MAD Project 1 Milestone 2

Invision mockup: <https://invis.io/JTDSBTOG3>

Pseudocode

var stepLabel

var time //track how many seconds are on the timer

create timer

bool timerRunning //keep track if timer is running

bool pauseTapped //keep track if timer is paused/resumed

array steps //keep track of step names and default times

int count //keep track of what step user is on

var originalTime //keep track of original value of time

create backgroundCircle CAShapeLayer

create movingCircle CAShapeLayer

create circular path

func setTime{ //called from time picker

get new time from picker

set time

}

func nextStep{ //called from next button

call reset timer

increase count

update stepLabel

update next button title

update previous time label

update backgroundImage

}

func previousStep{ //called from previous button

call reset timer

decrease count

update stepLabel

update next button title

update previous time label

update backgroundImage

}

func startOver{ // called from Start Over button

call reset timer

count=0

update stepLabel

update next button title

update previous time label

update backgroundImage

}

func startTimer{ //called from start button

IF timer is not running{

call function to run timer

disable start button

}

}

func runTimer{ //called from startTimer, resume timer

convert label time to int seconds

run timer from time on label

set timerRunning to true

enable reset and pause button

}

func pauseTimer{ //called from Pause/Resume button

IF pauseTapped is false { //first time button is tapped so user wants to pause timer

pause timer

set timerRunning to false

set pauseTapped to true

set title of pause button to resume

} else{ //second time button is tapped so user wants to resume timer

run timer

set pauseTapped to false

set timerRunning to true

set title of resume button to pause

}

}

func resetTimer{ //called from Reset button

stop timer

reset time label (get default time from array)

set timerRunning to false

set pauseTapped to false

set title for pause/resume button to pause

enable start button

disable reset button

reset end point of movingCircle to initial position

}

func updateTimer{ //decrements the time and checks to see if timer is below 0

IF time is less than 1{ //time is up

stop timer

} else {

time-=1

update label to display correct time

remove previous animations

set movingCircle end point proportionally to the time left

}

}

func timeString 🡪 String { //convert seconds to display friendly format

minutes = time / 60 % 60

seconds = time % 60

return String(minutes:seconds)

}

viewDidLoad{ //initialization

disable pause button

set look of background circle including end point

set look of moving circular timer (stroke, fill, line width, etc.)

add circle drawings to pathView

}