# The game "Cross the road"

In this tutorial we will develop a graphic game that presents the player with the challenge of crossing very busy streets ... This game is a simplified version of the classic [Crossy Road](https://en.wikipedia.org/wiki/Crossy_Road), a smartphone game launched in 2014. The name and concept of the game was based on the joke '' Why did the chicken cross the road? "

A picture containing toy, truck, orange, green

Description automatically generated

Crossy Road - [wikipedia](https://en.wikipedia.org/wiki/Crossy_Road)

# Game configuration

This game will be developed in a graphic environment, using the p5 module.

We will define these three constants:

* LARGURA - width of the graphic window;
* ALTURA = height of the graphic window;
* NIVEL = definition of the game level layout, defining the roads and their direction (1 to the right or -1 to the left) or the lanes without traffic (0).

And these are calculated based on the previous ones:

* DELTA - the grid dimensions;
* COMP = number of gridlines on the roads.

And we will also need global variables:

* jogador - player's position;
* filas - list with traffic queues;
* timer - timer - stores the seconds of the last update.

Start your source code file with the following instructions:

from p5 import \*

import random

# definições globais

LARGURA = 600

ALTURA = 400

NIVEL = [0, -1, -1, 1, 1, 0, -1, -1, 1, 0, -1, 1, 0] # 0-passeio; 1-direita, -1-esquerda

DELTA = ALTURA // len (NIVEL) # dimenções da quadrícula

COMP = LARGURA // DELTA # numero de quadrículas das estradas

# variáveis globais

jogador = []

filas = []

timer = 0

# Abstraction "Fila" (Queue)

The game involves "traffic queues", so the most suitable abstraction to perform it will be the "Fila" (Queue). This is an abstraction that allows you to store a sequence of data, allowing for easy insertion of data at the end of the queue and easy extraction of information from the beginning of the queue. It is thus a FIFO (First In First Out) structure.

There is no data abstraction in Python that fully implements the concept of Queue, so we will use the Queue abstraction defined in section 10.2:

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

# ABSTRAÇÃO FILA #

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

# construtor

def cria\_fila():

return []

# seletores

def fila\_vazia(fila):

if len (fila) == 0:

return True

else:

return False

def frente\_da\_fila(fila):

if not fila\_vazia(fila):

return fila[0]

else:

return False

def cauda\_da\_fila(fila):

if not fila\_vazia(fila):

return fila[-1]

else:

return False

# modificadores

def entra\_na\_fila(fila, elem):

fila.append(elem)

def sai\_da\_fila(fila):

if not fila\_vazia(fila):

return fila.pop(0)

else:

return False

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

# Game structure

The game will be structured based on the pair of functions associated with the p5 module:

* setup() - game setup;
* draw() - game loop.

And also another function for the control of the game by the player through the keyboard:

* key\_pressed() - responds to key press events.

We will also do the decomposition of the base problem, in subproblems that facilitate the creation of the game cycle:

* desloca() - move a point according to a vector defined by (dx, dy);

# desloca jogador

def desloca(p, dx, dy):

p[0]+= dx

p[1]+= dy

* ponto() - calculates the screen coordinates of a given grid cell, defined by (linha, coluna)

# calcula as coordenadas do ecrã

def ponto(q):

return (q[0]\*DELTA + DELTA//2, q[1]\*DELTA + DELTA//2)

* fora() - checks if a particular vehicle is off the screen;

# verifica se um automóvel está fora

def fora (carro):

return carro[0] < 0 or carro[0] > COMP

* trafego() - updates vehicle queues by moving vehicles.

The algorithm for this function is as follows:

1. Determine whether the 1 second delay has passed. Otherwise, ignore the rest of the function, returning True;
2. Go through all traffic queues (-1 or 1) and, for each traffic queue, move all vehicles;
3. If the vehicle happened to run over the player, then return False, indicating the end of the game;
4. If a queue has less than 10 cars, insert a new car in the queue;
5. If a car is left off the screen, it is removed from the queue.

Enter the code corresponding to the function trafego().

# atualiza filas de veículos

def trafego():

global jogador, filas, timer

# timer

if second()==timer:

return True

else:

timer = second()

# gere filas

for i in range(len(NIVEL)-1):

if NIVEL[i] != 0:

# move fila de automóveis

for carro in filas[i]:

desloca(carro, NIVEL[i], 0)

if carro == jogador:

# se carro atropela jogador retorna falso

return False

# insere novos carros até um máximo de 10

if len(filas[i]) < 10:

ultimo = cauda\_da\_fila(filas[i])

if ultimo:

novo = ultimo[::]

desloca(novo, random.randint(-5, -2)\*NIVEL[i], 0)

elif NIVEL[i] > 0:

novo = [random.randint(-2,0), i]

else:

novo = [COMP + random.randint(0, 2), i]

entra\_na\_fila(filas[i], novo)

# remove carros que saiam do ecrã

if fora(frente\_da\_fila(filas[i])):

sai\_da\_fila(filas[i])

return True

# Game setup

The game setup follows the following algorithm:

1. creates a graphic window;
2. creates the player's avatar on the bottom line of the game level;
3. creates the game level traffic queues;
4. add traffic to the traffic queues ... so that the game does not start with empty roads.

# inicialização do jogo

def setup():

global jogador, filas

# janela

title("Travessia da Estrada")

size(LARGURA, ALTURA)

no\_stroke()

# jogador

jogador = [COMP // 2, len(NIVEL)-1]

# filas de automóveis

for i in range(len(NIVEL)-1):

filas.append(cria\_fila())

# insere tráfego nas filas

for i in range(10):

timer = -1

trafego()

|  |  |
| --- | --- |
| **Head with gears** | In this last for loop the value of variable timer is assigned -1.  Since function seconds() returns values between 0 and 59, why is it necessary to use -1?  *Hint: Observe the first step of function* trafego()... |

# Game loop

The game loop follows the following algorithm:

1. Check if the player has won (reached the starting line (0). In that case the game ends with the screen painted in green.
2. Draw the game level: green background with black roads;
3. Draw the player's avatar through a yellow circle;
4. Move the traffic queues, using the traffic function, and draw the cars. If the player is run over, the player's avatar is repositioned on the last line, as initially.

Type the following code:

def draw():

global jogador

# verifica se terminou o nível

if jogador[1] == 0:

background("green")

circle(ponto(jogador), DELTA)

no\_loop()

return

# desenha nível - fundo verde com estradas a preto

background("green")

fill ("black")

oy = 0

for faixa in NIVEL:

if faixa != 0:

# rectangulo da faixa

rect((0,oy), LARGURA, DELTA)

oy += DELTA

# jogador

fill("yellow")

circle(ponto(jogador), DELTA)

# atualiza filas de tráfego

if trafego():

# desenha veículos

fill("blue")

for fila in filas:

for carro in fila:

rect(ponto(carro), DELTA, DELTA//2)

else:

# jogador atropelado - reinicia

jogador = [COMP // 2, len(NIVEL)-1]

# Input of the player

The player interacts with the game using the cursor keys, which allow her/him to move the avatar.

The code is as follows, which is based on a multiple selection:

# controla a interação com o utilizador através do teclado

def key\_pressed():

global jogador

# verifica teclas premidas

if key == "UP":

desloca(jogador, 0, -1)

elif key == "DOWN":

if jogador[1] < len(NIVEL)-1:

desloca(jogador, 0, 1)

elif key == "LEFT":

if jogador[0] > 0:

desloca(jogador, -1, 0)

elif key == "RIGHT":

if jogador[0] < COMP-1:

desloca(jogador, 1, 0)

if \_\_name\_\_ == '\_\_main\_\_':

run()

# Final challenge

A similar type of game is the classic [*Speed Race*](mailto:https://www.arcade-museum.com/game_detail.php?game_id=9709) (1974) from Taito. It is a vertical development game, where the objective is to overtake all competing cars, without hitting them or leaving the track.

Based on the current game code develop your version of Speed Race. Analize the original [gameplay](mailto:https://www.youtube.com/watch?v=5jc9_xrD2_M) on youtube.

|  |  |
| --- | --- |
| **Share with person** | **Share** your game! |