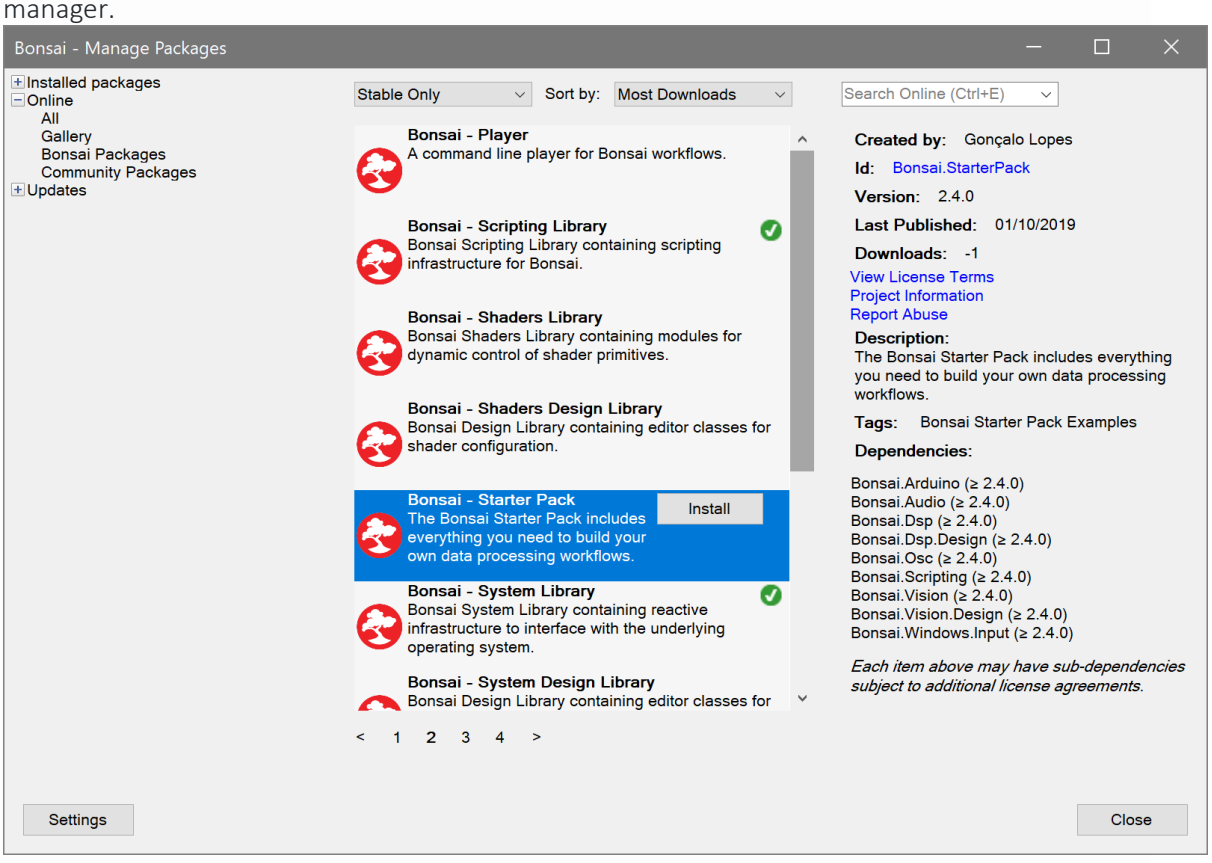
Day 4 – Exercises

Bonsai

Adapted from Neurogears @ Wustl19, https://neurogears.org/wustl-2019/worksheets/acquisition/

Getting Started

1. Download Bonsai from http://bonsai-rx.org.
2. Install **Bonsai - Starter Pack** from the package manager.



3. Click the `Updates` tab on the left side of the screen and install any available upgrades.
4. Refer to http://bonsai-rx.org/docs/editor for an introduction to the user interface.

Video Acquisition

Bonsai can be used to acquire and record data from many different devices. The exercises below will make you comfortable with the most common Bonsai data types. The first data type we will discuss is an image, which is represented as a 2D matrix of pixels. Each pixel represents either a brightness value in a grayscale image, or a BGR colour value in a colour image.

Exercise 1: Saving a video

- Insert a `CameraCapture` source.
- Insert a `VideoWriter` sink.
- Configure the `FileName` property of the `VideoWriter` operator with a file name ending in `.avi`.
- Run the workflow and check that it generates a valid video file.

Audio Acquisition

Audio data is captured at much higher temporal sampling frequencies than video. However, the data is typically buffered into chunks of multiple samples before being sent to the computer. Also, multiple audio channels can be acquired simultaneously in the case of a stereo microphone, or high-density ephys probes. For this reason, such multi-sample, multi-channel data is also typically represented as a 2D matrix of amplitude values, where rows represent channels, and columns represent time.

Exercise 2: Saving a WAV file

- Insert an `AudioCapture` source.
- Insert an `AudioWriter` sink.
- Configure the `FileName` property of the `AudioWriter` operator with a file name ending in `.wav`.
- Make sure that the `SamplingFrequency` property of the `AudioWriter` matches the frequency of audio capture.

- Run the workflow for some seconds. Playback the file in Windows Media Player to check that it is a valid audio file.

Exercise 3: Trigger an auditory stimulus

- Insert an `AudioReader` source.
- Configure the `FileName` property to point to the audio file you recorded in *Exercise 3*.
- Insert an `AudioPlayback` sink.
- Run the workflow and check that the sound is played correctly.

- Insert a `KeyDown` source (Windows.input).
- Set the `BufferLength` property of the `AudioReader` to zero, so that all audio data is read into a single buffer.
- Combine the key press with the audio data using the `WithLatestFrom` combinator.
- Right-click the `WithLatestFrom` operator. Select `Output` > `Item2` from the context menu.
- Move the `AudioPlayback` sink so that it follows the selected `Item2` member.
- Run the workflow and press a key. What happens if you press several keys?

Exercise 4: Control an LED

- Upload StandardFirmata to your teensy in the Arduino software (File/ Examples/Firmata/StandardFirmata)
-
- In Bonsai: Insert a `Boolean` source.
- Insert a `DigitalOutput` sink.
- Set the `Pin` property of the `DigitalOutput` operator to 13. This is the LED pin of the teensy- it turns on the LED that is already attached to the board.
- Configure the `PortName` property.
- Run the workflow and change the `Value` property of the `Boolean` operator.
- **Optional:** Change the output pin to a different digital output pin. Connect an LED and a resistor to this pin through the breadboard and turn the LED on with Bonsai. The LEDs in your kit have three pins, you can treat them as a normal LED by leaving one of the short pins unconnected.

Video Tracking

Bonsai allows processing the captured raw data to extract real-time measures of behaviour or other derived quantities. The exercises below will introduce you to some of its online video processing capabilities.

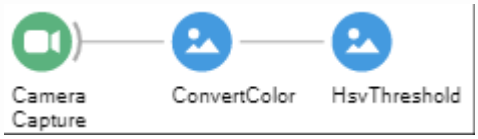
Exercise 5: Segmentation of a coloured object

Find a coloured object to track.

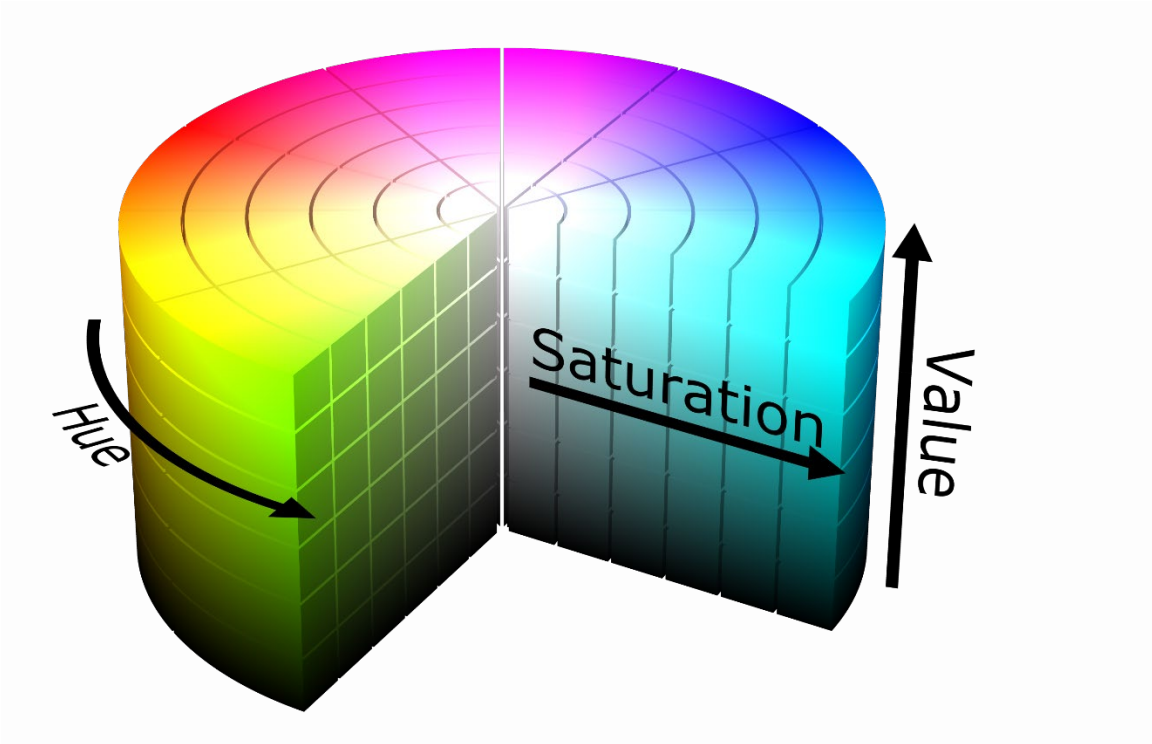
- Insert a `CameraCapture` source.
- Insert a `RangeThreshold` transform.
- Open the visualizer for the `RangeThreshold` operator.

- Configure the `Lower` and `Upper` properties of the `RangeThreshold` to isolate your coloured object (hint: click the small arrow to the left of each property to expand their individual values).

This method segments coloured objects by setting boundaries directly on the BGR colour space. This colour space is considered a poor choice for colour segmentation. Can you see why?



- Replace the `RangeThreshold` operator by a `ConvertColor` transform. This node converts the image from the BGR colour space to the Hue-Saturation-Value (HSV) colour space.



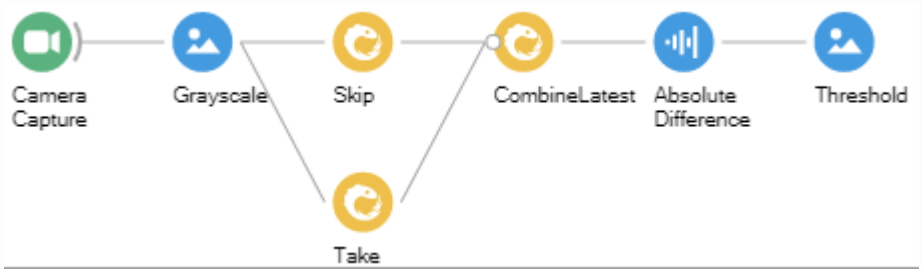
- Insert an `HsvThreshold` transform.
- Configure the `Lower` and `Upper` properties of the `HsvThreshold` to isolate the object.

Exercise 6: Real-time position tracking

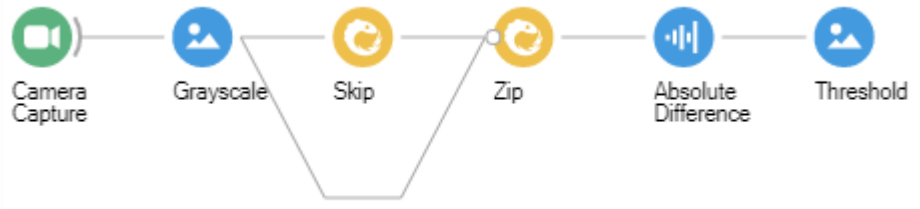
- Starting with the workflow from the previous exercise, insert a `FindContours` transform. This operator traces the contours of all the objects in a black-and-white image. An *object* is defined as a region of connected white pixels.

- Insert a `BinaryRegionAnalysis` transform. This node calculates the area, center of mass, and orientation for all the detected contours.
- Insert a `LargestBinaryRegion` transform to extract the largest detected object in the image.
- Select the `ConnectedComponent` > `Centroid` field of the largest binary region using the context menu.
- Record the position of the centroid using a `CsvWriter` sink.

Bonus Exercise 7: Background subtraction and motion segmentation



- Create a grayscale video stream similar to *Exercise 2*.
- Insert a `Skip` operator. Set its `Count` property to 1.
- In a new branch, insert a `Take` operator. Set its `Count` property to 1.
- Combine the images from both branches using the `CombineLatest` combinator.
- Insert the `AbsoluteDifference` transform after `CombineLatest`.
- Insert a `Threshold` transform. Visualize the node output and adjust the `ThresholdValue` property.



- Replace the `CombineLatest` operator with the `Zip` combinator.
- Delete the `Take` operator.

Describe in your own words what the above modified workflow is doing.