# 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.

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)

## Vysvetlenie kódu:

<https://www.geeksforgeeks.org/snake-game-in-python-using-pygame-module/>

1. Kód začína importom potrebných knižníc.
2. Sú to pygame, čas a náhoda. (pygame, time a random)
3. Ďalej kód definuje niektoré premenné.
4. Premenná **snake\_speed** riadi, ako rýchlo sa had pohybuje po obrazovke.
5. Premenné **window\_x a window\_y** definujú veľkosť herného okna na obrazovke.
6. Nasledujúci riadok kódu inicializuje pygame.
7. Je to dôležité, pretože nastavuje všetky herné objekty a ich vlastnosti tak, aby ich bolo možné neskôr použiť v programe.
8. Ďalej kód vytvorí inštanciu objektu triedy GameWindow pygame.
9. Tento objekt predstavuje obdĺžnikovú oblasť na obrazovke, ktorú je možné vyplniť grafikou a textovým obsahom.
10. Objekt **GameWindow** má dve vlastnosti: šírku a výšku.
11. Tieto hodnoty znázorňujú, aké široké a vysoké je herné okno.
12. Nasledujúci riadok kódu priraďuje hodnoty týmto vlastnostiam na základe hodnoty definovanej používateľom s názvom **snake\_speed**.
13. Táto premenná hovorí, ako rýchlo (v pixeloch za sekundu) pohybovať hadom po obrazovke.
14. Vyššie hodnoty umožnia rýchlejší pohyb, ale aj intenzívnejšiu hrateľnosť!
15. Ďalej PyGame začne načítavať rôzne obrázky do pamäte, ktoré sa použijú ako grafika na pozadí pre náš herný svet.
16. Kód vytvorí okno s rozmermi 720x480 pixelov.
17. Farby čierna, biela, červená, zelená a modrá budú použité na znázornenie rôznych prvkov hry.
18. Ďalej sa importuje a inicializuje modul pygame. (pygame.init())
19. To nám umožní začať pracovať s rôznymi objektmi a funkciami hry.
20. Hlavná slučka hry sa potom spustí volaním pygame.init(). Táto funkcia zabezpečí, že všetky potrebné moduly sú načítané a pripravené na použitie.
21. Nakoniec zavoláme konštruktor okna, aby vytvoril naše herné okno.
22. Kód začína vytvorením funkcie pygame.display.set\_mode() na nastavenie veľkosti a polohy okna.
23. Kód potom vytvorí herné okno a nastaví jeho režim na (0, 0).
24. Ďalej kód definuje niektoré premenné: **fps, snake\_position, snake\_body a fruit\_position**.
25. Tieto premenné sa použijú na kontrolu rýchlosti hada, odkiaľ začína (snake\_position), aká je široká (**snake\_body**), kde sa nachádza ovocie (fruit\_position) a či by sa ovocie malo alebo nemalo rozmnožovať (fruit\_spawn).
26. Ďalší blok kódu vypočíta vzdialenosť medzi jednotlivými bodmi na obrazovke pomocou pygame.time.Clock(). (**fps**)
27. To nám umožňuje pohybovať hadom na obrazovke bez toho, aby sme museli neustále prepočítavať jeho polohu.
28. Nakoniec nastavíme dve booleovské premenné: **fruit\_spawn a analyze()**.
29. Tie určia, či sa ovocie objaví na náhodných miestach na obrazovke a bude analyzované pre vstup hráča.
30. Kód nastavuje základné herné okno s hadom umiestneným na (100, 50) na osi X a (window\_x, window\_y) na osi Y.
31. Ovládač FPS je inicializovaný a nastavený tak, aby bežal rýchlosťou 60 snímok za sekundu.
32. Ďalší blok kódu definuje telo hada.
33. Vytvorí sa zoznam desiatich [100, 50] bodov, počnúc pozíciou (100, 50).
34. Prvé štyri body sú nastavené tak, aby boli v strede hadieho tela, zatiaľ čo zvyšných šesť bodov je rovnomerne rozmiestnených okolo neho.
35. Ďalej je pozícia ovocia definovaná ako [(random.randrange(1, (window\_x//10)) \* 10), (random.randrange(1 , (window\_y//10)) \* 10]
36. Kód začína inicializáciou niektorých premenných. (**fruit\_spawn = True, direction = 'RIGHT', change\_to = direction**)
37. Prvým je skóre, ktoré začína na 0.
38. Druhým je smerová premenná (**direction**), ktorá určí, ako sa had bude pohybovať.
39. Funkcia show\_score() sa volá vždy, keď hráč urobí voľbu.
40. Táto funkcia sa skladá z troch častí: vytvorenie objektu písma, vytvorenie objektu zobrazovanej plochy a zobrazenie textu na zobrazovacej ploche.
41. Najprv sa vytvorí score\_font objekt.
42. Tento objekt ukladá informácie o písme použitom na zobrazenie textu na obrazovke (v tomto prípade Times New Roman).
43. Ďalej sa vytvorí score\_surface objekt a inicializuje sa s informáciami o písme a veľkosti textu, ktorý sa zobrazí (veľkosť 50 bodov).
44. Nakoniec sa pomocou blit() objekt score\_rect skopíruje na score\_surface objekt, aby sa mohol zobraziť na obrazovke.
45. Funkcia game\_over() ukončí akúkoľvek aktuálnu hru a ukončí kód Pythonu bežiaci v tomto module (za predpokladu, že ho nevolajú žiadne iné funkcie).
46. Najprv sa vytvorí inštancia písma SysFont s názvom my\_font.
47. Potom sa zadá veľkosť 50 bodov pre Times New Roman ako jeho typ písma a farebné hodnoty.
48. Nakoniec sa zavolá game over(), aby ukončil všetky hry a ukončil kód Pythonu spustený.
49. Kód najprv inicializuje niektoré premenné vrátane premennej skóre.
50. Kód potom vytvorí funkciu s názvom show\_score().
51. Táto funkcia sa použije na zobrazenie aktuálneho skóre na obrazovke.
52. Funkcia show\_score() najprv vytvorí objekt písma s názvom score\_font a nastaví jeho veľkosť na 50 bodov.
53. Ďalej funkcia vytvorí objekt zobrazovanej plochy s názvom score\_surface a nastaví jeho farbu na bielu.
54. Nakoniec funkcia show\_score() vyfúkne score\_surface objekt na obrazovku okna hry.
55. Funkcia game over() je zodpovedná za čistenie zdrojov po skončení hry.
56. Najprv vytvorí objekt písma s názvom my\_font a nastaví jeho veľkosť na 20 bodov.
57. Kód najprv vytvorí objekt textovej plochy s názvom **game\_over\_surface.**
58. **Funkcia blit()** základným spôsobom, ako vykresliť jeden povrch (surface) na druhý. Používa sa na prenesenie pixelov z jedného obrázka alebo povrchu na cieľový povrch, napríklad na obrazovku hry.
59. Text sa vykreslí v písme my\_font a červenou farbou.
60. Ďalej sa vytvorí obdĺžnikový objekt pre objekt textového povrchu.
61. Tento objekt bude mať svoj stred v (window\_x/2, window\_y/4).
62. Nakoniec sa poloha textu na obdĺžniku nastaví pomocou game\_over\_rect.midtop().
63. Kód vytvorí objekt povrchovej vrstvy textu s názvom game\_over\_surface.
64. Tento objekt sa použije na zobrazenie skóre hráča a správy "Vaše skóre je :".
65. Ďalej sa vytvorí obdĺžnikový objekt s názvom game\_over\_rect.
66. Tento objekt sa použije na umiestnenie textu na povrchu.
67. Stred obdĺžnika je nastavený na (window\_x/2, window\_y/4).
68. Kód začína inicializáciou knižnice pygame.
69. Ďalej kód vytvorí okno a priradí ho k game\_window.
70. Okno má povrch (grafické znázornenie obrazovky) a objekt Obdĺžnik, ktorý určuje jeho veľkosť a polohu.
71. Ďalej kód prenesie text "GAME OVER" na game\_over\_surface objekt.
72. Text je nakreslený bielou farbou v strede nad game\_over\_rect objektom.
73. Program potom nastaví časovač, ktorý pobeží 2 sekundy.
74. V tomto bode sa program ukončí, pretože už nie je potrebné spustiť žiadny kód.
75. Kód skontroluje kľúčové udalosti a ak udalosť zodpovedá platnému kľúču, zodpovedajúcim spôsobom zmení text zobrazený na obrazovke.
76. Ak stlačíte akékoľvek iné tlačidlo, program bude pokračovať ako zvyčajne.
77. Kód začína kontrolou, či hráč stlačil dve klávesy súčasne.
78. Ak áno, kód zmení smer hada.
79. Ďalej kód skontroluje, či niektorý z klávesov nebol stlačený iným smerom, ako sa očakávalo.
80. Ak áno, potom kód zodpovedajúcim spôsobom upraví polohu hada.
81. Nakoniec aktualizuje, aké veľké je hadie telo.
82. Kód skontroluje, či sú dve súčasne stlačené klávesy "HORE" alebo "DOLE".
83. Ak sú, smer hada sa zodpovedajúcim spôsobom zmení.
84. Ak dve stlačené klávesy nie sú rovnaké, kód skontroluje, či sú v rôznych smeroch.
85. Ak nie sú, poloha hada sa upraví o 10 pixelov v každom smere.
86. Nakoniec sa vytvorí funkcia, ktorá zmení veľkosť hadieho tela pri pohybe.
87. Kód začína vytvorením zoznamu pozícií hadov.
88. Prvá pozícia v zozname je (0, 0) a posledná pozícia v zozname je (window\_x-10, window\_y-10).
89. Ďalej kód skontroluje, či sa niektorá z pozícií v hadovi rovná polohe ovocia.
90. Ak áno, potom toto ovocie získa 10 bodov a pridá sa k premennej ovocného spawnu.
91. Ak sa nenájdu žiadne ovocie, hra sa presunie na kontrolu kolízií medzi hadmi a ovocím.
92. Ak sa dva hady pretínajú, ich skóre sa zvýši o 10.
93. Ak sa had zrazí so stenou alebo iným hadom, potom tento had zomrie a spustia sa podmienky prehry.
94. Nakoniec, dotyk ktorejkoľvek časti hada spôsobí jeho smrť a tiež spustí podmienky hry.
95. Kód skontroluje, či sú dve polohy v hadom tele rovnaké.
96. Ak áno, skóre sa zvýši o 10 a zavolá sa funkcia game\_over().
97. Ak sa hráč v ktoromkoľvek bode dotkne hadieho tela, vyvolá sa funkcia game\_over().