

Bingo Master

```
library(shiny)
library(dplyr)

ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      actionButton("draw", "Draw"),
      br(),
      br(),
      actionButton("new_game", "New Game"),
    ),
    mainPanel(
      uiOutput("display")
    )
  )
)

server <- function(input, output) {
  v <- reactiveValues(
    current = NULL,
    previous = NULL,
    counter = 0,
    master = list(
      b = paste0("B-", sample(1:15)),
      i = paste0("I-", sample(16:30)),
      n = paste0("N-", sample(31:45)),
      g = paste0("G-", sample(46:60)),
      o = paste0("O-", sample(61:75))
    ) %>%
    unlist() %>%
    `[`(sample(1:75))
  )

  observeEvent(input$new_game, {
    v$current <- NULL
    v$previous <- NULL
    v$counter <- 0
  })

  observeEvent(input$draw, {

    if (v$counter > 74) {
      NULL
    } else {
      v$counter <- v$counter + 1
      v$current <- v$master[v$counter]
      v$previous <- c(v$current, v$previous)
    }
  })

  output$display <- renderUI({
    tagList(
      p(v$current, style = "font-size:50px"),

```

```

        br(),
        p(paste(v$previous, collapse = ", "))
      )
    })
  }

```

```
shinyApp(ui = ui, server = server)
```

Bingo Player

```
library(shiny)
```

```
make_card <- function() {
  out <- data.frame(
    B = sample(1:15, 5),
    I = sample(16:30, 5),
    N = sample(31:45, 5),
    G = sample(46:60, 5),
    O = sample(61:75, 5)
  )

```

```
  out$N[3] <- NA
```

```
  return(out)
}
```

```
ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(
      actionButton("card", "Make a New Card")
    ),
    mainPanel(
      column(2, uiOutput("b")),
      column(2, uiOutput("i")),
      column(2, uiOutput("n")),
      column(2, uiOutput("g")),
      column(2, uiOutput("o"))
    )
  )
)
```

```
server <- function(input, output) {
  v <- reactiveValues(b = NULL, i = NULL, n = NULL, g = NULL, o = NULL)

  observeEvent(input$card, {
    card <- make_card()
    v$b <- card$B
    v$i <- card$I
    v$n <- card$N
    v$g <- card$G
    v$o <- card$O
  })

```

```
  output$b <- renderUI({
    tagList(
      p("B", style = "font-size:25px"),
      checkboxInput("b1", v$b[1]),
      checkboxInput("b2", v$b[2]),

```

```

        checkboxInput("b3", v$b[3]),
        checkboxInput("b4", v$b[4]),
        checkboxInput("b5", v$b[5])
    )
})

output$i <- renderUI({
  tagList(
    p("I", style = "font-size:25px"),
    checkboxInput("i1", v$i[1]),
    checkboxInput("i2", v$i[2]),
    checkboxInput("i3", v$i[3]),
    checkboxInput("i4", v$i[4]),
    checkboxInput("i5", v$i[5])
  )
})

output$n <- renderUI({
  tagList(
    p("N", style = "font-size:25px"),
    checkboxInput("n1", v$n[1]),
    checkboxInput("n2", v$n[2]),
    checkboxInput("n3", "FREE", TRUE),
    checkboxInput("n4", v$n[4]),
    checkboxInput("n5", v$n[5])
  )
})

output$g <- renderUI({
  tagList(
    p("G", style = "font-size:25px"),
    checkboxInput("g1", v$g[1]),
    checkboxInput("g2", v$g[2]),
    checkboxInput("g3", v$g[3]),
    checkboxInput("g4", v$g[4]),
    checkboxInput("g5", v$g[5])
  )
})

output$o <- renderUI({
  tagList(
    p("O", style = "font-size:25px"),
    checkboxInput("o1", v$o[1]),
    checkboxInput("o2", v$o[2]),
    checkboxInput("o3", v$o[3]),
    checkboxInput("o4", v$o[4]),
    checkboxInput("o5", v$o[5])
  )
})
}

shinyApp(ui = ui, server = server)

```

