

The background of the slide is a dark purple field filled with numerous triangles of various sizes and colors, including bright pink, orange, and light blue. These triangles are scattered across the entire surface, creating a vibrant, geometric pattern.

Tag. Программа игры «Пятнашки»

Проект выполнили студенты
группы ИП-811: Пимонова Юлия, Бобова Ирина

A decorative header at the top of the slide featuring a dark purple background with a pattern of small, light-colored triangles. Overlaid on this are larger, semi-transparent triangles in shades of teal, orange, and red.

Функционал проекта:

Курсовой проект представляет собой программу игры "Пятнашки", реализованную на языке С. Проверка осуществляется при помощи тестирования.

A decorative header at the top of the slide featuring a dark purple background with several colorful triangles (cyan, orange, red, and purple) scattered across it.

Краткий алгоритм работы:

Разрабатываемое приложение представляет собой программную реализацию известной игры «Пятнашки», цель которой заключается в упорядочивании игроком фишек путем их последовательного перемещения на соседнее пустое место, желательно сделав как можно меньше перемещений.

Прохождение теста:

```
mrskiren@mrskiren-VlvoBook-15-ASUS-Laptop-X540MA-D540MA:~/tag$ bin/main
```

```
  7  14   9   5
```

```
13   8  15  10
```

```
  2  12   3   6
```

```
11   4   1
```

```
control: w,a,s,d
```

```
quit: Esc
```

```
restart: r
```


Тестирование:

Проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранным определенным образом.

```
mrskiren@mrskiren-VivoBook-15-ASUS-Laptop-X540MA-D540MA:~/tag$ bin/main-test
TEST 1/6 checkWin:correct_win_combination1 [OK]
TEST 2/6 checkWin:random_correct_combination1 [OK]
TEST 3/6 checkWin:random_uncorrect_combination2 [OK]
TEST 4/6 checkWin:uncorrect_15_14_combination [OK]
TEST 5/6 checkWin:correct_win_combination2 [OK]
TEST 6/6 checkWin:uncorrect_win_combination [OK]
RESULTS: 6 tests (6 ok, 0 failed, 0 skipped) ran in 0 ms
```

Реализация алгоритмов:

1. Алгоритм генерации случайных чисел

```
void randomized(int board[][4]){
    while(1){
        int tmpBoard[4][4]={0};
        int test[16]={0},k=0,sum=0;
        int checksum[16]={0};
        for(int i=1;i<=15;i++){
            while(1){
                int column = rand()%4;
                int row = rand()%4;
                if(tmpBoard[column][row]==0&&!(column==3&&row==3)){
                    test[column*4+1+row]=i;
                    k++;
                    for(int j=k;j>=0;j--){
                        if(test[j]<i)
                            checksum[j]++;
                    }
                    tmpBoard[column][row]=i;
                    break;
                }
            }
        }
        for(int i=1;i<16;i++){
            for(int j=15;j>i;j--){
                if(test[i]>test[j])
                    sum++;
            }
        }
        if(sum%2==0){
            for(int i=0;i < 4;i++){
                for(int j=0;j<4;j++){
                    board[i][j]=tmpBoard[i][j];
                }
            }
            break;
        }
    }
}
```

Реализация алгоритмов:

2. Алгоритм генерации поля

```
void printBoard(int board[][4])
{
    for(int i=0;i < 4;i++)
    {
        for(int j=0;j<4;j++)
        {
            if(board[i][j]==0)
                printf(" ");
            else
                printf("%4d",board[i][j]);
        }
        printf("\n\n");
    }
    printf("\ncontrol: w,a,s,d\nquit: Esc\nrestart: r\n");
}
```

Реализация алгоритмов:

3. Алгоритм выигрышной комбинации

```
int checkWin(int board[][4])
{
    int count = 1;
    for(int i=0; i<4; i++)
    {
        for(int j=0; j<4; j++)
        {
            if(count > 15) return true;
            if(board[i][j] == count)
            {
                count++;
            }
            if(board[i][j] != count)
                return false;
        }
    }
    return 0;
}
```


Общий вид истории коммитов:

```
* 5c2d3d1 (HEAD -> master, origin/master, origin/HEAD) Merge branch fixbranch
| \
| * d78753d (fixbranch) Fix
| /
* 4173594 Merge branch tests
| \
| * d8142fa Add tests
| * a07e37b Change file fot travis
| * f3b01af Change Makefile for tests
| * 5f29edd Add main.c for tests
| /
* a455752 Merge branch thirdparty
| \
| * 1808f25 Add clang-format
| * 3eca4b8 Add ctest.h
| /
* b118d94 Merge branch print
| \
| * a9011ed Change Makefile
| * 27ede05 Add print.h
| * 8025328 Add print
| /
* 592bdda Merge branch random
| \
| * a3073b7 Change Makefile
| * 04222a2 Add random.h
| * d4c7443 Add random
| /
* c91a3bd Merge branch CheckWin
| \
| * e84bf5d Add CheckWin file
| /
* 75c6182 Merge branch 'travis'
| \
| * 549d26c Add travis
| /
* c91cd9a Merge branch stick
```