**Министерство образования Приморского края**

**КРАЕВОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ПРИМОРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

**(КГА ПОУ «ППК»)**

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

|  |  |
| --- | --- |
| **Выполнил**  студент группы 1125 | **Руководитель**  преподаватель |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Д. Проскурин | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Пушкин |

**Владивосток 2021**

**Содержание**

[**1 Разработка приложения** 3](#_Toc75097866)

[**1.1 Игровая логика приложения** 3](#_Toc75097867)

[**1.2 Разработка исходного кода приложения** 4](#_Toc75097868)

[**1.3 Описание интерфейса приложения** 6](#_Toc75097869)

# **1 Разработка приложения**

# **1.1 Игровая логика приложения**

Суть игры в тетрис состоит в том, что случайные фигурки (каждая из которых состоит строго из 4-х сегментов) падают сверху вниз на поле. Игрок во время полета каждой фигурки может поворачивать ее вокруг своей оси и двигать влево-вправо по горизонтали, выбирая место, куда она должна упасть. Когда по горизонтали заполняется строка – она исчезает. Очки начисляются за каждую исчезнувшую строку. Главное, чтобы фигуры не дошли до верхней границы поля, иначе игра закончена.

С помощью клавиш «🡪», «🡨», «🡩», «🡫» осуществляется перемещение фигуры вправо, влево, по своей оси меняет положение, ускоряется вниз – соответственно.

С помощью клавиши «F3» игра приостанавливается на паузу, а при помощи клавиши «F2» игра начинается сначала.

При запуске приложения игра сразу начинается и будет продолжаться пока пользователь не поставит на паузу, не начнёт заново или не выйдет.

# **1.2 Разработка исходного кода приложения**

В MainWindow.xaml.cs созданы переменные в public class Tetramino созданы переменные private Point currPosition; private Point[] currShape; private Brush currColor; private bool rotate.

Public void movLeft, movRight, movDown, movRotate – команды для перемещения объектов в игре с помощью координат currPosition.X -= 1, +=1, currPosition.Y -= 1 соответственно; с Rotate другое: double x = currShape[i].X; currShape[i].X = currShape[i].Y \* -1; currShape[i].Y = x – ввиду чего можем поворачивать объект.

Private Point[] setRandomShape() – здесь созданы фигуры по координатам и для них выбран цвет, например:

1. с помощью switch создаём case 2: rotate = true;
2. создаём цвет объекта - currColor = Brushes.Orange;
3. return new Point[] { new Point(0,0), new Point(-1,0), new Point(1,0), new Point(1,-1) – создаём фигуру объекта, грубо говоря – рисуем по точкам;

ввиду чего у нас получится фигура в виде L оранжевого цвета. Аналогично добавлены и другие объекты других цветов и фигур.

Так же здесь есть экземпляр рандом Random rand = new Random(); с помощью которого (rand.Next() % 7) фигуры будут попадаться в случайном порядке.

private void CheckRows() – в ней расписаны свойства, которые нужны для того, чтобы приложение могло понимать, что получились строки или строка из объектов и засчитывать очки игроку.

private void RemoveRow(int row) – в ней свойство, которое служит для того, чтобы эта полная заполненная строка удалялась.

1. public Brush getCurrColor() { return currColor; } – определяет цвет,
2. public Point getCurrPosition() { return currPosition; } – определяет позицию,
3. public Point[] getCurrShape() { return currShape; } – определяет форму.

public void CurrTetraminoMonLeft() - переменная для определения формы и позиции в зависимости от фигуры координат каждой части объекта при перемещении объекта влево.

public void CurrTetraminoMonRight() - переменная для определения формы и позиции в зависимости от фигуры координат каждой части объекта при перемещении объекта вправо.

public void CurrTetraminoMonDown() - переменная для определения формы и позиции в зависимости от фигуры координат каждой части объекта при перемещении объекта вниз.

public void CurrTetraminoMonRotate() - переменная для определения формы и позиции в зависимости от фигуры координат каждой части объекта при перемещении объекта по своей оси.

private void HandleKeyDown(object sender, KeyEventArgs e) – привязываем клавишы к командам, такие как «🡪» (вправо), «🡨» (влево), «🡩» (менять положение по своей оси), «🡫» (вниз), «F3» (пауза), «F2» (старт новой игры).

void MainWindow\_Initilized(object sender, EventArgs e) – создан таймер, который запускается по началу игры. Грубо говоря – здесь задаётся скорость падения объектов, следовательно, после чего появляется новый объект(фигура).

private void GameStart() – команда старта приложения(игры), после чего запускается и таймер игры (создаются объекты).

private void GamePause() – команда паузы приложения(игры), после чего игра останавливается и таймер игры (не создаются объекты).

public Board(Grid TetrisGrid) – создаётся поле для игры.

private void currTetraminoDraw() – рисует объект.

private void currTetraminoErase() – удаляет объект.

private void CheckRows() – происходит проверка строк.

void GameTick(object sender, EventArgs e) – нужна для того, чтобы показывать счёт в главном окне.

Главном окно MainWindow.xaml с высотой 570 и шириной 525 поделено на два поля:

1. Первое поле – это StackPanel с шириной 172.Оно нужно для выведения счёта пользователя тетриса. В параметре Label выводятся числа, которые были описаны в коде MainWindow.xaml.cs.
2. Второе поле – это Grid («MainGrid») с высотой 500 и шириной 250. На этом поле и происходит главное событие игры – на этом поле появляются объекты в виде фигур и падают вниз. Поле Grid поделено на много клеток с помощью Grid.RowDefinitions, в котором несколько раз поле разбивается на разметку с помощью RowDefinition по вертикали. Чем больше используется RowDefinition, тем меньше будет клетка по размеру и высоте и больше клеток будет. Так же поле Grid поделено на много клеток с помощью Grid.ColumnDefinitions, в котором несколько раз поле разбивается на разметку с помощью ColumnDefinition по горизонтали. Чем больше используется ColumnDefinition, тем меньше будет клетка по ширине и по размеру и больше клеток будет.В совокупности Grid.RowDefinitions и Grid.ColumnDefinitions вместе рисуют клетки. Это необходимо для данного кода приложения, потому что по этим клеткам и создаются объекты в коде MainWindow.xaml.cs.

# **1.3 Описание интерфейса приложения**

При запуске приложения появляется окно с серым полем, которое поделено на много клеток, квадратиков, и игра тут же начинается. Можно поставить игру на паузу, нажав на клавишу «F3» или начать сначала, нажав на клавишу «F2». По этим клеткам и будут перемещаться различные объекты в виде фигур сверху вниз. Объекты имеют разные цвета, разные фигуры. При заполнении горизонтальной строки – она исчезает, при этом очки начисляются игроку (за каждую исчезнувшую строку) и очки выводятся в верхнем правом углу, где написано «Score».

При нажатии на клавиши «🡪», «🡨», «🡩», «🡫» осуществляется перемещение фигуры вправо, влево, по своей оси меняет положение, ускоряется вниз – соответственно. Фигуры будут спускаться определённой скоростью, и новая фигура появится только тогда, когда предыдущая фигура будет поставлена на поле или на прошлую фигуру.