# Vytvorenie pracovných programov.

priprava.py – vytvorenie pracovnej hracej plochy

fruit.py – generovanie pozície ovocia

*cervik.py –*

červík sa pohybuje doprava ako je určené v riadkoch 40 a 41

direction = 'RIGHT'

change\_to = direction

až kým mu kurzormi neurčíme ináč

V programe nie sú riešené otázky ohraničenia plochy a teda že červík môže ukončiť hru tým že narazí do steny.

Posun sa uskutočňuje po 10 dielov a pomocou príkazu pop sa vždy usekne zadná časť to z listu ktorá prezentuje dĺžku hadíka

Napríklad pohyb doprava je reprezentovaním vykonaním príkazu

snake\_position[0] += 10 a teda zvýšením x ovej súradnice o 10

*vkladanie\_suradnic.py* – program pre bližšie vysvetlenie vytvárania list súradníc červíka

doplnky: prerušenie behu úlohy – v hlavnom programe

from pygame.locals import \* - pre symboly typu QUIT

3.6.2025 -

*ovovcie a had.py* – spojenie programov fruit.py a cervik.py bez kontrol

Vysvetlenie situácie umiestnenia kontroly zväčšenia hadíka po zjedení ovocia.

Na začiatok zoznamu súradníc telá hadika sa pridá nová pozícia ktorá bola vytvorená posunom klávesnice smerového riadenia a hadíkovi sa usekne koniec poslednej súradnice ak sa mu nepodarí zjesť ovocie.

snake\_body.insert(0, list(snake\_position))

snake\_body.pop()

*koniec.py* – kópiou ovovcie a had.py a doplnením funkcie game\_over() ukončí akúkoľvek aktuálnu hru a ukončí kód Pythonu bežiaci v tomto module (za predpokladu, že ho nevolajú žiadne iné funkcie).

Na ukončenie hry sa používa testy.

## Otázky na riešenie - Snake.

Kontrola kolízie s hraniciami hracej plochy.

if snake\_position[0] < 0 or snake\_position[0] > window\_x-10:

game\_over()

if snake\_position[1] < 0 or snake\_position[1] > window\_y-10:

game\_over()

Ako skontrolovať, že hadík zjedol ovocie ?

**snake\_body.insert(0, list(snake\_position))**

if snake\_position[0] == fruit\_position[0] and snake\_position[1] == fruit\_position[1]:

score += 10

fruit\_spawn = False

else:

snake\_body.pop()

( ak pozícia hadíka x-ová súradnica **snake\_position[0]** je rovná x-ovej súradnici ovocia **fruit\_position[0]** a (and) to isté platí pre y-ové súradnice.)

Ako zistiť, že hlava hadíka narazila do tela ?

# Dotyk tela hada

for block in snake\_body[1:]:

if snake\_position[0] == block[0] and snake\_position[1] == block[1]:

game\_over()

Premenná **snake\_position** je hlava hadíka a kontroluje, či jej súradnice sú rovné blokom z premennej **snake\_body**, pričom prvý block tela hadíka je hlava teda

10.6.2025

V časti kódu kde sa podarí hadíkovi zjesť ovocie a teda okrem pridania novej hlavy

snake\_body.insert(0, list(snake\_position))

sa navýši skóre o 10. Základná hodnota premennej je nula.

score += 10

Keďže bolo zjedené ovocie zmení sa logická hodnota premennej pre zmenu pohybu a vygenerovania nového ovocia na nepravda.

fruit\_spawn = False

Text vykresľuje premenná game\_over\_surface

Skóre:

def show\_score(choice, color, font, size):

# creating font object score\_font

score\_font = pygame.font.SysFont(font, size)

# create the display surface object

# score\_surface

score\_surface = score\_font.render('Score : ' + str(score), True, color)

# create a rectangular object for the text

# surface object

score\_rect = score\_surface.get\_rect()

# displaying text

game\_window.blit(score\_surface, score\_rect)

## Nový zdroj nápadov.

Príklad z knihy Making Games with Python & Pygame napríklad na linke: <https://inventwithpython.com/pygame/>

Programy na opravovanie:

<https://inventwithpython.com/pygame/buggy/>

### Wormy.

Prepracovaná verzia hry červík už presnými definíciami jednotlivých funkcií jasnou štruktúrou programu programovacom jazyku Python