Artillery Deflection Steps

Ken Harmon

2019 November 26

Base Map

```
library(ggmap)
register_google(key = "AIzaSyAfnLNZjvYdMx-cyga_qA1oJ6P36dRGalA")
octc <- get_map(location = "11TNH7000080000", maptype="terrain" ,scale=4, zoom =10)
map<-octc
```

Shiny

```
ui <- fluidPage(
  h1("Compute Deflection"),
  h2("by Ken Harmon"),
  sidebarLayout(
    sidebarPanel(
       radioButtons(inputId = "maptype",label = "Pick a Map Type",choices =
                      c("roadmap"="OpenStreetMap.Mapnik",
                        "terrain"="Stamen.Terrain",
                        "satellite"="Esri.WorldImagery")),
       textInput("MGRS","Center MGRS", "11TNH7000080000"),
       numericInput("aof","Azimuth of Fire",2000,min = 0,max = 6399),
       actionButton(inputId = "get_map", label = "Get Map"),
       textOutput("Target"),
       textOutput("dist"),
       textOutput("az"),
       textOutput("defl")
    ),
    mainPanel(

     leafletOutput("mymap", width = "600px", height = "600px")
              )
  )
)
server <- function(input, output) {

  mgrs <- reactive ({
    mgrs <- mgrs_to_latlng(input$MGRS)
    clat <- mgrs[[2]]
    clng <- mgrs[[3]]
  })
```

```
  output$mymap <- renderLeaflet({
    if (input$get_map == 0)
      return()
    isolate({
    mgrs <- mgrs_to_latlng(input$MGRS)
    clat <- mgrs[[2]]
    clng <- mgrs[[3]]
    aof <- input$aof
    aoftip <- destPoint(c(clng,clat),input$aof/6400*360,10000)
    aofdf <- data.frame(lng = c(clng,aoftip[[1]]),lat = c(clat,aoftip[[2]]))


    m <- leaflet() %>%
            setView(clng,clat,zoom=10) %>%
            addProviderTiles(input$maptype) %>%
            addCircleMarkers(lng = clng, lat = clat, radius = 5) %>%
            addPolylines(data = aofdf, ~lng, ~lat, group = "aof",
                         color = "orange")
    })
  })
  observeEvent(input$mymap_click, {
    click <- input$mymap_click
    tlng <- click$lng
    tlat <- click$lat
    mgrs <- mgrs_to_latlng(input$MGRS)
    clat <- mgrs[[2]]
    clng <- mgrs[[3]]
    text<-paste("Latitude ", round(clat,2), "Longtitude ",
                round(clng,2))
    output$Target <-  renderText({paste("Target: ",
                                        latlng_to_mgrs(click$lat,click$lng))})

    dist <- trunc(distGeo(c(clng,clat),c(tlng,tlat)))
    output$dist <- renderText({paste("Distance to Target: ", dist, "m")})

    aof <- input$aof

    az <- trunc(bearing(c(clng,clat),c(tlng,tlat))*6400/360)
      if (az<0) {az <- az + 6400}
    output$az <- renderText({paste("Azimuth to Target: ", az, "mils")})

    defl <- 3200+(aof-az)
      if (defl < 0) {defl <- defl + 6400}
    output$defl <- renderText({paste("Deflection to Target: ", defl, "mils")})

    proxy <- leafletProxy("mymap")

    pointdf <- data.frame(lng = c(clng,tlng),lat = c(clat,tlat))
      ## This displays the pin drop circle

    proxy %>%
      clearGroup("new_point") %>%
          #clearMarkers(layerId=input$mymap_click$id) %>%
```

```
            #addPopups(click$lng, click$lat) %>%
        addCircles(click$lng, click$lat, radius=100, color="red", group =
                       "new_point") %>%
        addPolylines(data = pointdf, ~lng, ~lat, group = "new_point")
    })
}

shinyApp(ui, server)
```

Shiny applications not supported in static R Markdown documents