Isadora (for users)

General information

Isadora is a graphical programming environment we will be using to control digital media.

Isadora has a pretty intuitive feel, however for first time users it may be unclear which actors to use for specific purposes. Isadora has created a few tutorial videos that are extremely useful for first time users.

If you are looking for more information or are seeking advice there is a lot of information on the Isadora forums.

Custom Actors

Isadora uses modules (called actors) to control the flow of data within its interface. We have created two new actors

Face Tracker
To download the plugin only click here.
To download the full file including the source code click here.

This actor allows for facial tracking with a use of a webcam.
In order to use this actor we first need to enable our webcam within Isadora.
Steps:
1. Input > Live Capture Settings
2. Scan for Devices
3. Video Input (works well with the following settings in Roberts lab)

- Device: Choose a webcam
- Resolution: 320x240 (NTSC Half)
- Quality: Normal Quality

4. Start Live Capture
5. Output > Force Stage Preview

Inputs:

- VideoIn - A video source (like video in watcher)
- Bypass - An on/off switch that decides whether to apply the face tracking to the video feed
- Rectangle - An on/off switch that decides whether to apply a red rectangle around the face

Outputs:

- VideoOut - The processed video feed. Can be sent on for more manipulation or for display (projector)
- X Pos - The x-position of the face on the screen (may need to control the scaling when using it with other actors)
- Y Pos - The y-position of the face on the screen (may need to control the scaling when using it with other actors)

Python Interpreter
This actor allows Isadora to call a python function located anywhere on the system. There are a few restrictions that have been implemented to allow this framework to be possible.

File restrictions and rules:

- Must be a python file (".py" extension)

Function restrictions and rules:

- Functions can have at most 10 parameters
- Default values are not supported
- Parameters have to be of the type "int", "float", or "string"
- Parameter names must end with "_int", "_flt", or "_str" corresponding with their type. For example:

```
def add( x_int, y_int):
    """
    This function adds 2 values
    @x_int: an integer
    @y_int: an integer
    @return: an integer
    """
    return x_int + y_int
```

- Multiple input types is allowed
- Output types are also restricted to "int" or "float" but they can be a tuple (i.e. (0,1,2,3))

```
def count(iters_int):
    """
    This function counts from 1 to iters_int
    @iters_int: the number of values to count to
    @return: a tuple of integers ranging from 1 to iters_int inclusive
    """
    L = []
    for i in range(1,iters_int + 1):
        L.append(i)
    return tuple(L)
```

- Maximum of 20 outputs
- No string outputs

Inputs:

- Path - The directory containing your python file
- File - The python file without the extension (i.e. example not example.py)
- Func - The name of the function
- GetParams - An on/off switch that creates or destroys the python parameters within Isadora
- The parameter names of your python function will appear in the Isadora actor.

Outputs:

- Output - The number of outputs will change depending on how many outputs the python function returns.

Future work

Python Interpreter

- string outputs
- handle python inputs with default values
- handle video input to the python program