Samsara

# Instructions to run

Description:

Samsara is a virtual interactive soundscape for multiple real-world performers, each performing a different role within a virtual physics based environment. Breaking the traditional paradigm of a conductor (in control) and instrumentalists: we propose a new paradigm for music creation in the digital age.

The music produced is a collaborative consequence of the actions of multiple performers in the system each performing one of three roles (of control) - a creator (controlling sound generator entities), a preserver (controlling filter entities) and a destroyer (controlling transient destruction of the generators). Each type of entity is an atom that bounces around in a physics-based environment. The environment can be projected onto a big screen and each performer interacts with the system through their mobile phones (via an interface we designed using Touch OSC). The percussive sounds are controlled by the amount of entities in the system at a given time. The dynamically changing soundscape is the result of the generator atoms being filtered appropriately (weighted) by their distance from each of the filter atoms (Moog/LFO/Reverb). (Check out snapshots and video for a better idea)

Samsara is inspired from Hindu mythology where Samsara is a Sanskrit word meaning the "cyclicality of life." and the holy trinity of Gods - Brahma the creator, Vishnu the preserver and Shiva the destroyer. We imagine a multiple performer musical paradigm inspired by the same.

Github link – <https://github.com/eardrummer/Samsara_HackerEarth>

Instructions:  
This project has been built using 3 main elements –

1. OpenFrameworks – For controlling visuals and the environment

<https://openframeworks.cc/download/>

1. Pure Data – For controlling and generating the sounds

<https://puredata.info/>

<https://agraef.github.io/purr-data/> (linux alternative)

1. TouchOSC – For sending OSC msgs from controller on android device to Openframeworks

<https://hexler.net/software/touchosc>

Step 1) Install Openframeworks and create a new project using the Project generator.

(<https://openframeworks.cc/learning/01_basics/create_a_new_project/> )

Built in add on’s to be added from Project Generator (<https://openframeworks.cc/learning/01_basics/how_to_add_addon_to_project/> )

1. ofxBlur
2. ofxGif
3. ofxGui
4. ofxNetwork
5. ofxOsc
6. ofxCV

This will create the folders /bin /opj/ src/ and files ./addons.make ./config.make ./Makefile in your project directory.

Step 2) Download from the github repo of the project and

Replace the following folders in your project directory with the folders of the same name from the repo

* + 1. ./src
    2. ./bin/data

Step 3) In order to run on your wifi connection. Update your IP address as

In ./src/ofApp.h

@define HOST “YOUR IP ADDRESS HERE”

Step 4) From terminal go to the project directory and run

>make

>make RunRelease

**For the Controller:**

Open TouchOSC android application (from hexler: <https://hexler.net/software/touchosc> )

Download the samsara layout from the github repo and open the layout

Your device must be connected to the same WIFI network as your laptop running openframeworks

Step 1) (On TouchOSC application Go to OSC>HOST> “Enter the IP Address of your laptop here”)

Step 2) OSC>Port(outgoing)>

Make this 9001 for performing the role of Creator

9002 for performing the role of Preserver

9003 for performing the role of Destroyer

**For the Sound:**

Open PureData or Pd2Lork (purr data) on Linux

Open the PDPatches/mainPDPatch.pd from the github repo

**Now you have all the steps in place to run the system**

1. Open PureData patch.
   1. Click on pd dsp 1 and the big orange checkbox to turn on Sound on your audiocard
2. Run the openframeworks project (make RunRelease)
3. Open TouchOSC application connected to the IP address with a port number (9001/9002/9003)
4. Start interacting in the system from your mobile device controller

AND MUSIC.