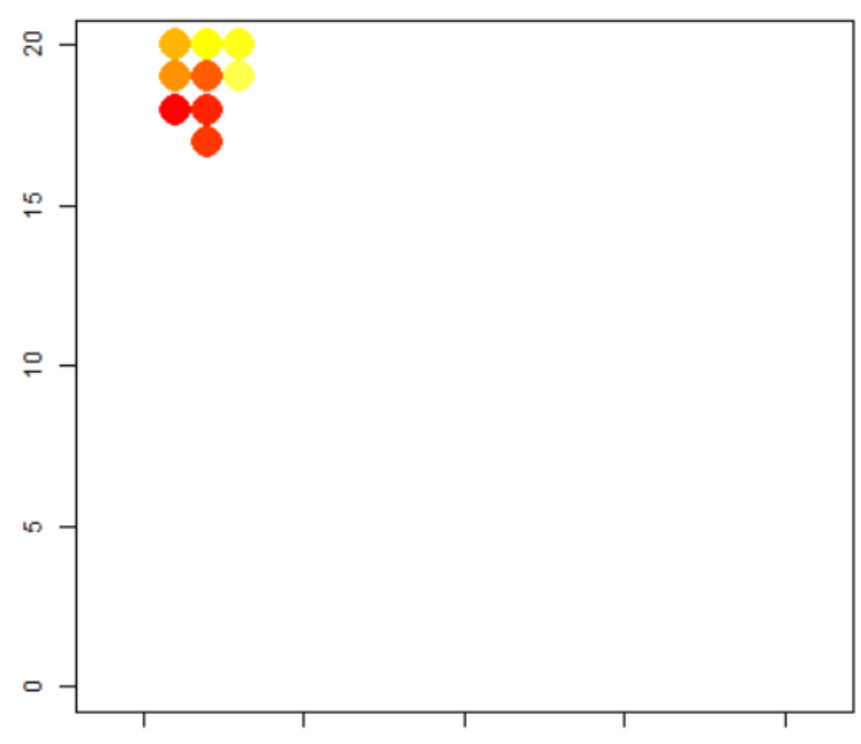
Snake!

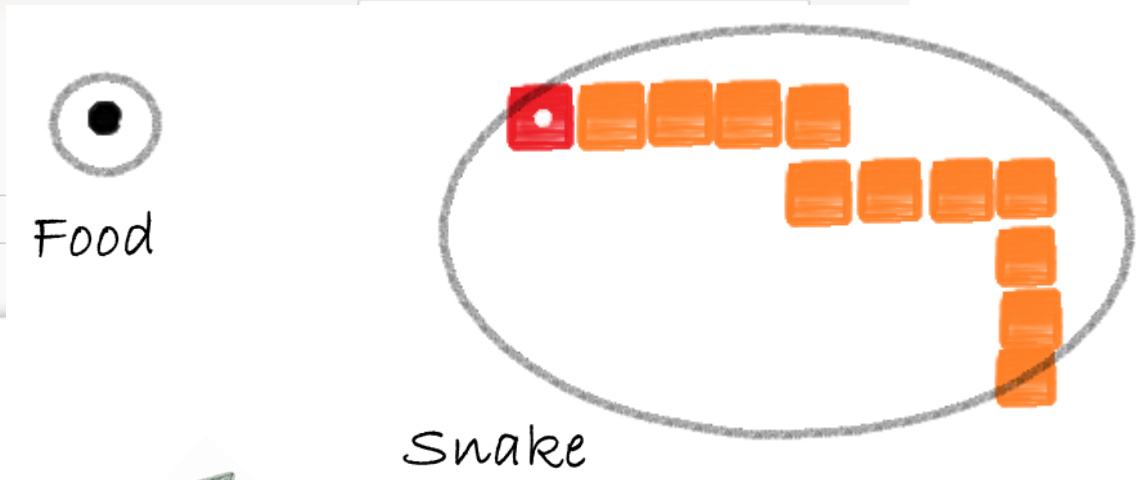
林柏佑 Lin, Po-Yu
Institution of Ecology and Evolutionary Biology
The second year of a master

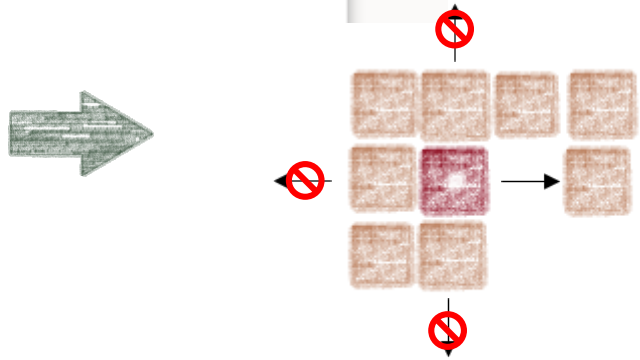
avid Zelený
kipages

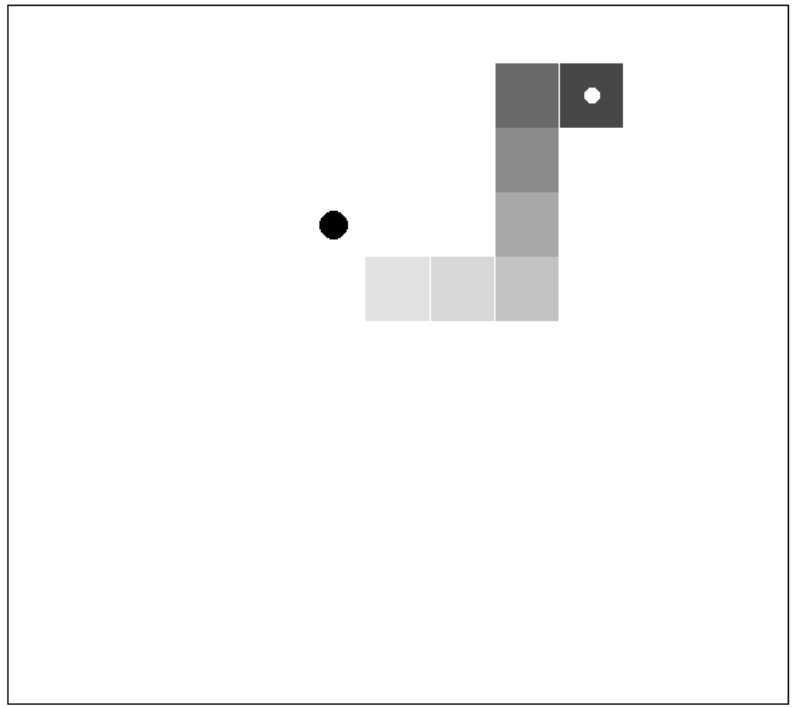
Animation of moving snake

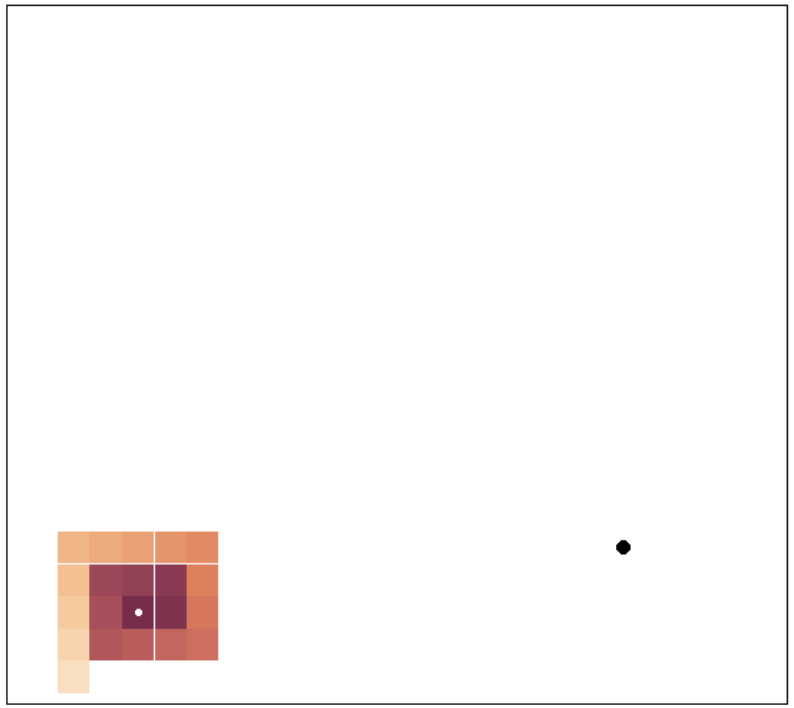
Try to reproduce this:

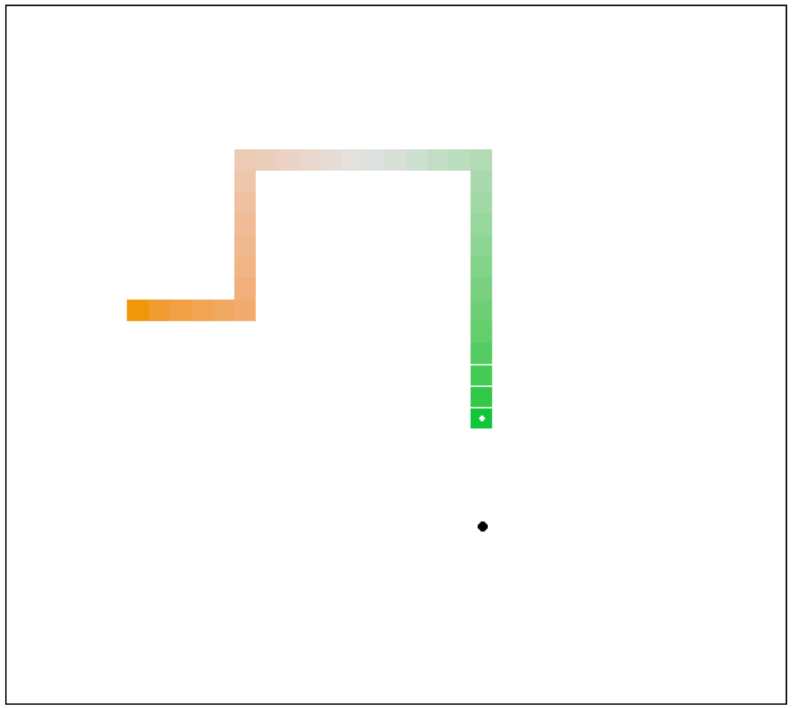












A function:
animated.snake()

The arguments:
color, moving mode, speed, size of playground

https://imgur.com/a/3SEwmDI

R code

```r
animated.snake <- function(snake.length = 3, square.size = 20, no.step = 0, speed = 300, mode = "shortest",
color = "heat"){
  # initiate -----
  # ----
  snake <- matrix(NA, ncol = 2, nr
  snake[1,] <- c(round(square.size
  for (i in 2:snake.length){
    snake[i,] <- snake[i-1,] + c(
  }
  body <- paste(snake[,1], snake[,
  fd <- outer(0:square.size, 0:squ
  food <- unlist(strsplit(sample(t
  food <- as.numeric(c(food[1], fo
  eat.food <- FALSE

  snake.food <- rbind(food, snake
    # randon ----
    if (mode == "random"){
      # boundary & body
      # body <- str_c(snake[-1,1]
      if ((snake[2,2] >= square.s
nt[rownames(nt) != "up",] # up
      if ((snake[2,2] <= 0) | (pa
= "down",] # down
      if ((snake[2,1] <= 0) | (pa
= "left",] # left
      if ((snake[2,1] >= square.s
nt[rownames(nt) != "right",] # ri
      # no way
      if (is.null(nrow(nt))) {tit
      snake[1,] <- snake[2,] + nt
    }
    # shortest ----
    else if (mode == "shortest"){
      if (food[1] > snake[2,1]){ #
        if ((!paste(snake[2,1]+1,
nt["right",] # can go right
        else if (food[2] <= snake[
          if ((!paste(snake[2,1],
can go down
          else if ((!paste(snake[2
<- nt["up",] # cna't go right and
          else if ((!paste(snake[2
nt["left",] # can't go right, down
          else {title(main = "No w
        }
        else{ # can't go right, fo
          if ((!paste(snake[2,1],
nt["up",] # can go up
          else if ((!paste(snake[2
nt["down",] # can't go right and u
          else if ((!paste(snake[2
nt["left",] # can't go right, up a
          else {title(main = "No w
        }
      }
      else if (food[1] < snake[2,1
      else if (food[2] < snake[2,2
      else{⟺}
      snake[1,] <- snake[2,] + nt
    }
    # clockwise ----
    else if (mode == "clockwise"){
      clock_count <- 0
      while (TRUE){
        # just food
        if (clock_count == 1){
          if (clock_dir == "left"){ # check le
            if ((snake[2,1] <= 0) | (paste(snak
"up"; clock_count <- clock_count + 1} # not go
            else {nt <- nt[clock_dir,]; break}
          }
          else if (clock_dir == "up"){ # check
            if ((snake[2,2] >= square.size) |
{clock_dir <- "right"; clock_count <- clock_cou
            else {nt <- nt[clock_dir,]; break}
          }
          else if (clock_dir == "right"){ # che
            if ((snake[2,1] >= square.size) |
{clock_dir <- "down"; clock_count <- clock_cour
            else {nt <- nt[clock_dir,]; break}
          }
          else if (clock_dir == "down"){ # chec
            if ((snake[2,2] <= 0) | (paste(snak
"left"; clock_count <- clock_count + 1} # not
            else {nt <- nt[clock_dir,]; break}
          }
        }
        else if (clock_count == 2){⟺}
        else if (clock_count == 3){⟺}
        else{⟺}
      }
      if (clock_count == 4) {title(main = "No
```