**Project 1: Initial Prototype**

User flow:

1. User presses app icon
2. Launch screen comes up
3. Then View comes up
4. User clicks “Meditate” button:
   1. ((Options… A pop-up alert opens when user clicks “Mediate” button that says, “Breathe in for 10 seconds, breathe out 20.” OR… Just start meditation))
5. Timer starts
6. Breathe in for 10 seconds
7. Breathe out for 20
8. User repeats till done meditating
9. When done meditating, click “Done” (or whatever the button is) and it stops
   1. ((Option… could have an alert pop-up open and say “enjoy your day?” or something after done meditating?))

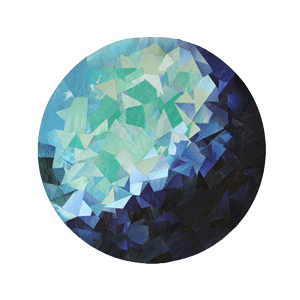
Assets in Timer1 Xcode prototype

Icon: 58, 87, 80, 120, 180

Launch: 200x200

Main image: 300x300

Animation:



**NSTimer notes**:

Option 1: (Timer1 prototype)

Stop and restart timer based on if/else statements and when timerLabel.text = “10”

If startButtonIsOn=true, timer runs

If startButtonIsOn = true & timerLabel.text == “10”, call restartTimer function

// if statement in updateTimer function

restartTimer function:

timer.invalidate stops timer, seconds = 0, timerLabel.text = “\(seconds)”, and call startTimer function

startTimer function()

Option 2: Figure out loop?

“*NSRunLoop* continuously executes a loop in which it checks for input events. One of the events that is checked for is the ‘timer interval elapsed event’. When the run loop detects this kind of event it will fire the method on the target associated with the timer.”

>>> So I need to run the loop and have the input be “10” (breathe in for 10 seconds) for the “timer interval elapsed time event.”

**Connecting timer to animation**:

With current prototype:

If timerLabel = 11, restartTimer, and doSomeAction that makes changeImage = true

If timerLabel = 11, resterTimer, and changeImage = true, doSomeAction2 that makes changeImage = false

doSomeAction (in my prototype, it’s changing a label text color to blue)

Option 1: timer keeps counting up…

If timerLabel = 11 or increment of 11, call animate function breatheOut

if timerLabel = 21 or increment of 21, call animate function breatheIn

Option 2: keep timer repeating as it is now…

if timerLabel = 11, call animate function breatheOut

timer seconds = 0 and start timer again

if timerLabel = 21, call animate function breatheIn

How do you call a function every other time?

Connecting NSTimer to object:

<http://stackoverflow.com/questions/24889279/passing-parameters-to-a-method-called-by-nstimer-in-swift>

**Pulse animation**:

* https://github.com/vinayjn/CircularPulse
* https://www.youtube.com/watch?v=ektBr-UZRvc
* https://github.com/shu223/PulsingHalo
* https://github.com/ctews/SwiftPulse
* https://github.com/shu223/Pulsator

Animation timing:

http://ronnqvi.st/controlling-animation-timing/