BioGears Visualizer

A. Baird[1], J. Carter[1], L. Marin[1], M. McDaniel[1], N. Tatum[1], S. White[1]

*1. Applied Research Associates Inc.*

Streamlined UI for experimenting with Insults/Injuries on a patient.

- Timeline:
  - Now: 0.5 – Alpha Release
    - Custom Actions
    - Simulation Controls
    - Organized Data Visualization
  - December 2020: Version 1.0
    - Improved Theme
    - Curated Physiology Widgets
    - Patient State Creation
    - Scenario Timeline
  - Version 2.0:
    - Multi Window UI
    - LUA Based Scripting Engine
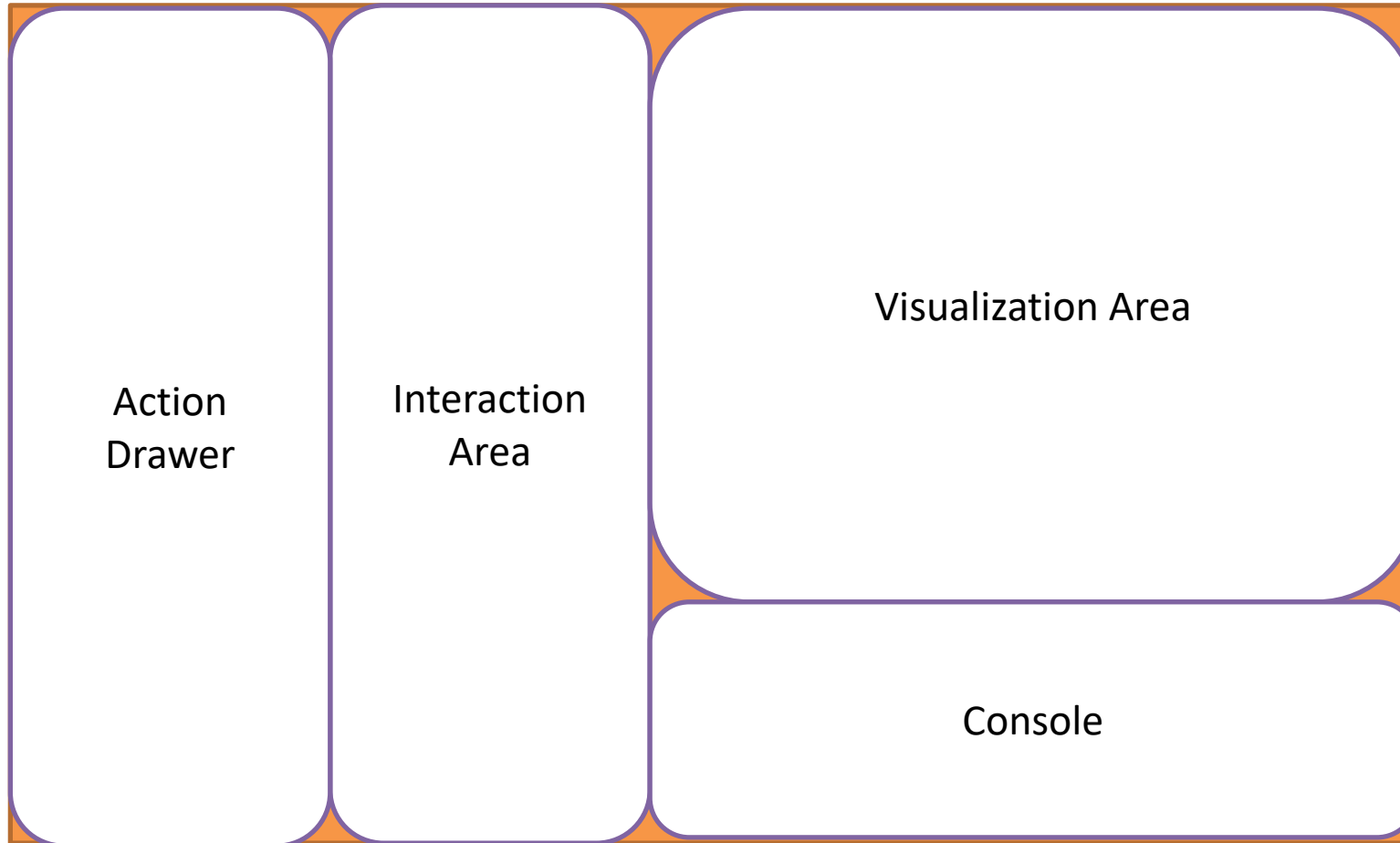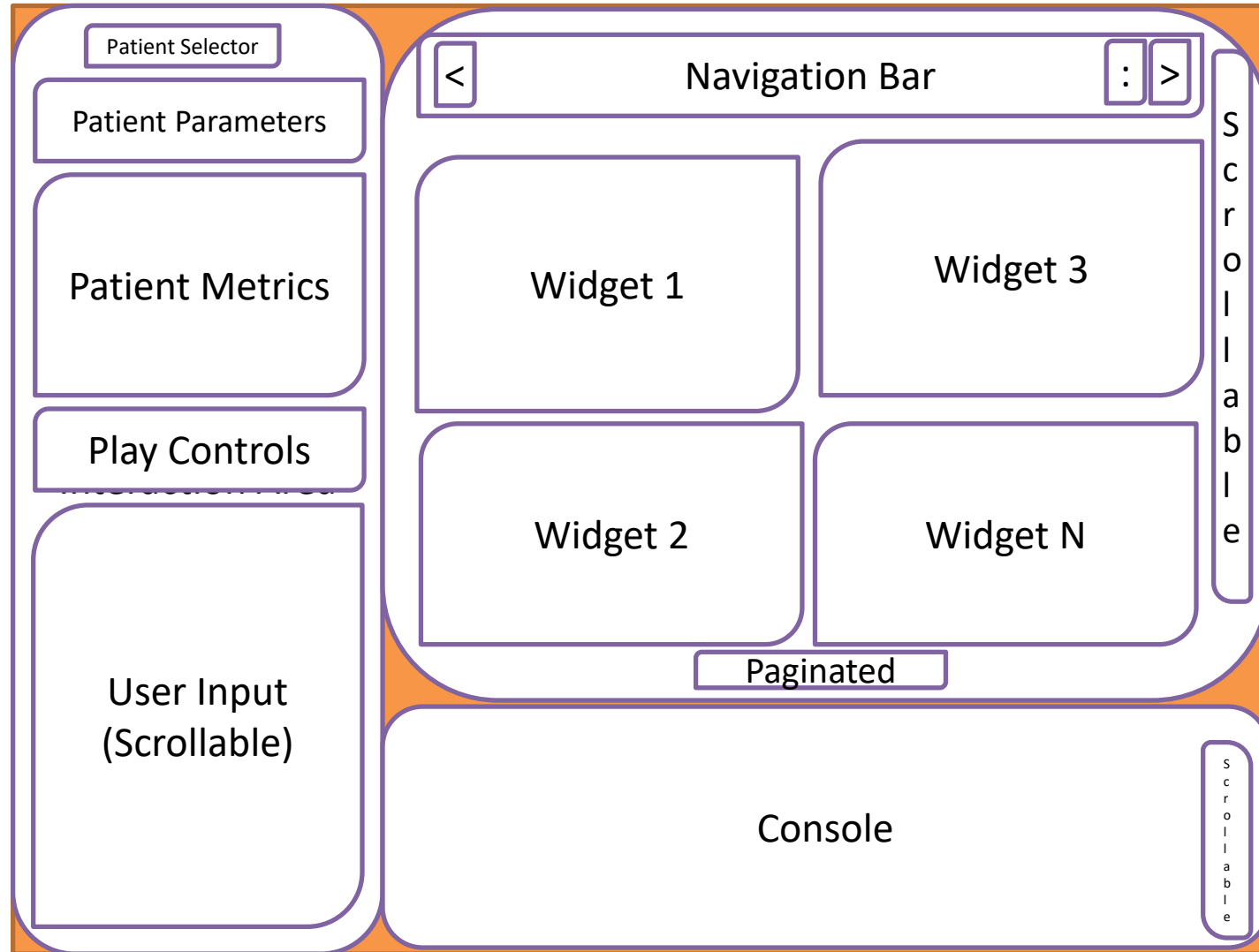   - Parallel Simulations
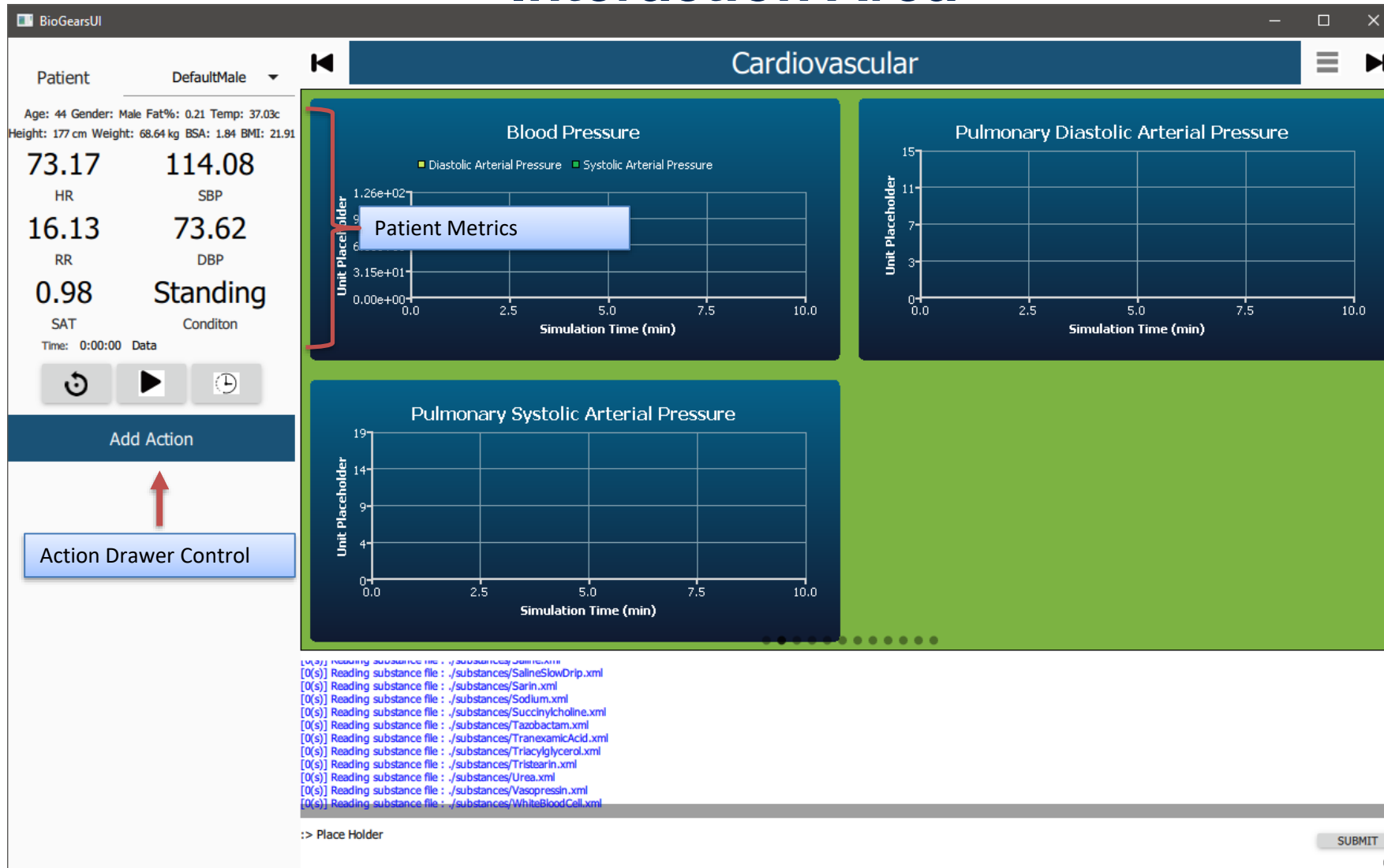
Alpha Release Walkthrough
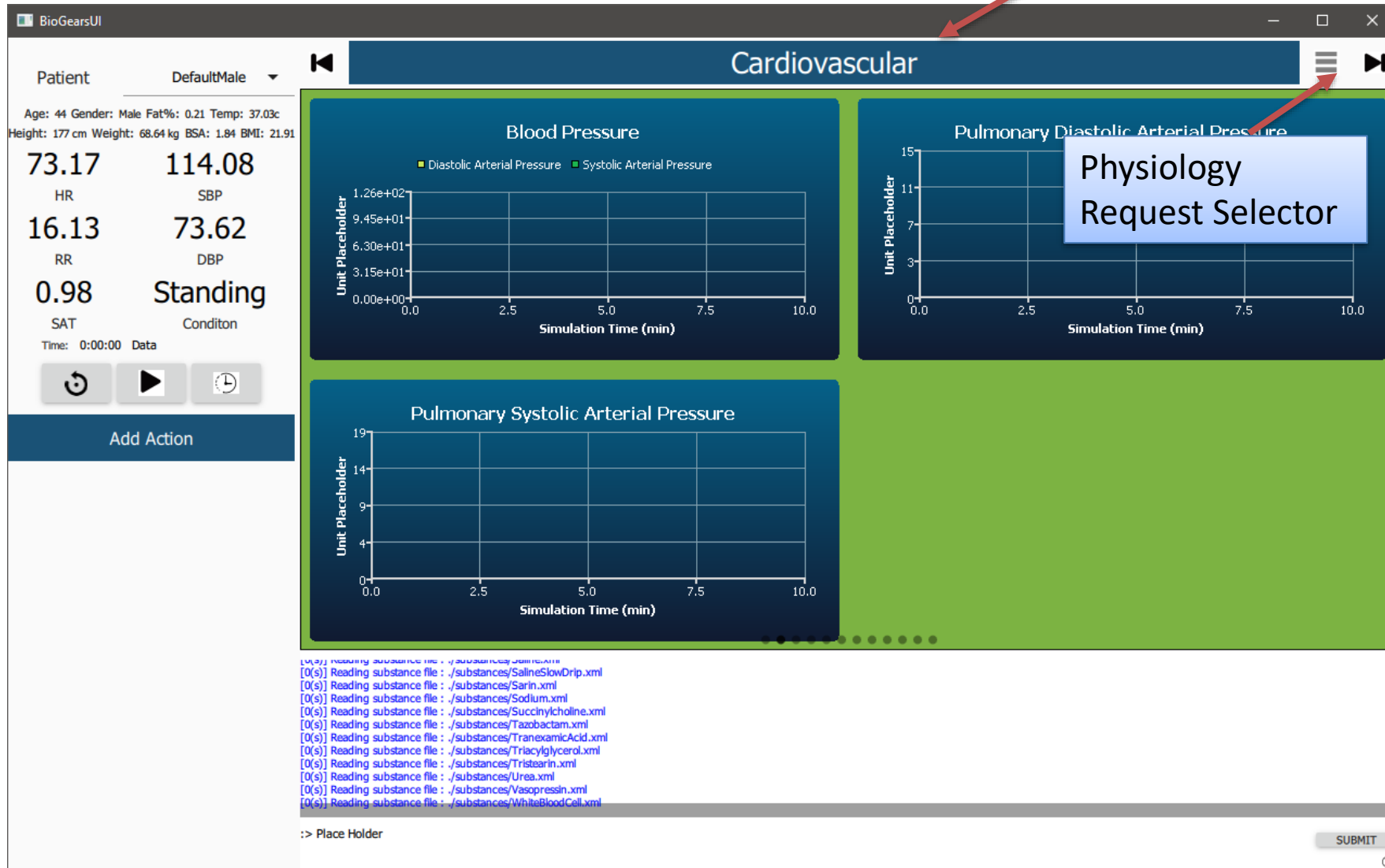
Current UI
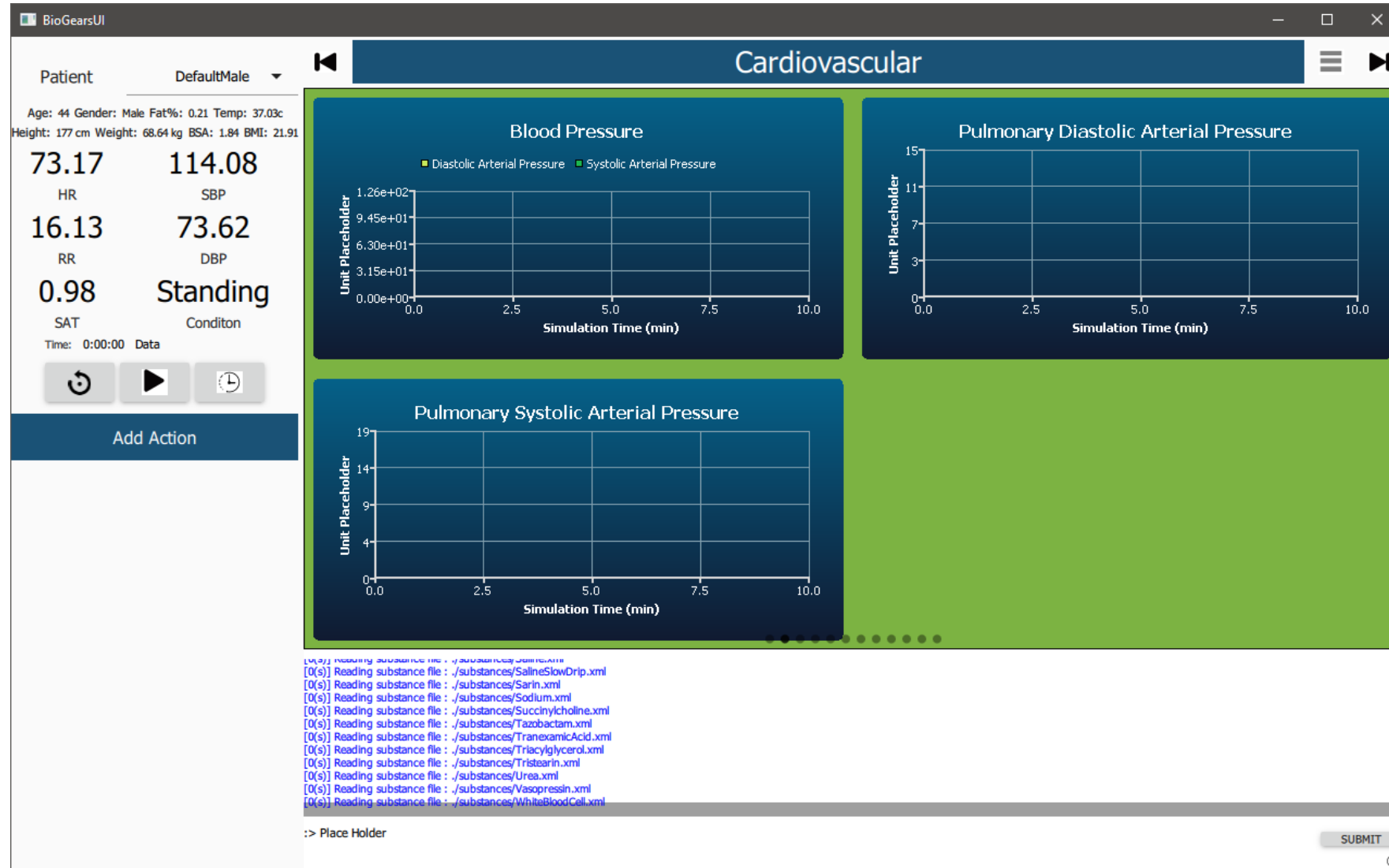
Current UI

UI Breakdown

Interaction Area

Visualization Area

Current Physiology Category

Physiology Request Selector

Physiology Request Selection

Current UI

Physiology Results

Color Coded BioGears Log

Action Selection

Action Creation

Current UI

Data Request Organization

Vitals

- Blood Pressure
- Respiration Rate
- Heart Rate
- O2 Saturation
- Cardiac Output
- Blood Volume
- Stroke Volume
- Central Venous Pressure
- Tidal Volume

Cardiopulmonary

- Cerebral Perfusion Pressure
- Intracranial Pressure
- Systemic Vascular Resistance
- Pulse Pressure
- IE Ratio
- Total Pulmonary Ventilation
- Lung Volume
- Alveolar Ventilation
- Dead Space Ventilation
- Transpulmonary Pressure

Data Request Organization

Data Request Blood Chemistry

- Blood Gas Concentrations
- Combined CO2 and O2 Partial Pressure
- Blood pH
- Hematocrit
- Strong Ion Difference

Energy & Metabolism

- Core Temperature
- Sweat Rate
- Skin Temperature
- Total Metabolic Rate
- Stomach Contents
  - Make this a single plot w/ sub-requests for different nutrients (calcium, sodium, …)
- Oxygen Consumption Rate
- CO2 Production Rate

Data Request Organization

Drugs & Substances

- Active drugs
- Substances
- Nutrients

Fluid Levels

- Extravascular fluid volume
- Extracellular fluid volume
- Intracellular fluid volume
- Total body fluid

Visualizer Initial Release

Release December 2020

Goals for initial binary release (v1.0)

- **Improved Interaction**
  - Ability to adjust variable actions in main UI
  - Distinguish between monetary actions and continual actions.
- **Improved Theme**
  - Uniform use of fonts across the UI
  - Common color scheme for metrics (Normal, Warning, Critical…)
  - Animated exercise state
  - Uniform scheme for widgets (Label font, legend position, line style…)
- **Curated Physiology Widgets**
  - Custom visualization to match common clinical interfaces (Patient Vitals, Metabolic Panels…)
- **Data Exports**
  - Patient States
  - Patient Files
  - Scenario Timelines
- **Scenario File Loading**

Interaction Improvements

Nest Patient Selection

Patient Selector

Patient Parameters

Patient Metrics

Play Controls

User Input
(Scrollable)

Patient   <Drop Down>

Bob (3)
Sally (3)
Larry
Other

```
Larry @ 0s
Larry With Asthma
Larry With
Hemorrhage
Larry @ 24 h Severe
Sepsis
```
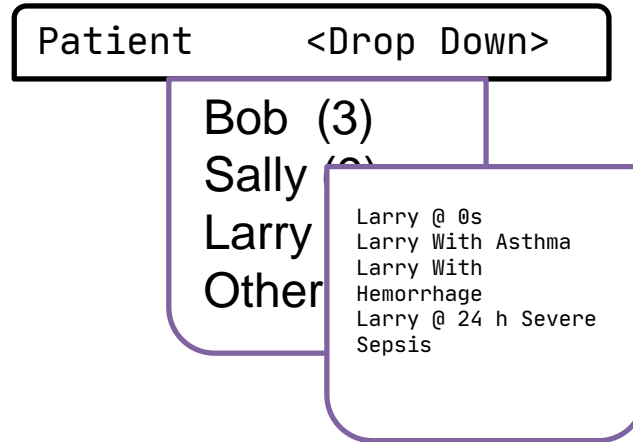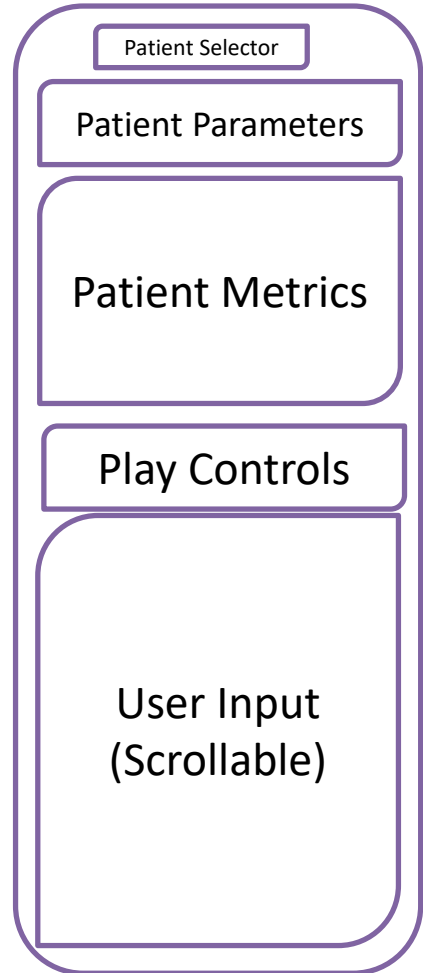
- Future function File -> Save
- Decide Name
- Decide Description
- Write File to States
- In Subfolder
  states/<patient_name>/

Purpose:
- Select initial conditions
- Switch state and restart

Initialization
- UI Parses all patients in patients/
- UI Parses all states in states/ and maps them to patient names

UX
- Select drop down browse to patient
- Cascading menus allow selection of specific states for a common patient
- Mouse over text gives full description string from state file

On Click
- Stop simulation
- Reset all visualization widgets
- Load state

Interaction Improvements

Customizable Action Values (Where Possible)

Patient Selector

Patient Parameters

Patient Metrics

Play Controls

User Input (Scrollable)

Severity Widget

[Right-Arm][0.4]

ON
OFF

- Configure
- Remove

Anesthesia Machine (Mask)

Source : Wall;Tank1;Tank2

T1 Volume

T2 Volume

ON
OFF

Auto Ticks Down As Action Occurs. Can Refill by Dragging.
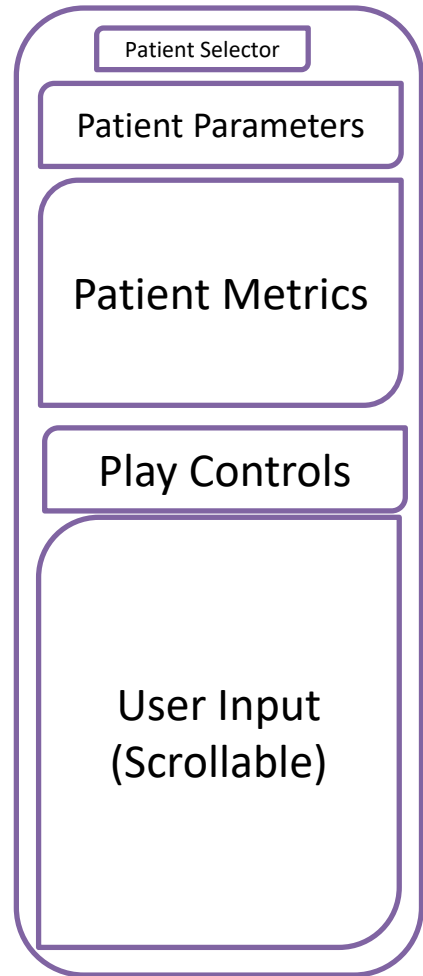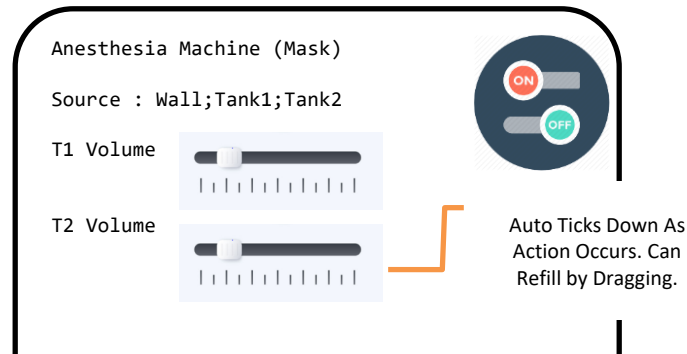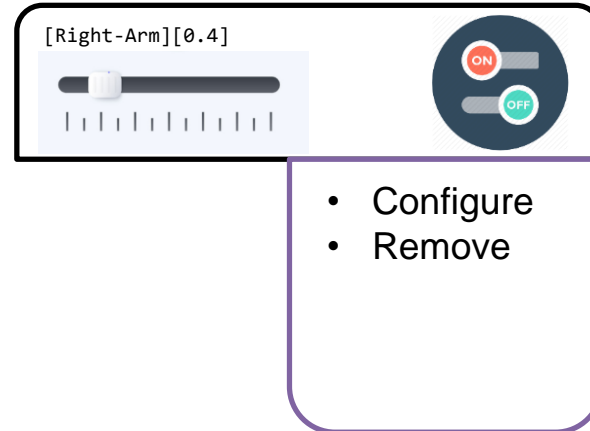
Purpose:
- Each Action has its own custom widget
- User Input is scrollable to allow us to see different actions
- Optimized font size for data density while ensuring readability

UX:
Right Click
- Configure: display original configuration widget and adjust bound parameters.
- Remove: deletes widget from action parameters

Toggle
- On: apply Action with current value
- Off: reverse Action if possible
  - Timed actions can not be toggled off but will show off once they are no longer active

Fields
- Manipulate : changes Action live if active else just stores new values
  - Interaction model is different based on UI configurable actions.

*Some Actions are unique (anesthesia machine, pneumothorax)

Interaction Improvements

Momentary Actions

Patient Selector

Patient Parameters

Patient Metrics

Play Controls

User Input
(Scrollable)

Consume Meal (Eggs & Bacon)

Protein

Water
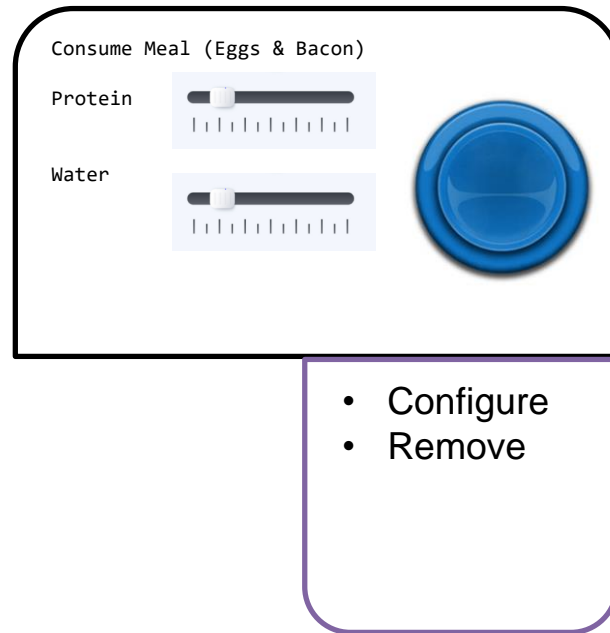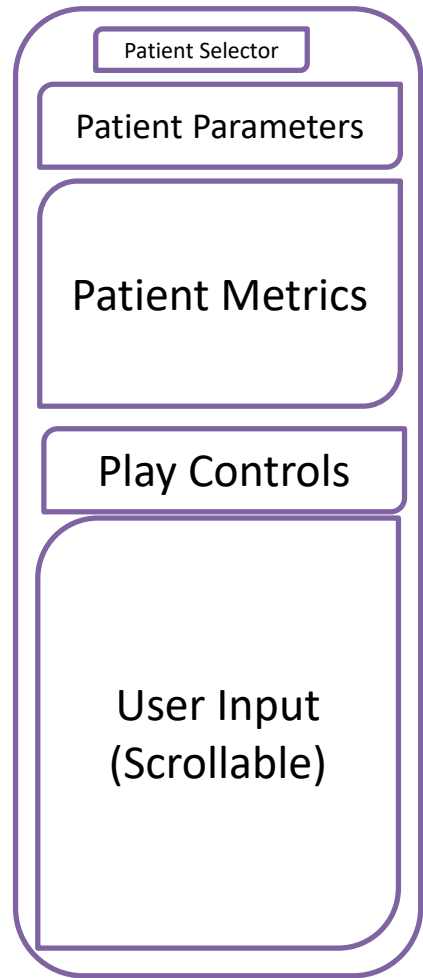
- Configure
- Remove

Purpose:
- Some actions are instantaneous and can not be toggled
  - Ex: Consume Meal

UX:
Right Click
- Configure: display original configuration widget and adjust bound parameters
- Remove: delete widget from action parameters

On Click
- Apply Action with current value

Fields
- Manipulate : changes Action live if active else just stores new vales
  - Interaction model is different based on UI configurable actions.

Visualization Improvements
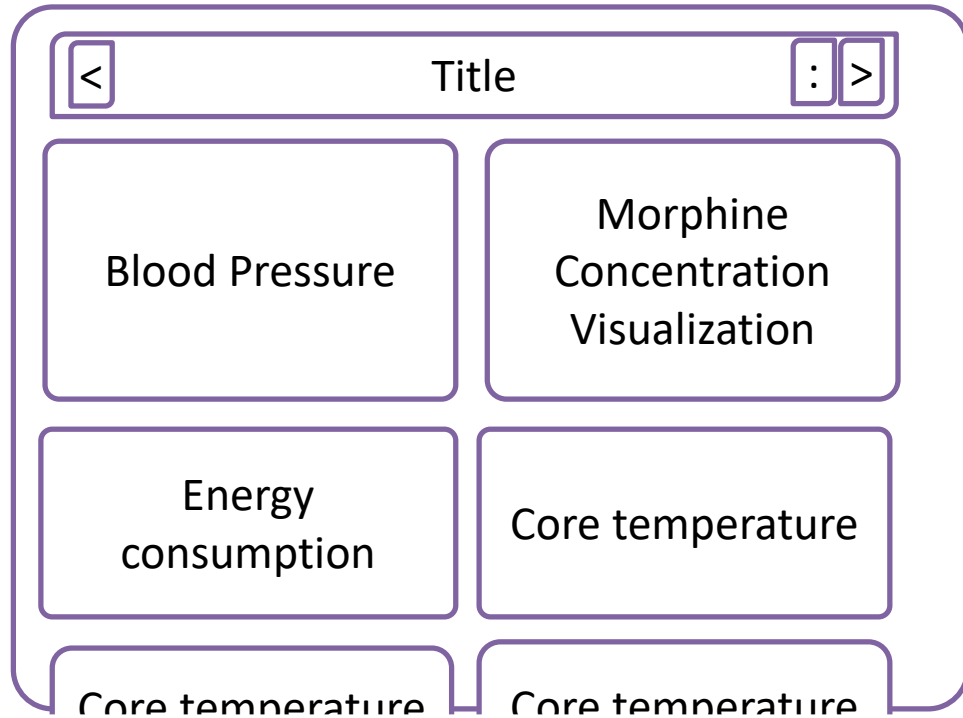
Purpose: Paged Visualization Area of BioGears Metrics

Categories
- Categories Based on Data Request Organization

Controls
- **<** Previous Category (Loops)
- **>** Next Category (Loops)
- **:** Opens Selection Menu of Widgets Available on Current Page

Features:
- Dynamic Graphs
- Custom Visualizations
- Scrollable Area to Allow

Custom Visualization

Moving forward we would like to produce several non plot-based visualizations.

Some ideas include:

- Patient Vitals
- Substance concentration visualizer (in the body)
- Pulmonary Activity
- Breathing Cycle
- Bladder volume

Data Exports

File | Export | Help

Save
Save a
Open
Exit

Patient
State
Environment
Scenario
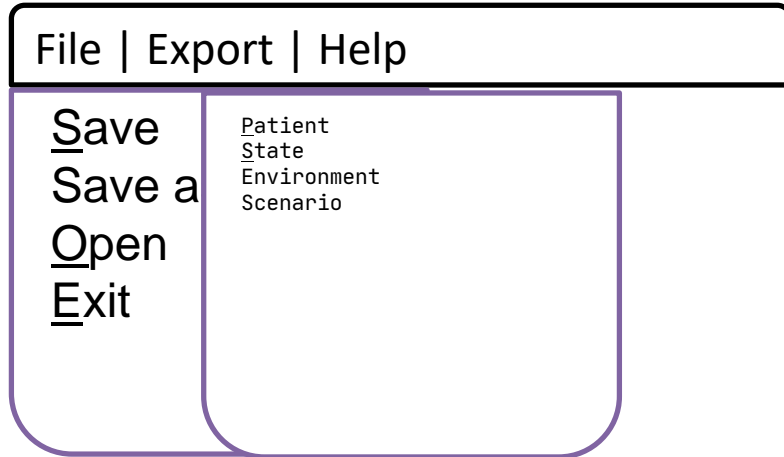
Purpose:
- Allows users to save work for later use
- Convenient way to create save points for long simulations and repeated demonstrations

UX

File
- Save – <patient_name>@<scenario_time>.xml
- Save As – User Chosen Filename
- Open – Open a Scenario File
- Exit – Exit Program

Export
- Patient – Export Patient to a File
- State   - Same as Save
- Environment – Export env to a File
- Scenario – Export Current Scenario to a File

Scenario Open

File | Export | Help

<u>S</u>ave
Save as
<u>O</u>pen

```
1. AsthmaAttack.xml
2. SepsisSevere.xml
3. MaceToTheFace.xml
4. Marathon.xml
```
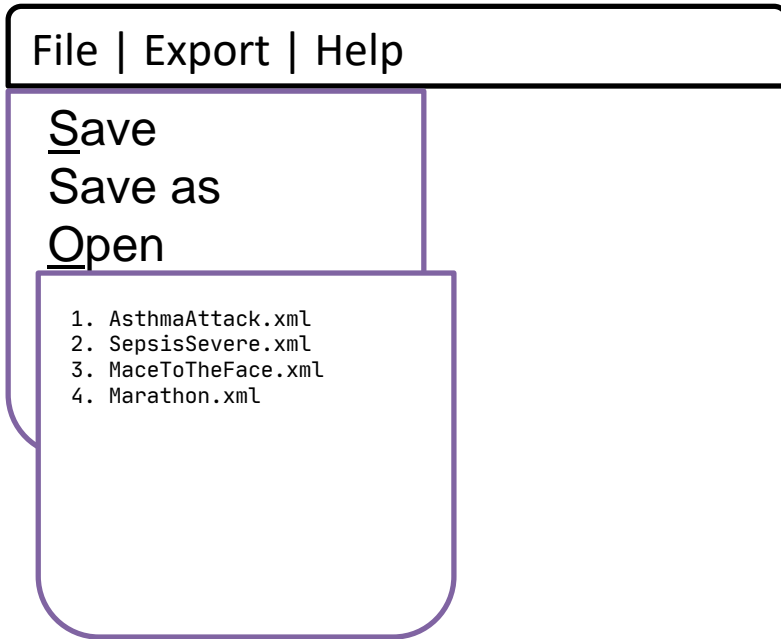
Purpose:
- Allows user to quickly open a scenario file
- State opening handled in interaction area

UX:
- Jump list of previous 10 scenarios

Hot Keys
- Ctrl 0-9 (Load matching entry)
- Control + O (File Dialog)

On Click
- (Entry) Load shown scenario
- (Load) File Dialog for finding new scenario file

ARA

Scenario Timeline

- In the current UI actions cannot be scheduled
- Instantaneous applications of actions is great for experimentation and demonstration, but bad for repeatable trials
- Scenario timeline would allow scheduling of events at a given simulation time
- Timelines could be imported and exported by CDM::ScenarioData format

Future Functionality

- Multi Window UI
  - Ability to detach widgets from main window to a sibling window
- LUA Based Scripting Engine
  - Command line interface to apply actions
  - Compound actions through LUA scriptable buttons
- Parallel Simulations
  - Scenario 2.0 batch runs where branch points launch multiple simulations
  - Ability to perform state generation
- cmd_bio Parity
  - Patient State Generation
  - Substance Data Generation
  - .config file support for batch runs.

Multi UI Window

- Pop out windows for complex widgets
- Allows users to see multiple categories simultaneously
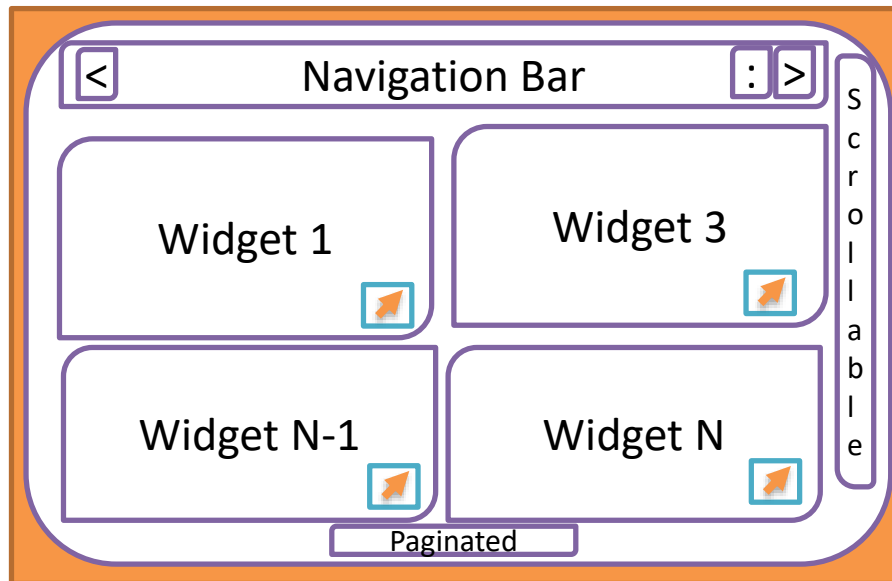
LUA Scripting

Breakdown

```
┌─────────────────────────────────────┐
│                                  S   │
│                                  c   │
│                                  r   │
│              Console             o   │
│                                  l   │
│                                  l   │
│                                  a   │
│  ┌────────────────────────────┐  b   │
│  │           Prompt           │  l   │
│  └────────────────────────────┘  e   │
└─────────────────────────────────────┘
```
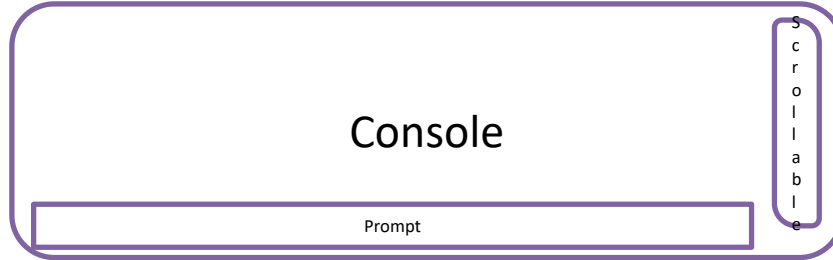
Example

```
[23:59] Applied Hemorrhage
[23:59] Applied Tourniquet
[00:10] Applied TXA Infusion




┌──────────────────────────────────────┐
│ :> apply_hemorrhage LeftLeg .8         │
└──────────────────────────────────────┘
```
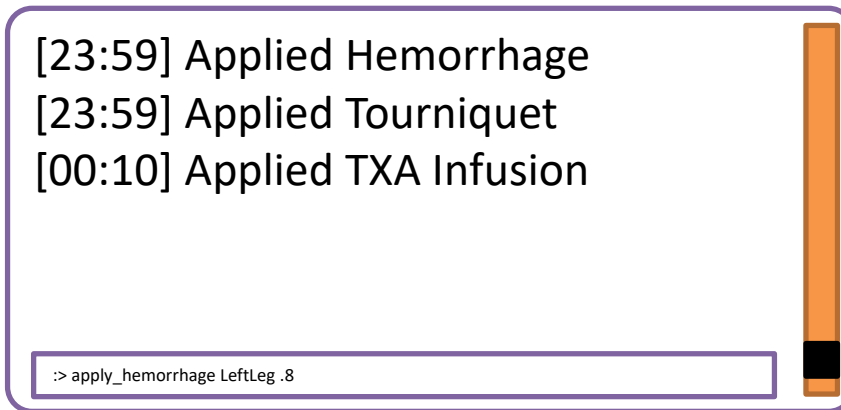
Purpose: All CMD-like control of biogears actions and insults.

LUA engine will allow API access to current engine

Features
- Tab Complete
- Prompt History = 1000 inputs
- Prompt History Storage File = biogears_history
- Interactive help function

- *Example*
- *hemorrhage LeftLeg .8*
- *pain_stimulus LeftLeg .5*
- *infuse_compound txa .5 1*
- Help hemorhage

LUA Scripting (Stretch Goal)

- Custom Widgets using Data Request API

  - Advance features to assist in model validation

  - Generic plot support

  - Configurable Y-scale factors

  - Text field entry of compartment and physiology requests

- Custom Action using Exposed API

  - Advance feature to assist in action development

  - Editable text fields for lua scripting

  - Syntax verification before saving

  - Log messages reported in console area

CMD_BIO Parody

BioGears intends to eventually reach feature parity with our command line utilities.

- **Future Features:**
  - Generate Patient States
    - Run stabilization for all patients in patient_dir and create a patient@0s.xml in states_dir
  - Generate Substance Data
    - Read Substance.csv or excel master file and generate CDM files in substances/
  - Run Parallel Simulations using Multi-Threading from a *.config file
    - Configure multiple batch runs with varying parameters
    - Support multiple patients and multiple timelines
    - Automatically run a configured number in parallel and save results to csv file
  - Export Graph Plots to Image Format.
    - After a run give csv results file
    - Generate PNG images for presentations for proof of validation
    - Useful for presenters and for historical review
    - Plots will feature baseline support to show variation from expected or clinical data points
    - Complex feature which may take several iterations to implement fully