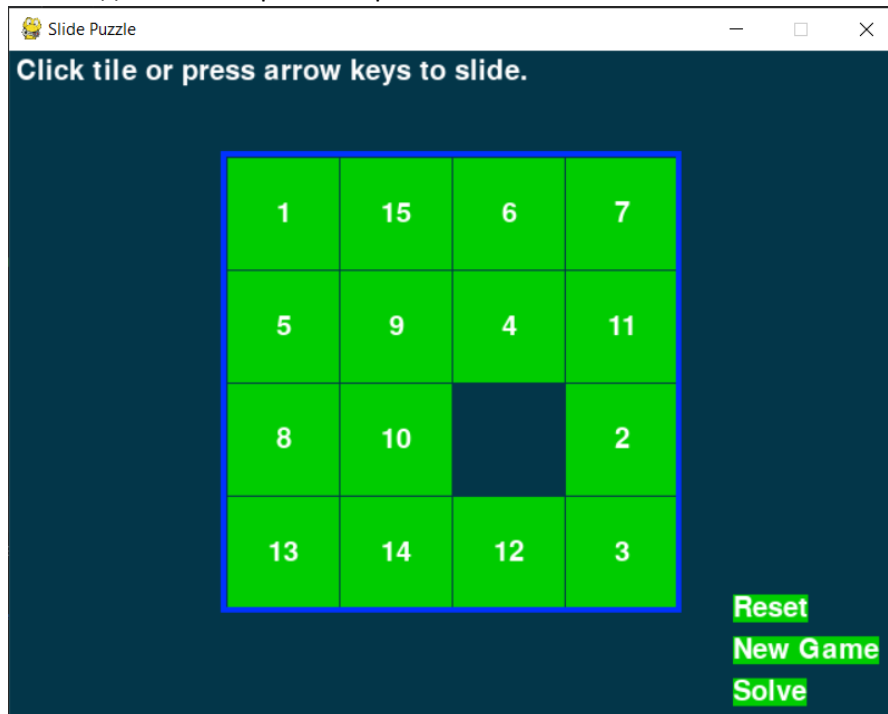
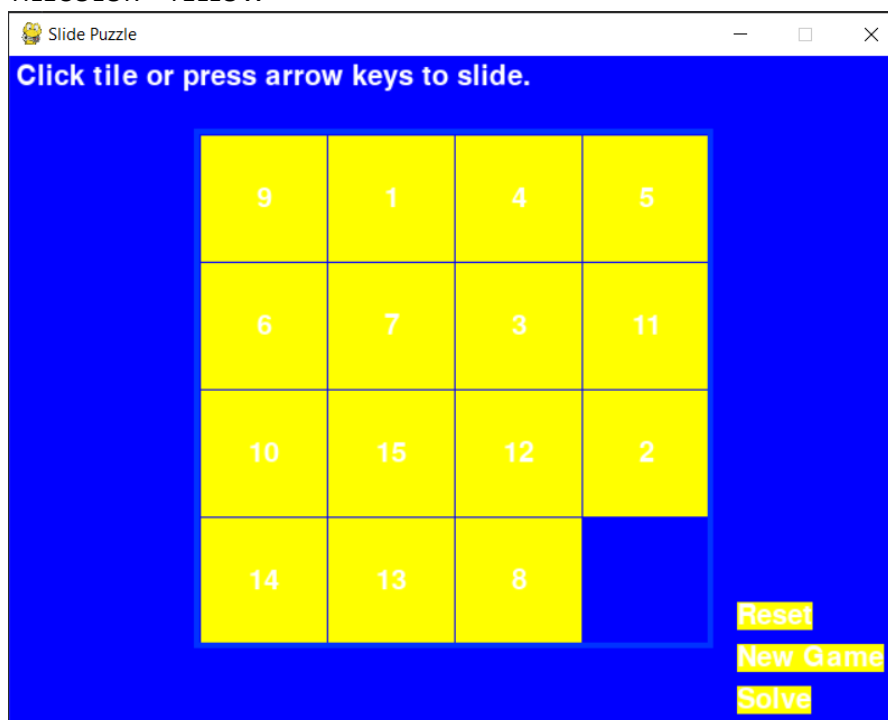


SlidePuzzle работа на час

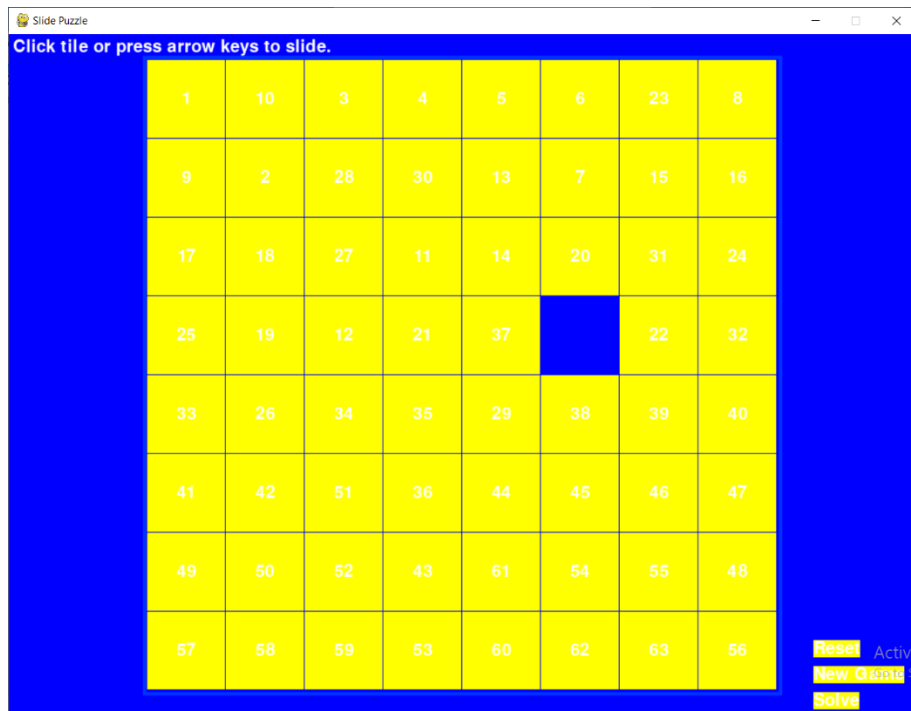
1. слика од почетен екран на играта.



2. Се згломува големината на полињата за 10 $TILESIZE = 80+10$, се додаваат нови бои $YELLOW = (255, 255, 0)$, $BLUE = (0, 0, 255)$ и се доделуваат на променливите кои ги чуваат вредностите за позадинска боја и боја на поле соодветно. $BGCOLOR = BLUE$, $TILECOLOR = YELLOW$



3. Се зголемува бројот на полиња со цел да се добие 8x8 табла тоа се постигнува со менување на вредностите на BOARDWIDTH = 8 и BOARDHEIGHT = 8. Со цел приказот да остане валиден односно во прозорецот да ги собере сите ново додадени полиња и притоа да се задржи нивната големина од претходното барање ги зголемуваме димензиите на прозорецот со: WINDOWWIDTH = 1050 и WINDOWHEIGHT = 790



4. Правиме промена во функцијата generateNewPuzzle така што numSlides добива дефолтна вредност None и Доколку numSlides == None , numSlides добива случајна вредност од 1 до 100. Дополнително во main и во main game loop повикот на функцијата го менуваме од generateNewPuzzle(80) во generateNewPuzzle() со овие промени се задоволува барањето при секоја нова игра бројот на случајни почетни движења да се избира случајно помеѓу 1 и 100

```
def generateNewPuzzle(numSlides = None):
```

```
    if numSlides == None:
```

```
        numSlides = random.randint(1,100)
```

5. При креирање на нова игра се генерира решена табла и потота последователно се извршува низа од случајни поместувања кои се зачувуваат во SolvedSeq а потоа секое движење тригерирано од играчот се зачувува во AllMoves

Па така при клик на копчето Solve се повикува функцијата resetAnimation со конкатенирана низа од SolvedSeq и AllMoves каде што се извршуваат сите движења но во обратен редослед и во обратна насока. Со ова гарантираме дека ќе се вратиме во почетната состојба (пред извршувањето на случајните движења) односно играта ќе се доведе до состојба на решена табла односно во состојба на победа што всушност е и цел на функционалонста на Solve button

Забелешка: Овој начин не е најдобро решение не ја решава играта во најмалку потези но е наједноставно и најлесно решение за имплементација кое секогаш гарантира дека играта ќе се доведе до состојба на победа

6. Наредбата `board[blankx][blanky], board[blankx][blanky + 1] = board[blankx][blanky + 1], board[blankx][blanky]` во пајтон се користи за замена на вредностите помеѓу две променливи за да се постигне истиот ефект во друг програмски јазик е потребна следната секвенца на наредби:

```
tmp = board[blankx][blanky]
board[blankx][blanky] = board[blankx][blanky + 1]
board[blankx][blanky + 1] = tmp
```

7. Овој ефект се добива кога пред да се примени анимацијата на лизгање не се пребришува полето кое што се движи(лизга). А тоа може да се добие со закоментирање на наредбата
`pygame.draw.rect(baseSurf, BG_COLOR, (moveLeft, moveTop, TILESIZE, TILESIZE))`
Која се наоѓа во функцијата `slideAnimation`
8. Прв начин: секаде каде што се повикува функцијата `slideAnimation` и каде сакаме да се забрза анимацијата го зголемуваме четвртиот параметар односно `animationSpeed` за онолку колку што сакаме да ја забрзаме анимацијата

Втор начин: Во функцијата `slideAnimation` го множиме параметарот `animationSpeed` со некој фактор на забрзување со ова брзината на анимација на лизгање се забрзува секаде

```
def slideAnimation(board, direction, message, animationSpeed):
    speed_up_factor = 1.25 # фактор за забрзување на анимацијата
    animationSpeed *= speed_up_factor
    ...
```

9. Дефинираме глобални променливи `blankxpos`, `blankypos` каде ќе се чува позицијата на празното поле

Дополнително правиме измени во:

- функцијата `generateNewPuzzle` така што
`blankxpos, blankypos = BOARDWIDTH-1, BOARDHEIGHT-1`
односно се поставува почетната позиција (пред извршувањето на случајните движења) на празното поле да е долниот десен кош
- Функцијата `getBlankPosition` само ќе ја враќа вредноста на променливите
`return blankxpos, blankypos`
- Функцијата `makeMove` така што во зависност од насоката на движењето ја ажурираме вредноста на `blankxpos` односно на `blankypos` променливата