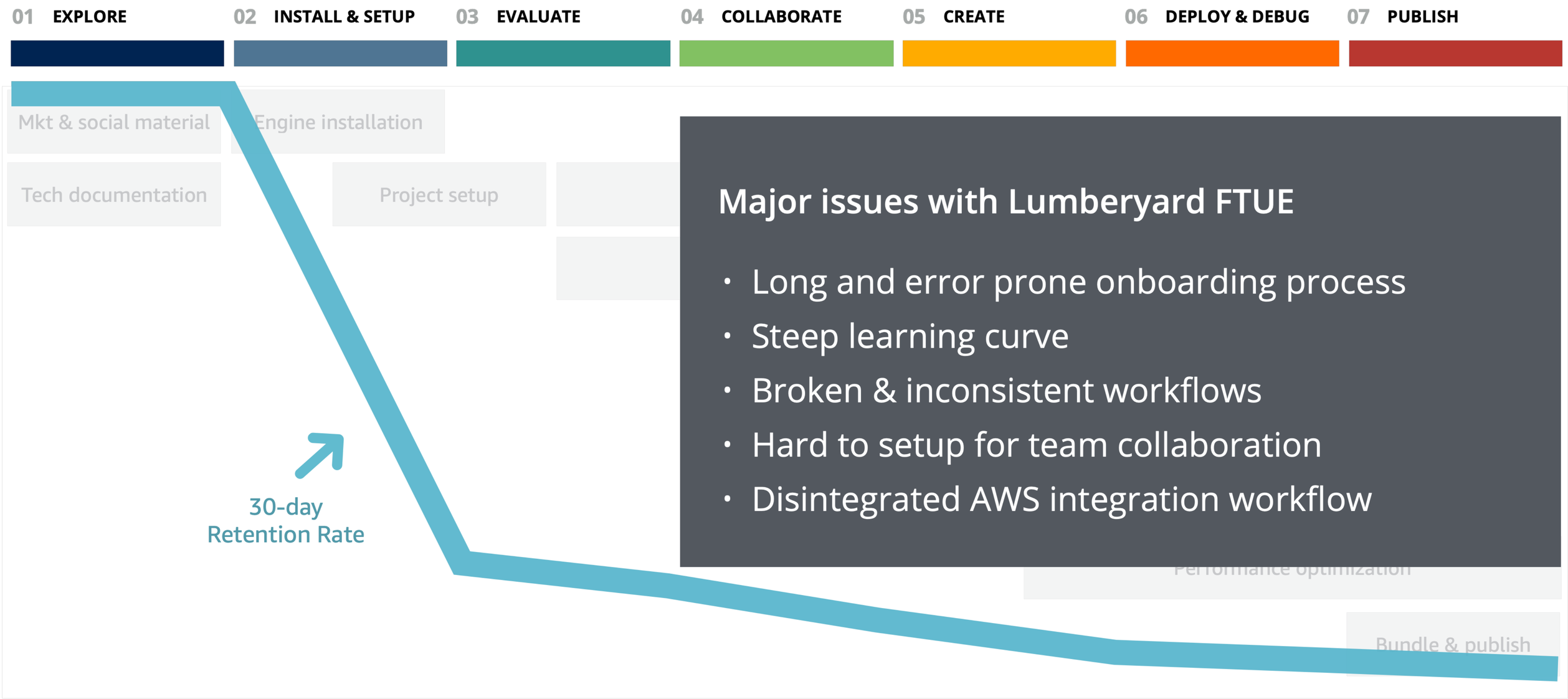
- How is the UX team transitioning to an open source model?
- Where do all the docs live?
    - https://github.com/o3de/sig-ui-ux/issues
    - https://github.com/o3de/sig-ui-ux/tree/main/governance
    - https://github.com/o3de/sig-ui-ux/tree/main/rev-the-engine
- How should the community start engaging with Git Hub?
- What could community engagement look like?

Yuyi’s updates

UX programs to drive up awareness of user experience

- Rev the Engine Program
- UI/UX Dev Library

Rev the Engine



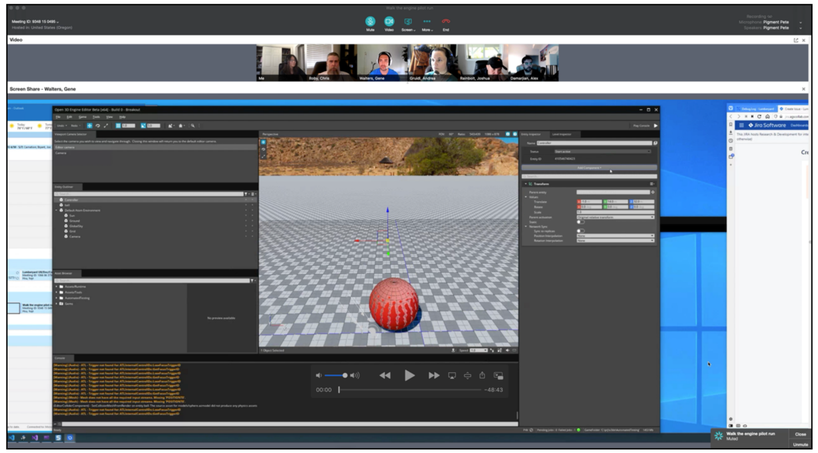
Major issues with Lumberyard FTUE

- Long and error prone onboarding process
- Steep learning curve
- Broken & inconsistent workflows
- Hard to setup for team collaboration
- Disintegrated AWS integration workflow

Rev the Engine

Day 1	Week 1	Month 1
• WF1: Onboarding experience	• WF2: Game play exploration • WF3: Actor development • WF4: World Building • WF5: Look development	• WF6: Team collaboration • WF7: Multiplayer setup • WF8: AWS integration • WF9: Editor customization • WF10: Packaging and deployment

Walk the Engine



Measurable KPI

- Number of blocker and critical issues fixed
- Time-on-task & completion rate: time (in minutes and seconds) that a user needs to complete the workflow successfully. Goal: one hour / per workflow.
- System usability score: an industry standard to evaluate the usability of a product or workflow using a 10-question survey.

SUS Score	Grade	Adjective rating
> 80.3	A	Excellent
68 ~ 80.3	B	Good
51 ~ 68	C	Poor
< 51	F	Awful

- Customer satisfaction rate

Rating system

- **Excellent (5)-** Users are able to complete the workflow within an hour.
- **Good (4)-** The workflow is easy to use. Even through it might take longer than usual for users to figure out how to
- **Passable (3)-** Users might have to read documentation to learn how to use the workflow. But they can still accomplish it within reasonable time, without encountering any blocker or critical issues.
- **Poor (2)-** Users aren't able to accomplish the workflow easily due to blocker or critical issues, or it might take them extremely long hours to accomplish the workflow with workaround solutions.
- **Failed (1)-** Users aren't able to accomplish the workflow even after reading the documentation, and there's no workaround solution

Rev the Engine Milestone 1

Workflow	Issues
WF1: Onboarding experience	1. Gem catalog UX improvement 2. Project building experience isn't optimized. 3. Lack of support for community Gems. 3. Lack of a good troubleshooting workflow.
WF2: Gameplay	1. Lack of samples to learn how to build quickly and easily. 2. Not easy to troubleshoot. 3. Inconsistent experience in viewport, hotkey, outliner, and general workspace in the editor. 4. Prefab editing. 5. Script Canvas not easy to use. 3. Lack of a good troubleshooting workflow.
WF3: Actor development	1. Take longer than expected to make a character move in the scene. 2. Lack of samples to learn how to build quickly and easily. 3. Lack of a good troubleshooting workflow.
WF4: World building	1. Lack of samples to learn how to build quickly and easily. 2. Lack of a good support system to help troubleshoot. 3. Whitebox tool isn't working 4. Lack of large world terrain, vegetation and environmental rendering and tools. Landscape Canvas isn't working (for vegetation or terrain) 5. Improved physically based lighting components and tools 6. Improved PostFX volumes, layers stack tools and workflows.
WF5: Look development	1. Lack of samples to learn how to build quickly and easily. 2. Material editor improvements 3. Lack of a good console errors and warnings support system to help troubleshoot. 4. Improved shader/materil authoring (Shader Canvas [azsl] and Material Canavs [node based material authroing tool]) 5. PostFX 'looks' and workflow improvements
WF6: Team collaboration	1. Lack of automation to package the files. The experience is error-prone. 2. No documentation supporting user to integrate with Perforce and Git.
WF7: Multiplayer setup	1. Lack of samples to learn how to build quickly and easily. 2. Prefab not loading 3. Lack of a good support system to help troubleshoot.
WF8: AWS integration	1. Poor and error-prone onboarding experience 2. No clear service status on display. 3. Lack of AWS services discoverability support 4. No team management support
WF9: Editor extension	1. Lack of Gem creation experience 2. Lack of samples to learn how to customize the editor with user's own tools or workflow quickly and easily. 3. Dispersed Python authoring experience
WF10: Packaging & deployment	We are generally lacking of support on this workflow.

Rev the Engine Milestone 2

- Extend the use cases for each workflow to Month 1
- Raise the quality rating for all workflows to Passable (3 out of 5 rating) by GA
- Engage with the community
  - Run walk the engine evaluation with the community to reassess the ratings
  - Game jams
  - Apply the UX measurement framework with Time-on-Task

UI/UX Dev Library

UI Components

35+ UI components with standardized guidelines and use cases.

UX Patterns

10+ common UX workflows, interaction flows and writing best practices.

Brand Guides

Brand value, principles and identify.

Icon Assets

Icon design templates and icon assets in svg format.

Code Snippets

Implementation and code-style guidelines for easy adoption of the UI components.

APIs

UI extension C++ API reference for the UI components.

UI/UX Dev Library

aws | Search in this guide | English | Sign In to the Console

AWS > Documentation > Amazon Lumberyard > UI Extensions Guide | Feedback | Preferences

- Control gallery tool
- Component development guidelines
- UI component library
  - Breadcrumbs
  - Browse edit
  - Button styles
  - Card
  - Checkbox
  - Combobox
  - Context menu
  - Filtered search
  - Line edit
  - Number edit spinbox
  - Progress indicators
  - Radio button
  - Reflected property editor
  - Scrollbar
  - Slider and slider combo
  - Styled dock
  - Tab
  - Table view
  - Toggle switch
  - Tree view
- UI icon assets
- C++ API reference index

Lumberyard UI card component

PDF

Use cards to display information in highly interactive, flexible containers. Users can easily stack, reorder, and collapse these containers. All content within a card should relate to only one idea. Cards should be easy to scan for relevant and actionable information, and are predominantly used to display editable details for a component or a class.

For example, a card might include common properties, action buttons, advanced settings, and a context menu for additional actions:

To make it easier for users to scan a card, make the content layou
spacing for headings and content. Be consistent with your use of
required). Card titles are key to allowing users to easily scan the c

Anatomy of the card widget

Cards allow for a certain amount of customization. The basic layo

On this page

- Anatomy of the card widget
- Basic card
- Card with context menu and help icon
- Card with secondary content
- Card with modified content
- Disabled card
- Mock disabled card
- Card with warning state
- Card with notification


```
#include <AzQtComponents/Components/Widgets/Card.h>
#include <AzQtComponents/Components/Widgets/CardHeader.h>

// Create the card.
AzQtComponents::Card* card = new AzQtComponents::Card(parent);

// Set the card icon.
card->header()->setIcon(QIcon(QStringLiteral(":/stylesheet/img/some_icon.svg")));

// Set the card title.
card->setTitle("Example Card");

// Disable the context menu (enabled by default).
card->header()->setHasContextMenu(false);

// Add the content widget.
card->setContentWidget(new QWidget());

// Add the card to a UI layout as needed.
```

UI/UX Dev Library

aws | Search in this guide | English | Sign In to the Console

AWS > Documentation > Amazon Lumberyard > UI Extensions Guide | Feedback | Preferences

guidelines

- UI component library
  - Breadcrumbs
  - Browse edit
  - Button styles
  - Card
  - Checkbox
  - Combobox
  - Context menu
  - Filtered search
  - Line edit
  - Number edit spinbox
  - Progress indicators
  - Radio button
  - Reflected property editor
  - Scrollbar
  - Slider and slider combo
  - Styled dock
  - Tab
  - Table view
  - Toggle switch
  - Tree view
- UI icon assets
  - Component icons
  - Entity outliner icons
- C++ API reference index

Lumberyard UI Extensions C++ API Reference index

PDF

This table lists the available C++ classes for working with the Lumberyard UI 2.0 system to extend the editor and tools. The C++ API reference is hosted outside of the user guide, so use this index to quickly find what you need to look up on the external site. Each class is also accompanied by a brief description.

A complete guide to the UI API can be found in the Lumberyard UI Extensions C++ API Reference. The custom Qt widget library includes the following components:

Class	Description
AssetBrowserFolderPage	
AzQtComponents::AssetFolderListView	
AzQtComponents::AssetFolderThumbnailView	
AzQtComponents::AutoCustomWindowDecorations	
AzQtComponents::AutoSettingsGroup	
AzQtComponents::AzMessageBox	
AzQtComponents::BranchDelegate	
AzQtComponents::BreadCrumbs	
AzQtComponents::BrowseEdit	A widget combining a Line Edit and a Push Button. Use this widget when you want to extend a Line Edit with support for user interaction.
AzQtComponents::ButtonDivider	
AzQtComponents::ButtonStripe	
AzQtComponents::Card	
AzQtComponents::CardHeader	
AzQtComponents::CardNotification	
AzQtComponents::CheckBox	

Next steps: Thanks for coming!

- Next meeting: Tentative date is Thursday, July 22nd 11-12 PST *(+1 month)*
  - Allow time for our first release of O3DE and closing out any urgent issues directly after launch
- Notes from todays meeting will be at https://github.com/o3de/sig-ui-ux/tree/main/meetings/notes
- Office hours are TBD and will be based on need.
- Please use the discord group and tag **Liv [AMZN]** or **Amzn_Joshua** if you have any question or any agenda items before our next meeting.