

Лабараториска вежба 2

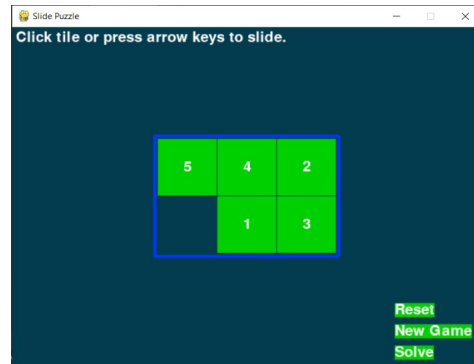
Slide puzzle

Трајче Ифтимов 173252

(забелешка: намалена е solutionSeq на 5 и полињата 3x2 заради побрзо тестирање)

1. Бројот на полиња во редиците и колоните да биде различен

```
# Create the constants (go ahead and experiment with different values)
#BOARDWIDTH = 4 # number of columns in the board
#BOARDHEIGHT = 4 # number of rows in the board
BOARDWIDTH = 3 # number of columns in the board
BOARDHEIGHT = 2 # number of rows in the board
```



2. Help menu

При секој клик на Help menu,

- се врши проверка помеѓу solutionSeq и allMoves доколку се исти првите елементи со рор се бришат во двете листи и проверката продолжува со следните елементи.
- Доколку се различни се бришат елементи само од allMoves.
- Доколку allMoves листата е празна се бриши елемент од solutionSeq

(Бришењето се однесува само на еден елемент при еден клик и се врши автоматски анимација)

```
elif HELP_RECT.collidepoint(event.pos):
    counthelp = counthelp + 1
    print(solutionSeq)
    print(allMoves)

    if ((len(solutionSeq) != 0) and (len(allMoves) == 0)):
        helpAnimation1(mainBoard, solutionSeq) # clicked on Help button
        solutionSeq.pop()
    flag = 1
    if len(allMoves) > 0:
        if flag == 1:
            for move in allMoves:
                if move == UP:
                    oppositeMove = DOWN
                elif move == DOWN:
                    oppositeMove = UP
                elif move == RIGHT:
                    oppositeMove = LEFT
                elif move == LEFT:
                    oppositeMove = RIGHT
                if oppositeMove == solutionSeq[-1]:
                    solutionSeq.pop()
                    allMoves.pop()
            else:
                flag = flag - 1

    if len(allMoves) != 0:
        helpAnimation(mainBoard, allMoves) # clicked on Help button
        allMoves.pop()
```

```
def helpAnimation(board, allMoves):
    revAllMoves = allMoves[:] # gets a copy of the list
    revAllMoves.reverse()

    for move in revAllMoves:
        if move == UP:
            oppositeMove = DOWN
        elif move == DOWN:
            oppositeMove = UP
        elif move == RIGHT:
            oppositeMove = LEFT
        elif move == LEFT:
            oppositeMove = RIGHT
        slideAnimation(board, oppositeMove, '', animationSpeed=int(TILESIZE / 4))
        makeMove(board, oppositeMove)
        break
```

```
def helpAnimation1(board, solutionSeq):
    revAllMoves1 = solutionSeq[:] # gets a copy of the list
    revAllMoves1.reverse()

    for move in revAllMoves1:
        if move == UP:
            oppositeMove = DOWN
        elif move == DOWN:
            oppositeMove = UP
        elif move == RIGHT:
            oppositeMove = LEFT
        elif move == LEFT:
            oppositeMove = RIGHT
        slideAnimation(board, oppositeMove, '', animationSpeed=int(TILESIZE / 4))
        makeMove(board, oppositeMove)
        break
```

3. Бројач на moves

