

version1

[index](#)
[d:\an3_sem1\python\project\version1.py](#)

@autor: Jinga Andreea

Modules

[random](#)

[tkinter](#)

Functions

calculate_points(last_modified_hole_position, player)

Functia calculeaza punctele castigate de jucator intr-o runda astfel: se considera puncte castigate pietrele din toate casutele consecutive, incepand cu ultima casuta modificata de jucator, care contin 2 sau 3 pietre. Toate casutele din care se calculeaza punctele trebuie sa fie ale adversarului.

:param last_modified_hole_position: ultima casuta in care jucatorul a pus o piatra

:param player: jucatorul care a facut mutarea

:return: numarul de puncte castigate

choose_to_move1(position_button, canvas, main_window)

Functie apelata de jucatorul doi, corespunzand casutelor din partea de sus a tablei. Functia valideaza mutarea aleasa de jucator si afiseaza un mesaj de eroare corespunzator in caz de invaliditate. Daca miscarea este corecta realizarea mutarea efectiva a pietrelor si calculeaza punctajul obtinut.

:param position_button: mutarea aleasa de jucator

:param canvas: o referinta la obiectul canvas pentru redesenarea casutelor

:param main_window: o referinta la obiectul window pentru afisare castigatorului la final

choose_to_move2(position_button, canvas, main_window)

Functie apelata de jucatorul doi, corespunzand casutelor din partea de sus a tablei. Functia valideaza mutarea aleasa de jucator si afiseaza un mesaj de eroare corespunzator in caz de invaliditate. Daca miscarea este corecta realizarea mutarea efectiva a pietrelor si calculeaza punctajul obtinut.

:param position_button: mutarea aleasa de jucator

:param canvas: o referinta la obiectul canvas pentru redesenarea casutelor

:param main_window: o referinta la obiectul window pentru afisare castigatorului la final

draw_board(main_window, no_of_players)

Functia de ocupa de desenarea tablei de joc, a casutelor initializare cu cate 4 pietre si a butoanelor corespunzatoare fiecarei mutari alese de jucatori

:param main_window: o referinta la o variabila de tip window pentru desenare

:param no_of_players: numarul de jucatori, adica daca jocul va fi single-player sau multi-player

draw_holes(canvas)

Functia de ocupa cu desenarea initiala a casutelor cu pietre pe tabla de joc si de redesenarea acestora dupa fiecare modificare adusa numarului de pietre dintr-o casuta (adica dupa o mutare si dupa colectarea punctelor).

:param canvas: o referinta catre obiectul Canvas pe care se vor desena casutele

get_next_position_in_weird_circle(current_position)

Sensul de mutare al pietrelor este contrar acelor de ceasornic. Astfel deplasarea in lista holes cu 12 elemente se va face urmand sensul indicilor: 6, 7, 8, 9, 10, 11, 5, 4, 3, 2, 1, 9. Aceasta functie da urmatoarea pozitie care urmeaza in acest sens al deplasarii.

:param current_position: pozitia initiala pe care ne aflam

:return: pozitia finala dupa deplasare

get_previous_position_in_weird_circle(current_position)

Sensul de mutare al pietrelor este contrar acelor de ceasornic. Astfel deplasarea in lista holes cu 12 elemente se va face urmand sensul indicilor: 6, 7, 8, 9, 10, 11, 5, 4, 3, 2, 1, 9. Aceasta functie da pozitia anterioara, cea de pe care am ajuns pe pozitia curenta.

:param current_position: pozitia pe care ne aflam la inceput

:return: pozitia anterioara

init2()

Functia deseneaza fereastra de start a jocului in care se poate alege numarul de jucatori care vor juca

:return: o referinta de tip window folosita ulterior pentru desenare

init_game()

Functia initializeaza partea de logica din joc, adica lista de pietre si scorul fiecarui jucator

is_game_over()

Functia specifica daca jocul s-a terminat, adica daca vreunul dintre jucatori are mai mult de 24 de pietre dintr-un total de 48.

:return: Functia returneaza True daca jocul s-a terminat, adica daca avem un castigator, si False altfel.

is_opponent_starving(copy_holes, pressed_button, player)

Funcția verifică încălcarea regulii "Adversarul nu trebuie să fie niciodată înfometat". Conform acestei reguli dacă adversarul rămâne fără pietre în urma mutării lui, jucătorul este obligat să aleagă o mutare care să îi ofere pietre adversarului pentru mutarea viitoare. Funcția verifică dacă adversarul se află în situație de înfometare, iar în acest caz simulează mutarea pentru a verifica dacă acesta se va afla în stare de înfometare și după mutare.

:param copy_holes: o copie a listei de casute și pietre pe care se va simula mutarea

:param pressed_button: mutarea aleasă de jucător

:param player: jucătorul care face mutarea

:return: Funcția returnează True în cazul în care în urma mutării adversarul este înfometat, False altfel

main()

Funcția main a programului care pornește fereastra grafică.

make_move_for_bot(canvas, main_window)

Funcția se ocupă de mutarea pietrelor pentru jucătorul doi când acesta este reprezentat de calculator. Calculatorul va alege random o mișcare din cele valide, va muta și îi se vor calcula punctele obținute.

:param canvas: o referință la o variabilă de tip canvas pentru desenarea casutelor

:param main_window: o referință la o variabilă de tip window pentru afișarea câștigătorului

move(starting_position, canvas, player)

Funcția face mutarea efectivă a pietrelor din casuta selectată de jucător, respectând regula Kroo, conform căreia nu se pun pietre în casuta din care s-a început mutarea. La final redesenează casutele și calculează punctele câștigate.

:param starting_position: poziția aleasă de jucător pentru a muta

:param canvas: o referință de tip canvas pentru redesenare

:param player: jucătorul care face mutarea

:return: numărul de puncte câștigate

one_player_game_init(main_window)

Funcția inițializează jocul cu un singur jucător, specificând numărul 1 în apelul funcției de desenare draw_board.

:param main_window: o referință la o variabilă de tip window pentru desenare

play_with_bot(position_button, canvas, main_window)

Funcția ocupă de mișcările în jocul single-player. La apăsarea unui buton de către jucătorul 1, se va realiza mutarea, iar apoi se va realiza și mutarea calculatorului.

:param position_button: mutarea aleasă de jucătorul 1

:param canvas: o referință la o variabilă de tip canvas pentru desenarea casutelor

:param main_window: o referință la o variabilă de tip window pentru afișarea câștigătorului

possible_moves_for_player2_aka_bot()

Funcția realizează o listă de mutări valide pentru jucătorul doi când acesta este reprezentat de calculator.

:return: lista de mutări valide

show_winner(main_window)

Funcția se ocupă cu stabilirea și afișarea câștigătorului jocului. Pentru a stabili câștigătorul verificăm scorul actual al fiecărui jucător, câștigător fiind cel care are peste 24 de puncte. Dacă niciun jucător nu are peste 24 de puncte atunci jocul s-a terminat forțat și fiecare jucător primește pietrele de pe partea lui de tablă. Astfel stabilirea jucătorului se va face după efectuarea acestui calcul. La final fereastra BoardGame va fi înlocuită cu winner_frame

:param main_window: o referință la window-ul principal peste care se va pune winner_frame

:return: fără tip de return.

two_players_game(main_window)

Funcția inițializează jocul cu doi jucători, specificând numărul 2 în apelul funcției de desenare draw_board.

:param main_window: o referință la o variabilă de tip window pentru desenare

valid_move(pressed_button)

Funcția validează o mișcare pe care un jucător vrea să o facă. Conform regulilor o mișcare este validă dacă jucătorul alege pietre din cadrul casutelor lui (de pe partea lui de tablă), dacă acesta nu îl lasă pe adversar fără pietre cu această mutare, și dacă nu îl înfometează pe adversar, adică el deja nu mai are pietre iar jucătorul este obligat să-i ofere.

:param pressed_button: Reprezintă poziția casutei aleasă de jucător pentru a muta

:return: Funcția returnează o tuplă. Primul element al tuplei este True dacă mutarea este validă, și False în caz contrar. Al doilea element al tuplei reprezintă un cod de eroare în caz că mutarea nu este validă, pentru a afișa motivul în mod corespunzător, și None în caz de validitate.

verify_breaking_rule(copy_holes, pressed_button, player)

Funcția validează regula conform căreia adversarul nu poate rămâne fără pietre în urma adunării punctelor obținute după o mutare. Astfel simulăm mutarea jucătorului și verificăm la final starea casutelor adversarului, adică verificăm la final ca măcar o casuta a adversarului să aibă pietre.

:param copy_holes: o copie a listei de casute, care ține evidența pietrelor din casute

:param pressed_button: mutarea care trebuie validată

:param player: jucătorul care a făcut mutarea

:return: True în cazul în care regula este încălcată și False altfel

ACTIVE = 'active'
ALL = 'all'
ANCHOR = 'anchor'
ARC = 'arc'
BASELINE = 'baseline'
BEVEL = 'bevel'
BOTH = 'both'
BOTTOM = 'bottom'
BROWSE = 'browse'
BUTT = 'butt'
CASCADE = 'cascade'
CENTER = 'center'
CHAR = 'char'
CHECKBUTTON = 'checkbutton'
CHORD = 'chord'
COMMAND = 'command'
CURRENT = 'current'
DISABLED = 'disabled'
DOTBOX = 'dotbox'
E = 'e'
END = 'end'
EW = 'ew'
EXCEPTION = 8
EXTENDED = 'extended'
FALSE = 0
FIRST = 'first'
FLAT = 'flat'
GROOVE = 'groove'
HIDDEN = 'hidden'
HORIZONTAL = 'horizontal'
INSERT = 'insert'
INSIDE = 'inside'
LAST = 'last'
LEFT = 'left'
MITER = 'miter'
MOVETO = 'moveto'
MULTIPLE = 'multiple'
N = 'n'
NE = 'ne'
NO = 0
NONE = 'none'
NORMAL = 'normal'
NS = 'ns'
NSEW = 'nsew'
NUMERIC = 'numeric'
NW = 'nw'
OFF = 0
ON = 1
OUTSIDE = 'outside'
PAGES = 'pages'
PIESLICE = 'pieslice'
PROJECTING = 'projecting'
RADIOBUTTON = 'radiobutton'
RAISED = 'raised'
READABLE = 2
RIDGE = 'ridge'
RIGHT = 'right'
ROUND = 'round'
S = 's'
SCROLL = 'scroll'
SE = 'se'
SEL = 'sel'
SEL_FIRST = 'sel.first'
SEL_LAST = 'sel.last'
SEPARATOR = 'separator'
SINGLE = 'single'
SOLID = 'solid'
SUNKEN = 'sunken'

```
SW = 'sw'
TOP = 'top'
TRUE = 1
TclVersion = 8.6
TkVersion = 8.6
UNDERLINE = 'underline'
UNITS = 'units'
VERTICAL = 'vertical'
W = 'w'
WORD = 'word'
WRITABLE = 4
X = 'x'
Y = 'y'
YES = 1
color_palette = {'active_button_color': 'thistle3', 'board': 'burlywood4', 'boardgame_background_color': 'misty rose', 'button_color':
'thistle', 'canvas_background': 'burlywood2', 'frame_background': 'thistle4', 'hole': 'wheat4', 'main_window_background_color':
'thistle2', 'random_colour': ['misty rose', 'tomato2', 'DarkOrchid3']}
holes = []
main_window_height = 500
main_window_width = 1000
player_one_score = 0
player_one_score_label = <tkinter.Label object .!label2>
player_two_score = 0
player_two_score_label = <tkinter.Label object .!label3>
turn = 'player_one'
turn_label = <tkinter.Label object .!label>
warning_unit = 0
```