|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| **Институт комплексной безопасности и специального приборостроения** | | |
|  |  |  |

Кафедра КБ-4 «Интеллектуальные системы инфомационной безопасности»

**Отчет**

по выполнению проектной работе

Игра «Tetris»

по дисциплине «Интерпретируемый язык программирования »

Выполнил:

Студент 3 курса

Группа ББСО-04-19

Шифр 19Б0325

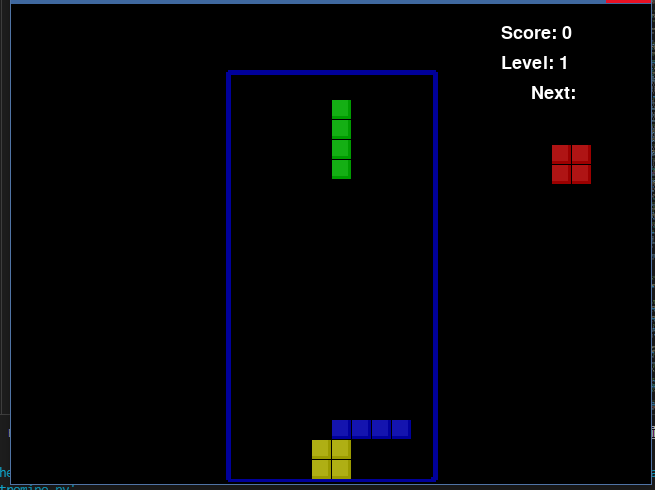
Кичкаев Никита Юрьевич

Проверил:

Тарланов А.Т.

Москва, 2021

Игра Tetris это платформер с рандомно генерируемым типом игры.



Суть игры заключается в том что необходимо перемещать падающие фигуры для заполнения игрового поля в случае если игровое поле заполнено и фигурам некуда встать то игра заканчивается.

.

Перемещение части в сторону

 if (event.key == K\_LEFT or event.key == K\_a) and isValidPosition(board, fallingPiece, adjX=-1):

                    fallingPiece['x'] -= 1

                    movingLeft = True

                    movingRight = False

                    lastMoveSidewaysTime = time.time()

                elif (event.key == K\_RIGHT or event.key == K\_d) and isValidPosition(board, fallingPiece, adjX=1):

                    fallingPiece['x'] += 1

                    movingRight = True

                    movingLeft = False

                    lastMoveSidewaysTime = time.time()

Вращение детали (если есть место для вращения)

  elif (event.key == K\_UP or event.key == K\_w):

                    fallingPiece['rotation'] = (fallingPiece['rotation'] + 1) % len(PIECES[fallingPiece['shape']])

                    if not isValidPosition(board, fallingPiece):

                        fallingPiece['rotation'] = (fallingPiece['rotation'] - 1) % len(PIECES[fallingPiece['shape']])

                elif (event.key == K\_q): # повернуть в другую сторону

                    fallingPiece['rotation'] = (fallingPiece['rotation'] - 1) % len(PIECES[fallingPiece['shape']])

                    if not isValidPosition(board, fallingPiece):

                        fallingPiece['rotation'] = (fallingPiece['rotation'] + 1) % len(PIECES[fallingPiece['shape']])

Переместить текущий кусок полностью вниз

 elif event.key == K\_SPACE:

                    movingDown = False

                    movingLeft = False

                    movingRight = False

                    for i in range(1, BOARDHEIGHT):

                        if not isValidPosition(board, fallingPiece, adjY=i):

                            break

                    fallingPiece['y'] += i - 1