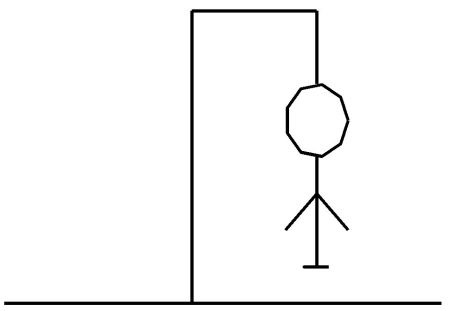
Hangman is a classic word game in which you need to need to guess as many possible letters in word, so you can guess the word, before running out of tries (lives).

Upon running out of tries, you are hanged!



The game can be played in R Studio, where the user inputs new letters in console, and the picture is being drawn. The picture consists of 7 false tries, so  it is drawn in 7 steps.

The diagram is created using simple X, Y coordinates with groups for determining the steps:

level1 <- data.frame(x=c(1,2,3,4,5,6,7,8), y=c(1,1,1,1,1,1,1,1), group=c(1,1,1,1,1,1,1,1))

level2 <- data.frame(x=c(4,4,4,4,4), y=c(1,2,3,4,5),group=c(2,2,2,2,2))

level3 <- data.frame(x=c(4,5,6), y=c(5,5,5), group=c(3,3,3))

level4 <- data.frame(x=c(6,6), y=c(5,4), group=c(4,4))

level5 <- drawHead(c(6,3.5),1,10,5)

level6 <- data.frame(x=c(6,6,5.8,6.2),y=c(3,1.5,1.5,1.5), group=c(6,6,6,6))

level7 <- data.frame(x=c(5.5,6,6.5),y=c(2,2.5,2), group=c(7,7,7))

levels <- rbind(level1,level2,level3,level4,level5,level6,level7)

Drawing itself is created by using a simple function using ggplot2 library:

drawMan <- function(st\_napak) {

ggplot(levels[which(levels$group<=st\_napak),], aes(x=x, y=y, group=group)) +

geom\_path(size=2.5) +

theme\_void()

}

The function draws the hanging man in 7 steps

All the rest of the logic is fairly simple, continue until you find the correct word, or until you are hanged. Section of the code:

beseda <- readline(prompt="Word: ")

iskana\_beseda <- replicate(nchar(beseda),'\_')

while (active == TRUE) {

if (i == 0) {

writeLines(paste(iskana\_beseda, collapse = " "))

}

crka <- readline(prompt="Enter Letter: ")

izbor <- rbind(izbor, crka)

#iskana\_beseda

if (grepl(crka, beseda) == TRUE) {

cilj <- rbind(cilj, crka)

iskana\_beseda <- zamenjaj2(beseda, crka)

#print(zamenjaj2(beseda, crka))

print(paste("Yay!","Try N:",i+1,"Wrong letters: {",(toString(paste0(cilj\_n, sep=","))),"}"))

if (as.character(paste(iskana\_beseda, collapse = "")) == beseda) {

active == FALSE

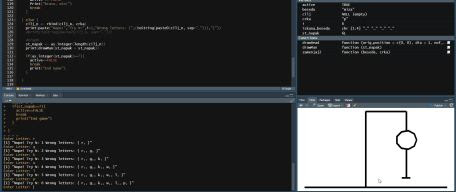
Print("Bravo, win!")

break

}

{code continues.....}

… and the rest of there code is attached on this repository



The Full R Code For Hangman Game:

#### ToDo: Write checker for existing letters -OK

#### ToDo: function for adding repetative same letters must be added - OK

#### ToDo: check small and capital letter for word and letters

#####################################

library(ggplot2)

#######################

### Helper functions

#######################

zamenjaj2 <- function(beseda, crka, iskana\_beseda){

if (regexpr(crka, beseda)[1] > 0) {

#pozicija <- regexpr(crka, beseda)[1]

#iskana\_beseda[pozicija] <- crka

beseda <- base::tolower(beseda)

crka <- base::tolower(crka)

pozicije <- which(strsplit(beseda, "")[[1]]==crka)

iskana\_beseda[pozicije] <- crka

message(paste(iskana\_beseda, collapse = " "))

#convert back to single string to check for equality

if (paste(iskana\_beseda, collapse = "") == beseda) {

return(iskana\_beseda)

message("End Game!")

}

return(iskana\_beseda)

}

}

drawHead <- function(orig\_position = c(0,0),

dia = 1,

nof\_points = 10,

group = 5){

vectT <- seq(0,2\*pi, length.out = nof\_points)

r <- dia/2

x\_data <- orig\_position[1] + r \* cos(vectT)

y\_data <- orig\_position[2] + r \* sin(vectT)

return(data.frame (x = x\_data, y = y\_data, group = group))

}

drawMan <- function(st\_napak) { #, iskana\_beseda) {

ggplot(levels[which(levels$group <= st\_napak), ], aes(x = x, y = y, group = group)) +

geom\_path(size = 2.5) +

theme\_void() +

ggtitle('Classic Hangman Game')

#ggtitle('Hangman Game, word is: ', iskana\_beseda)

}

CheckDuplicate <- function(crka, izbor){

if (length(izbor) == 0) {

message('Zero length; adding...')

izbor <- rbind(izbor, izb=crka)

izbor[,1] <- as.character(izbor[,1])

} else {

if (grepl(crka,izbor) == TRUE) {

izbor[,1] <- as.character(izbor[,1])

message("exists")

} else {

izbor <- rbind(izbor, izb=crka)

izbor[,1] <- as.character(izbor[,1])

message("added...")

}

}

return(izbor)

}

########################

###### Data for graph

########################

level1 <- data.frame(x = c(1, 2, 3, 4, 5, 6, 7, 8),

y = c(1, 1, 1, 1, 1, 1, 1, 1),

group = c(1, 1, 1, 1, 1, 1, 1, 1))

level2 <- data.frame(x = c(4, 4, 4, 4, 4),

y = c(1, 2, 3, 4, 5),

group = c(2, 2, 2, 2, 2))

level3 <- data.frame(x = c(4, 5, 6), y= c (5, 5, 5), group = c(3, 3, 3))

level4 <- data.frame(x = c(6, 6), y = c(5, 4), group = c(4, 4))

level5 <- drawHead(c(6, 3.5), 1, 10, 5)

level6 <- data.frame(x = c(6, 6, 5.8, 6.2),

y =c(3, 1.5, 1.5, 1.5), group = c(6, 6, 6, 6))

level7 <- data.frame(x = c(5.5, 6, 6.5), y = c(2, 2.5, 2), group = c(7, 7, 7))

levels <- rbind(level1, level2, level3, level4, level5, level6, level7)

rm(level1, level2, level3, level4, level5, level6, level7)

########################

### Helper variables

########################

suppressWarnings(rm(st\_napak, izbor, crka, cilj, cilj\_n, iskana\_beseda, beseda, i, active))

st\_napak = 0

i = 0

izbor = data.frame(izb=c(NULL))

cilj = NULL

cilj\_n = data.frame(izb=c(NULL))

active = TRUE

########################

## Hangman

#######################

StartNewGame <- function(sensitive.flag = TRUE) { # sensitive.flag: TRUE -> capital letters are available.

beseda <- readline(prompt = "Word: ")

cat("\f")

graphics.off()

if (sensitive.flag == FALSE) {

beseda <- base::tolower(beseda)

}

iskana\_beseda <- replicate(nchar(beseda), '\_')

while (active == TRUE) {

if (i == 0) {

writeLines(paste(iskana\_beseda, collapse = " "))

}

crka <- readline(prompt="Enter Letter: ")

if (nchar(crka)>1) message("Taking first letter")

crka <- substr(crka, 1, 1)

izbor <- CheckDuplicate(crka, izbor)

#iskana\_beseda

if (grepl(crka, beseda) == TRUE) {

cilj <- rbind(cilj, crka)

iskana\_beseda <- zamenjaj2(beseda, crka, iskana\_beseda)

message(paste("Yay!","Try N:",i+1))

if (as.character(paste(base::tolower(iskana\_beseda), collapse = "")) == base::tolower(beseda)) {

active == FALSE

message("Bravo, win!")

break

}

} else {

cilj\_n <- CheckDuplicate(crka=crka, izbor=cilj\_n)

message(paste("Nope!","Try N:",i + 1, "Wrong letters: {",(toString(paste0(cilj\_n[,1], sep = ","))),"}"))

#Graph

st\_napak <- as.integer(nrow(cilj\_n))

print(drawMan(st\_napak = st\_napak))#,iskana\_beseda=paste(iskana\_beseda, collapse = " ") ))

if(as.integer(st\_napak) == 7){

active == FALSE

break

message("End Game")

}

}

i= i + 1

if(st\_napak == 7){

active == FALSE

break

message("End game")

}

}

}

#####################

### Start new Game

#####################

StartNewGame()