

Organization: Python Software Foundation
Sub-Organization: FURY

FURY: Create new UI widget

FURY - Software Library for Scientific Visualization in Python

Contents

- [About Me](#)
- [Contact Information](#)
- [Code and Project Contribution](#)
- [Project Information](#)
- [Detailed Description](#)
- [Stretch Goals](#)
- [Project Timeline](#)
- [Commitment and Availability](#)

About Me

Brief Introduction

Hi, I am Antriksh Misri. I am a third year Computer Science student at Maharashtra Institute of Technology, Pune.

I have been using python and related frameworks for my personal/professional projects for over 3 years now. The major frameworks/modules that I have worked with are Flask, Numpy, Opencv, Pygame etc. and I have even developed my own CLI. Other than python I am proficient in C#, Javascript, Dart. I like to develop native/web based apps using Javascript/Dart based frameworks like ReactJS, Flutter.

On the professional front, I have worked on a research based project whose aim was to extract viens from the human hand in real time. The project used modules like Opencv, skimage to perform various operations on live video feed. The operations were backed by parallel processing to make it optimized enough to work efficiently for live video feed.

In terms of computer graphics I have introductory experience with VTK which is the core framework that FURY runs upon. I am currently improving/learning stuff about VTK to make even more useful and robust contributions to FURY's codebase.

Contact Information

Name	Antriksh Misri
Country	India
College	Maharashtra Institute of Technology
Degree (Pursuing)	Bachelor of Technology(CSE) 2018-2022
Email	antrikshmisri@gmail.com
Timezone	Indian Standard Time (IST)
Github	https://github.com/antrikshmisri
LinkedIn	https://www.linkedin.com/in/antriksh-misri-b631361a1/

Code and Project Contribution

Below is the list of PR's that I have worked on FURY's codebase:-

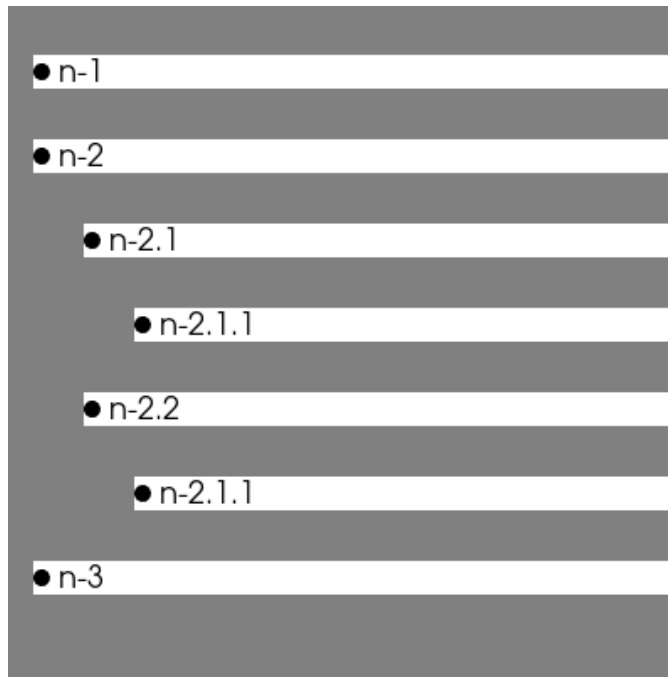
1. [#398 - Card2D widget](#)



Example of the Card2D widget

This was a **New Feature** PR in which I added a new widget to the UI Module of FURY. The widget is used to create a card with an image, a title and a body. I also added tests and tutorials for the newly added UI component.

2. [#410 - BulletList2D Widget](#)



Example of BulletList2D

This was a **New Feature** PR in which I added BulletList2D widget to the UI Module of FURY. The widget is used to create nested bullet points using a list. I also added tests and tutorials for the newly added UI component.

Below is the list of issues that I have raised in FURY's codebase:-

1. [#402 - ImageContainer2D renders .png in black and white](#)

This issue was related to the ImageContainer2D rendering RGB .png files in black and white.

Project Information

FURY

Software Library for Scientific Visualization in Python

1. Project Abstract

As per Idea's List I would be adding a couple of UI components to FURY's UI module. These 2D and 3D components will be sci-fi like as seen in [this](#) scene from the movie "Guardians of The Galaxy".

My main objective would be to develop these UI components with their respective test and tutorials such that it adds on to the already vast and rich UI module of FURY and doesn't hinder existing functionalities/performance.

The components that I will be working on are **SpinBox UI** , **Accordion UI** , **Tree UI**.

As **Stretch Goals** I would like to work on the Layout management UI component for FURY's UI module.

2. Detailed Description

1. SpinBox UI Element

A SpinBox component is used to select of a discrete value in steps or Increments/decrements. This is done by two buttons, one to increment the value (up button) and another to decrement the value (down button). The current value of the spinbox is shown in a textblock.

The SpinBox UI element can be implemented using these already existing components in the UI module:-

- a. **Button2D** - This component will be used in implementing the increment and decrement buttons of the SpinBox.
- b. **TextBlock2D** - This component will be used to show the current value of the SpinBox. The value will update when one of the two buttons are pressed

2. Accordion UI Element

An accordion UI element is essentially a vertical dropdown list that when clicked reveals the component/content that has been mapped with it. This UI element can be used to map different components to specific labels. This can come in handy in tutorials where different components are to be rendered on the screen.

The Accordion UI element can be implemented using these already existing components in the UI module:-

- a. **Panel2D** - This is used to contain all the other components.
- b. **TextBlock2D** - This is used to display the label
- c. **Button2D** - This will be used to toggle between collapsed/uncollapsed state

3. Tree UI Element

A tree UI element is used to display information according to an already set hierarchy. This means each node in that hierarchy can have a parent and **N** number of children. It is not compulsory for a node to have a parent or children, for instance a node may have one parent but no children and another node may have no parent but many children. In terms of data structures this can be considered as a **N-ary Tree**.

In terms of design each node with at least one child can be expanded or collapsed. Each child of a particular node has an indentation before it that helps us to differentiate it from the root node.

Some applications of this UI component are, directory view structure that can be seen in file managers, presenting hierarchical data such as an XML document.

The Tree UI can be implemented using the following:-

1. **Panel2D**:- This is used to containing all other components
2. **TextBlock2D**:- This is used to display the nodes
3. **Button2D**:- This will be used to implement the drop-down button, which will expand/collapse the root node
4. **TreeUI**:- Child node to a specific node

3. Stretch Goals

1. Layout Management

Layout Management is very crucial when working with many UI elements at once. Layout Management is used to determine the positions of the UI elements present inside a container. The UI elements may have a predefined position but Layout Manager can override the positions to get the desired arrangement.

This can be achieved by using already available module `layout.py` and by adding some more functionalities to it.

Project Timeline

Below is the distribution of the work that I will do per week for the official 10 weeks coding period of Google Summer of Code.

May 17, 2021 - June 7, 2021 (Community Bonding)

- Getting to know more about the community, mentors as well as admins
- Discuss the goals/vision for the above mentioned features
- Discuss the order of implementation of the mentioned features
- Research and suggest any new idea for the mentioned features
- Research and suggest new stretch goals appropriate for the codebase
- Decide the implementation order of the stretch goals

June 7, 2021 - August 16, 2021 (Coding Period)

Spinbox UI (June 7, 2021 - June 20, 2021)	
June 7, 2021 - June 13, 2021	<ol style="list-style-type: none">1. Start with the structure of the code, determine the attributes, parameters2. Create the text block to show the current value of SpinBox using <code>TextBlock2D</code> widget3. Create buttons to perform increment and decrement operations using <code>Button2D</code> UI Element4. Assign relevant callbacks to the buttons to increment/decrement value of the Spinbox UI Element.
June 14, 2021 - June 20, 2021	<ol style="list-style-type: none">5. Create relevant properties/setters for the SpinBox UI element6. Combine all the individual components together in a <code>Panel2D</code>7. Write unit-tests for the SpinBox UI element8. Create the tutorial for the SpinBox UI element9. Perform bug fixing and debugging

Accordion UI (June 21, 2021 - July 4, 2021)	
June 21, 2021 - June 27, 2021	<ol style="list-style-type: none">1. Start with the structure of the code, determine the attributes, parameters2. Create the text block to show the label text using <code>TextBlock2D</code> in the Accodion UI3. Create buttons to collapse and expand the view of the label using <code>Button2D</code> and assign relevant callbacks to the buttons in the Accordion UI4. Create relevant properties/setters for Accordion UI element
June 28, 2021 - July 4, 2021	<ol style="list-style-type: none">5. Create the <code>Panel2D</code> to contain all the labels along with their content. The content of a label should not necessarily be text, it can be any UI Element.6. Map the content to their respective labels in the Accordion UI Element7. Write unit-tests for the Accordion UI element8. Create the tutorial for the Accordion UI element9. Perform bug fixing and debugging

Tree UI/Stretch Goals (July 5, 2021 - July 20, 2021)	
July 5, 2021 - July 11, 2021	<ol style="list-style-type: none"> 1. Start with the structure of the code, determine the attributes, parameters 2. Create the text block to show the node value using <code>TextBlock2D</code> in Tree UI 3. Create drop-down button to expand/collapse the view of root node using <code>Button2D</code> in Tree UI 4. Work on the "add_node" feature that adds a child node to the root node in Tree UI. The nodes will be an instance of Tree UI.
First Evaluations (July 12 - 16, 2021)	
July 12, 2021 - July 18, 2021	<ol style="list-style-type: none"> 5. Create a <code>Panel2D</code> to contain all the nodes along with their child nodes and update relevantly on new node addition in Tree UI. 6. Optimization for the added changes (like storing node structure as dictionary) 7. Create relevant properties/setters for <code>Tree UI</code> widget
July 19, 2021 - July 25, 2021	<ol style="list-style-type: none"> 8. Combine all the individual components together in a <code>Panel2D</code> 9. Write unit tests for the <code>Tree UI</code> element 10. If time persists work on stretch goals
July 26, 2021 - August 1, 2021	<ol style="list-style-type: none"> 11. Create Tutorial for the <code>Tree UI</code> Element 12. Perform bug fixes and debugging 13. If time persists work on stretch goals

Documentation/Buffer Period (August 2, 2021 - August 16, 2021)	
August 2, 2021 - August 8, 2021	<ol style="list-style-type: none"> 1. Complete any remaining tasks and fix found bugs 2. Document all the work done during GSoC
August 9, 2021 - August 16, 2021	<ol style="list-style-type: none"> 3. Debug the changes extensively 4. Create demonstration video for the newly added features 5. Conclude with adding demos for all the changes made
Final Evaluation (August 16 - 23, 2021)	

Commitment and Availability

1. My working hours would most likely be(EST) from 9:00 a.m to 1:00 p.m and 7:00 p.m to 9:00 p.m for both weekdays and weekends, these times are flexible and can be re-planned if needed.
2. Might have to travel as my college might reopen in mid july, will take 5 days at most. I will not be having any exams for the total period of GSoC.
3. I will not be travelling/ be on vacation.
4. I will only be applying to FURY for GSoC 2021.