

Разработка игры камень, ножницы, бумага на C#

Ульянов Егор Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается и описывается разработка игры камень, ножницы, бумага на языке программирования C#. Практическим результатом является разработанная игра.

Ключевые слова: Игра камень, ножницы, бумага, C#, Visual Studio

Development of the game rock, paper, scissors in C#

Ulianov Egor Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses and describes the development of a game of rock, scissors, paper on the C # programming language. The bottom line is a developed game.

Keywords: Rock, Paper, Scissors game, C#, Visual Studio

Камень, ножницы, бумага — популярная игра на руках, известная во многих странах мира. Часто используется как способ жеребьёвки для определения очерёдности хода в других играх.

Игра популярна в силу своей простоты, непредсказуемости и тем, что не требует специальных приспособлений и реквизитов. Вся игра проходит с помощью рук людей. В игре могут участвовать два и более игрока. Игроки одновременно сжимают в кулак одну из своих рук, трясут кулаками 2-3 секунды, произнося «раз-два-три» или «цу-е-фа», затем разжимают кулаки и показывают друг другу с помощью пальцев руки один из трех знаков — камень, ножницы или бумагу. Камень — сжатый кулак, ножницы — распрямленные указательный и средний пальцы руки, бумага — раскрытая ладонь.

Победитель определяется по следующим правилам:

- Бумага побеждает камень («бумага обёртывает камень»).
- Камень побеждает ножницы («камень затупляет или ломает ножницы»).
- Ножницы побеждают бумагу («ножницы разрезают бумагу»).

Целью данной статьи является создание игры камень, ножницы, бумага в среде разработки «Visual Studio» на языке программирования C#.

В своей работе Н. Н. Додобоев, О. И. Кукарцева, Я. А. Тынченко рассмотрели вопросы появления различных языков программирования (в частности C#), определения особенностей этих языков, а также составления основных видов и классификаций языков программирования[1]. З. С. Магомадова рассмотрела языки программирования высокого уровня, особенности, недостатки и сложности в изучении, а также описала несколько легких алгоритмов[2]. В своей работе И. Ю. Просвирнина создала приложение «Морской бой», обладающее игровым искусственным интеллектом, в котором предусмотрен режим игры «Игрок против компьютера» [3]. В статье Н. А. Базеевой и Д. С. Лебедева описано начало игровой индустрии, а также рассмотрены особенности языков программирования для разработки игр [4].

Создаем проект «Windows Forms App» и называем его. Далее создаем четыре кнопки, и в поле «Text» называем бедующие камень, ножницы и бумагу. Также в поле «Name» необходимо изменить стандартное значение на «btnRock», «btnPaper», «btnScissors» и «btnRestart» соответственно см. рисунок 1.

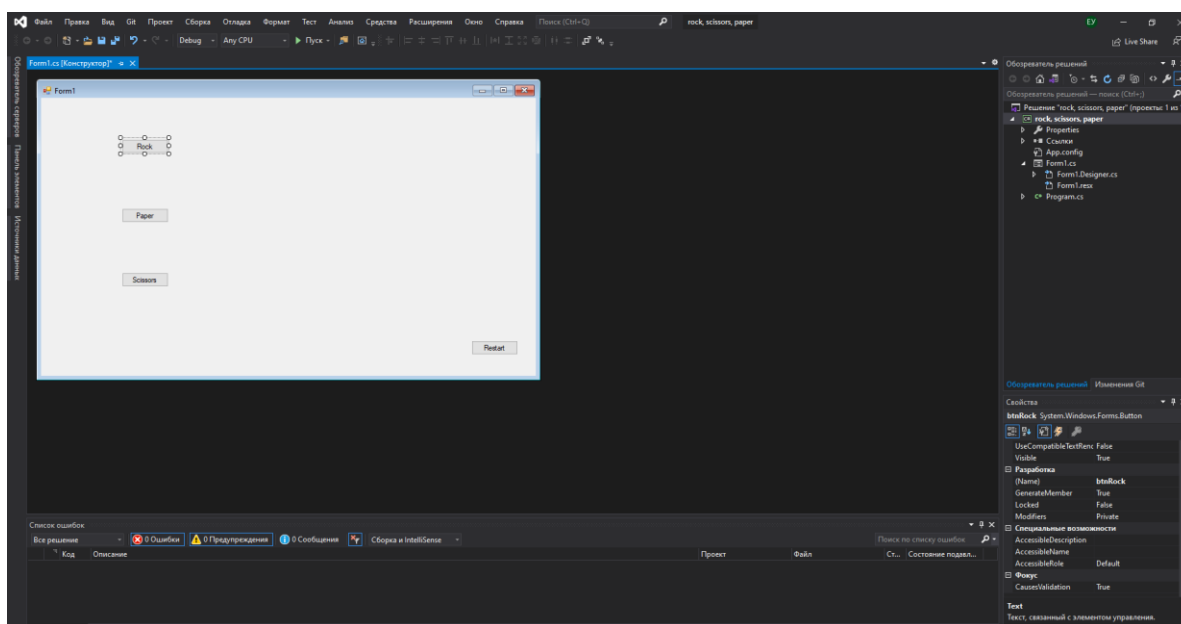


Рис. 1. Первоначальная настройка и создание кнопок

Далее добавляем два «PictureBox», и в поле «Name» вводим слева «picPlayer» и справа «picCpu». Также понадобится несколько маркеров, которые будут указывать, где находится настоящий игрок, где компьютер, таймер, счет и раунд см. рисунок 2.

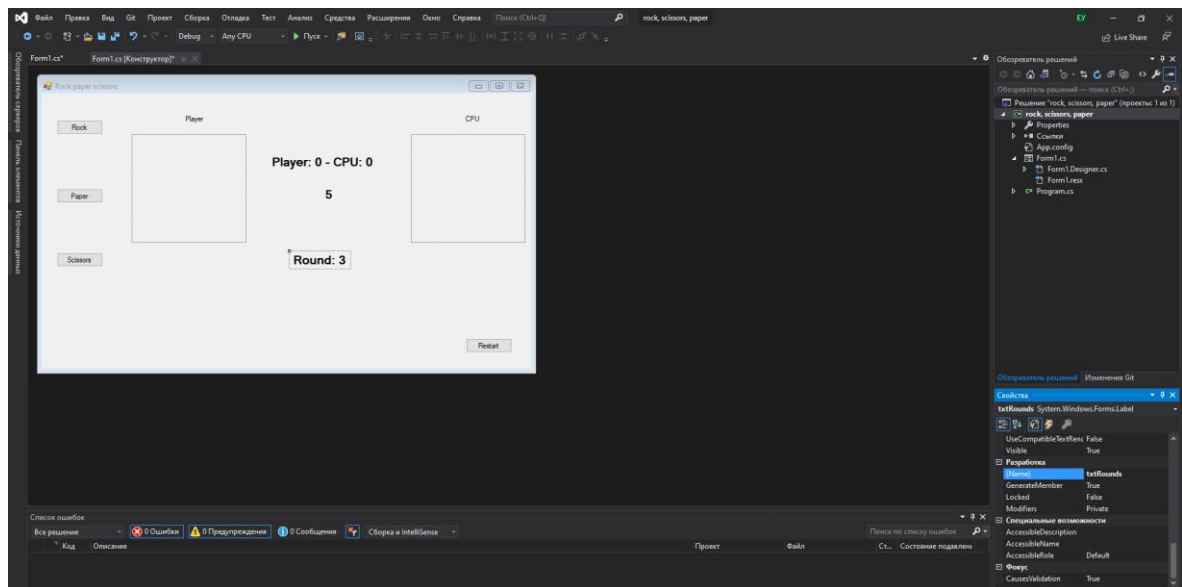


Рис. 2. Первоначальный интерфейс

Импортируем изображения в проект, знак вопроса ставим на стандарт. Создаем таймер, меняем интервал на 1000. Далее начинаем писать код. Создаем нужные переменные и генератор выбора действия «CPU» см. рисунок 3.

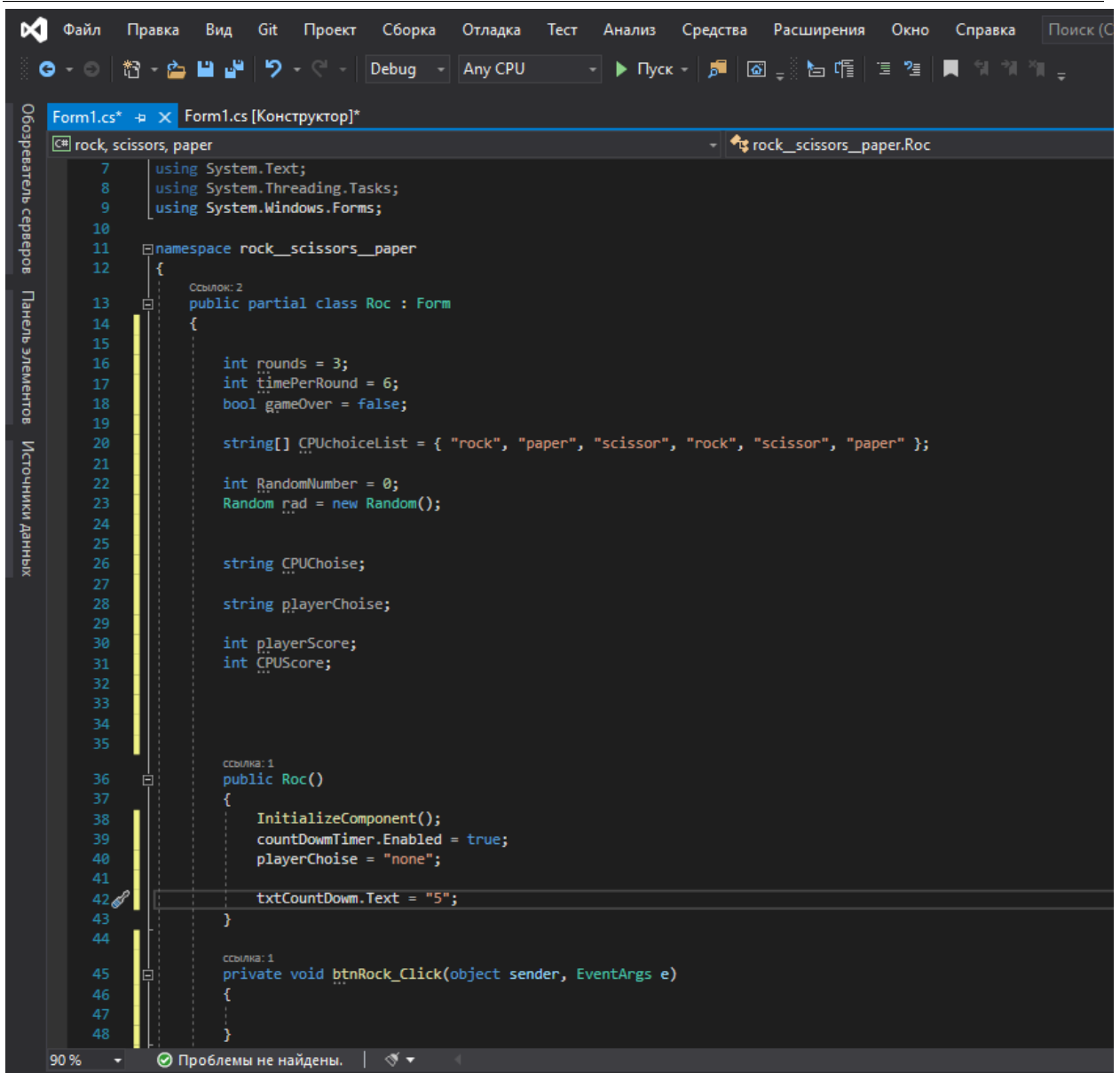
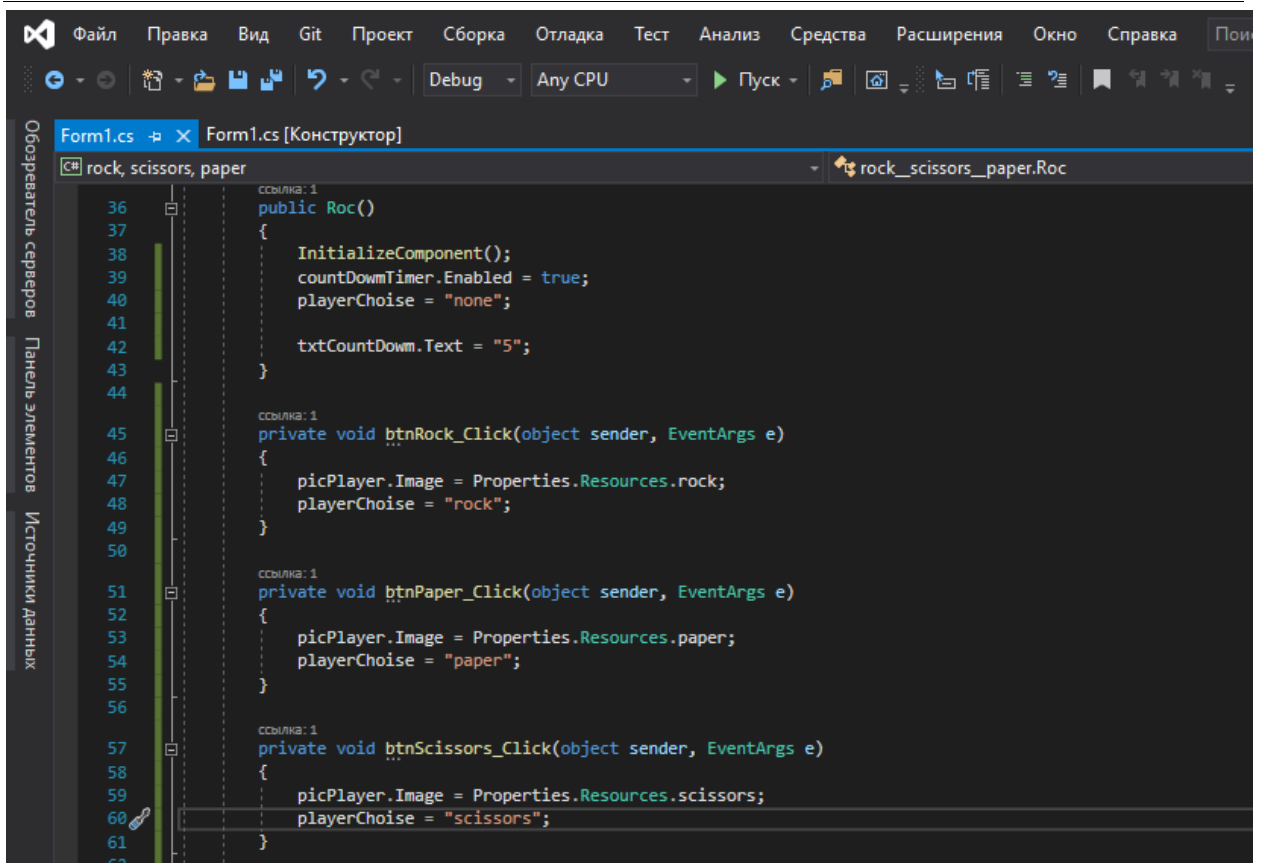


Рис. 3. Создание переменных

Далее необходимо запрограммировать выбор игрока. Для этого напишем триггер нажатия см. рисунок 4.

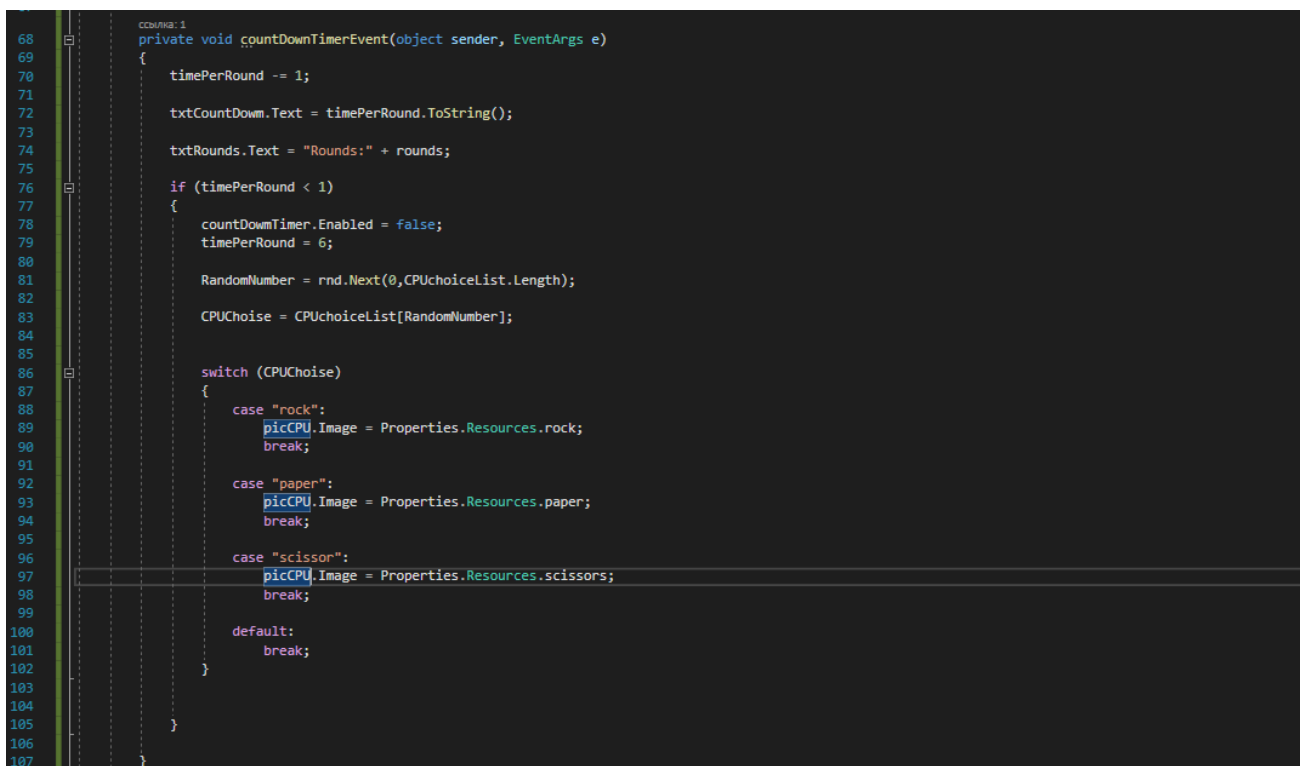


```
Form1.cs [Конструктор]
rock_scissors_paper
rock_scissors_paper.Roc

36  ссылка: 1
37  public Roc()
38  {
39      InitializeComponent();
40      countdownTimer.Enabled = true;
41      playerChoice = "none";
42
43      txtCountDown.Text = "5";
44  }
45
46  ссылка: 1
47  private void btnRock_Click(object sender, EventArgs e)
48  {
49      picPlayer.Image = Properties.Resources.rock;
50      playerChoice = "rock";
51  }
52
53  ссылка: 1
54  private void btnPaper_Click(object sender, EventArgs e)
55  {
56      picPlayer.Image = Properties.Resources.paper;
57      playerChoice = "paper";
58  }
59
60  ссылка: 1
61  private void btnScissors_Click(object sender, EventArgs e)
62  {
63      picPlayer.Image = Properties.Resources.scissors;
64      playerChoice = "scissors";
65  }
66  }
```

Рис. 4. Программирование выбора игрока

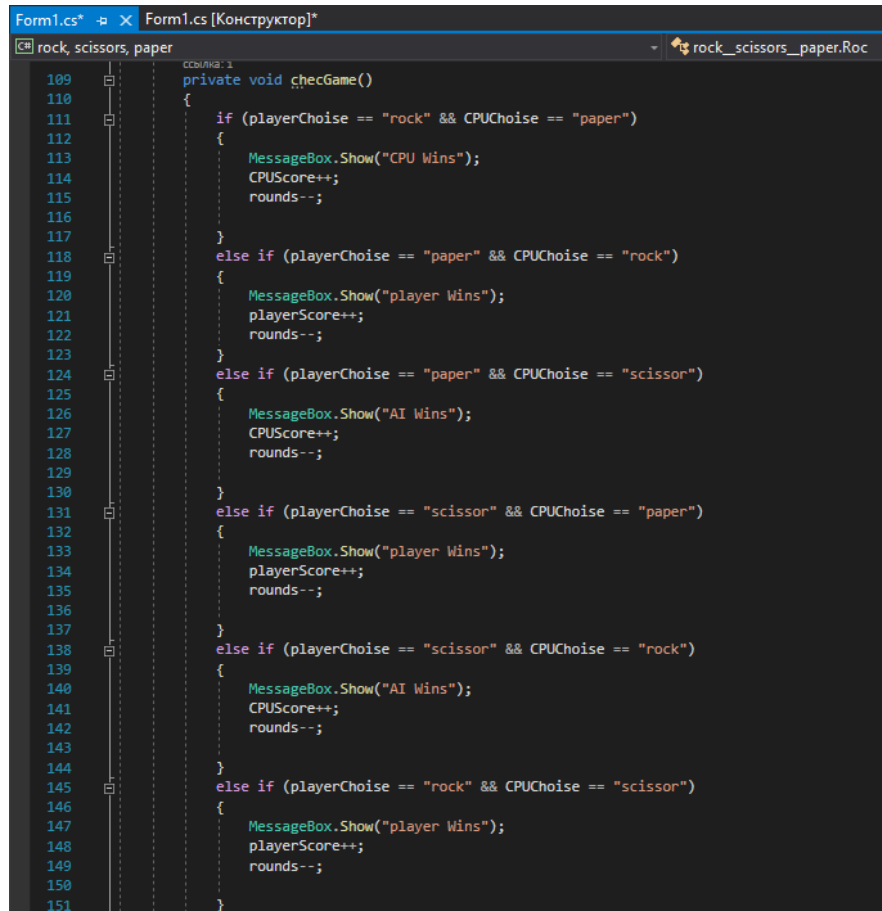
Теперь нужно запрограммировать выбор компьютера см. рисунок 5.



```
68  ссылка: 1
69  private void countdownTimerEvent(object sender, EventArgs e)
70  {
71      timePerRound -= 1;
72      txtCountDown.Text = timePerRound.ToString();
73      txtRounds.Text = "Rounds:" + rounds;
74
75      if (timePerRound < 1)
76      {
77          countdownTimer.Enabled = false;
78          timePerRound = 6;
79
80          RandomNumber = rnd.Next(0, CPUchoiceList.Length);
81          CPUchoise = CPUchoiceList[RandomNumber];
82
83          switch (CPUchoise)
84          {
85              case "rock":
86                  picCPU.Image = Properties.Resources.rock;
87                  break;
88              case "paper":
89                  picCPU.Image = Properties.Resources.paper;
90                  break;
91              case "scissor":
92                  picCPU.Image = Properties.Resources.scissors;
93                  break;
94              default:
95                  break;
96          }
97      }
98  }
99
100
101
102
103
104
105
106
107  }
```

Рис. 5. Программирование выбора компьютера

Далее через конструкцию «if-else» програмируем правила игры см. рисунок 6 и 7.



```
Form1.cs*  Form1.cs [Конструктор]*
rock, scissors, paper  rock_scissors_paper.Roc
109 private void checkGame()
110 {
111     if (playerChoise == "rock" && CPUchoise == "paper")
112     {
113         MessageBox.Show("CPU Wins");
114         CPUScore++;
115         rounds--;
116     }
117
118     else if (playerChoise == "paper" && CPUchoise == "rock")
119     {
120         MessageBox.Show("player Wins");
121         playerScore++;
122         rounds--;
123     }
124
125     else if (playerChoise == "paper" && CPUchoise == "scissor")
126     {
127         MessageBox.Show("AI Wins");
128         CPUScore++;
129         rounds--;
130     }
131
132     else if (playerChoise == "scissor" && CPUchoise == "paper")
133     {
134         MessageBox.Show("player Wins");
135         playerScore++;
136         rounds--;
137     }
138
139     else if (playerChoise == "scissor" && CPUchoise == "rock")
140     {
141         MessageBox.Show("AI Wins");
142         CPUScore++;
143         rounds--;
144     }
145
146     else if (playerChoise == "rock" && CPUchoise == "scissor")
147     {
148         MessageBox.Show("player Wins");
149         playerScore++;
150         rounds--;
151     }
152 }
```

Рис. 6. Конструкция if-else



```
152     else if (playerChoise == "none")
153     {
154         MessageBox.Show(label1.Text + " Make a choice");
155     }
156
157     else
158     {
159         MessageBox.Show("Draw");
160     }
161
162 }
163
164
165
166
167 }
```

Рис. 7. Продолжение кода

Сейчас в игре можно сыграть только один раунд, что бы это исправить, нужно с помощью той же конструкции проверить закончился ли предыдущий раунд см. рисунок 8.

```
    }
    startNextRound();
}

ССЫЛКА: 1
private void startNextRound()
{
    if (gameOver == true)
    {
        return;
    }
    txtScore.Text = "Player: " + playerScore + " - " + "CPU: " + CPUScore;

    playerChoise = "none";

    countdownTimer.Enabled = true;

    picPlayer.Image = Properties.Resources.qq;
    picCPU.Image = Properties.Resources.qq;
}
}
```

Рис. 8. Проверка завершения предыдущего раунда

Осталось только активизировать кнопку «Restart». Для этого вернем в начальное положение: счет, таймер и раунд см. рисунок 9.

```
ССЫЛКА: 1
private void btnRestart_Click(object sender, EventArgs e)
{
    playerScore = 0;
    CPUScore = 0;
    rounds = 3;
    txtScore.Text = "Player: " + playerScore + " - " + "CPU: " + CPUScore;

    playerChoise = "none";

    countdownTimer.Enabled = true;

    picPlayer.Image = Properties.Resources.qq;
    picCPU.Image = Properties.Resources.qq;
    gameOver = false;
}
}
```

Рис. 9. Программирование функции рестарта

Теперь игра готова и на рисунках ниже представлена ее реализация. Нажимаем кнопку «Play» в проекте и делаем выбор см. рисунок 10-14.

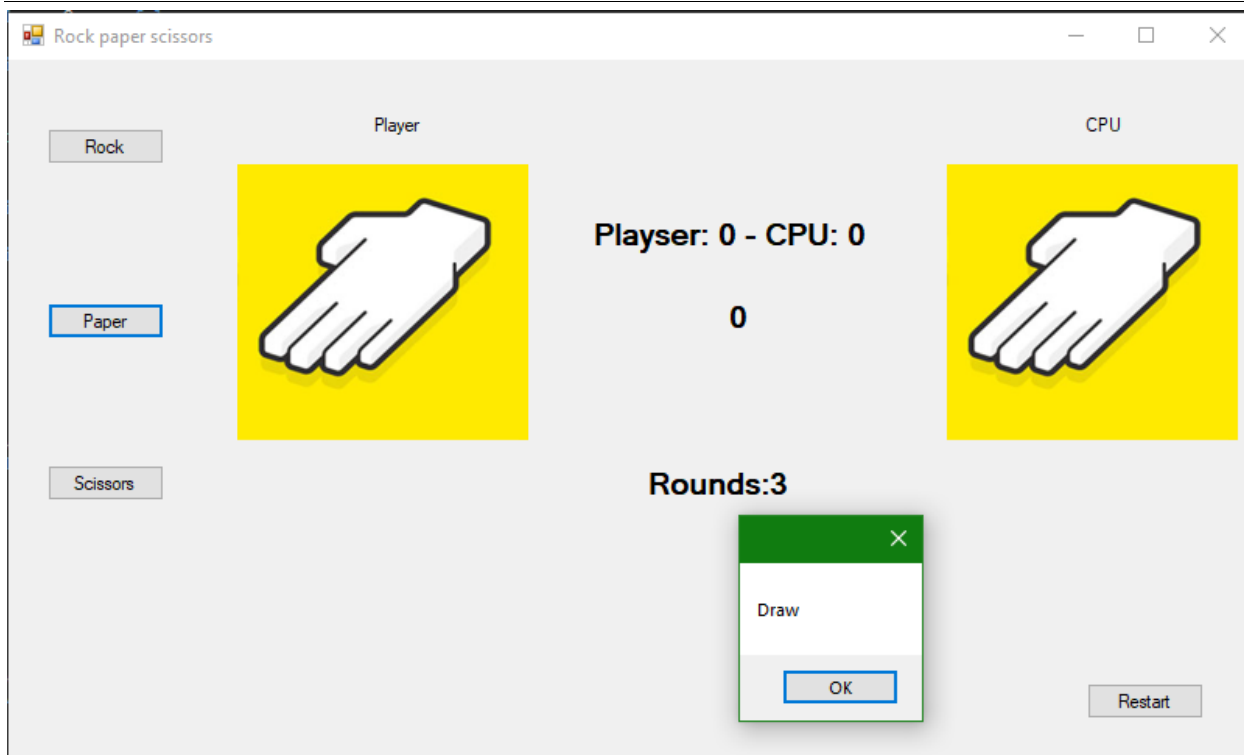


Рис. 10. Ничья

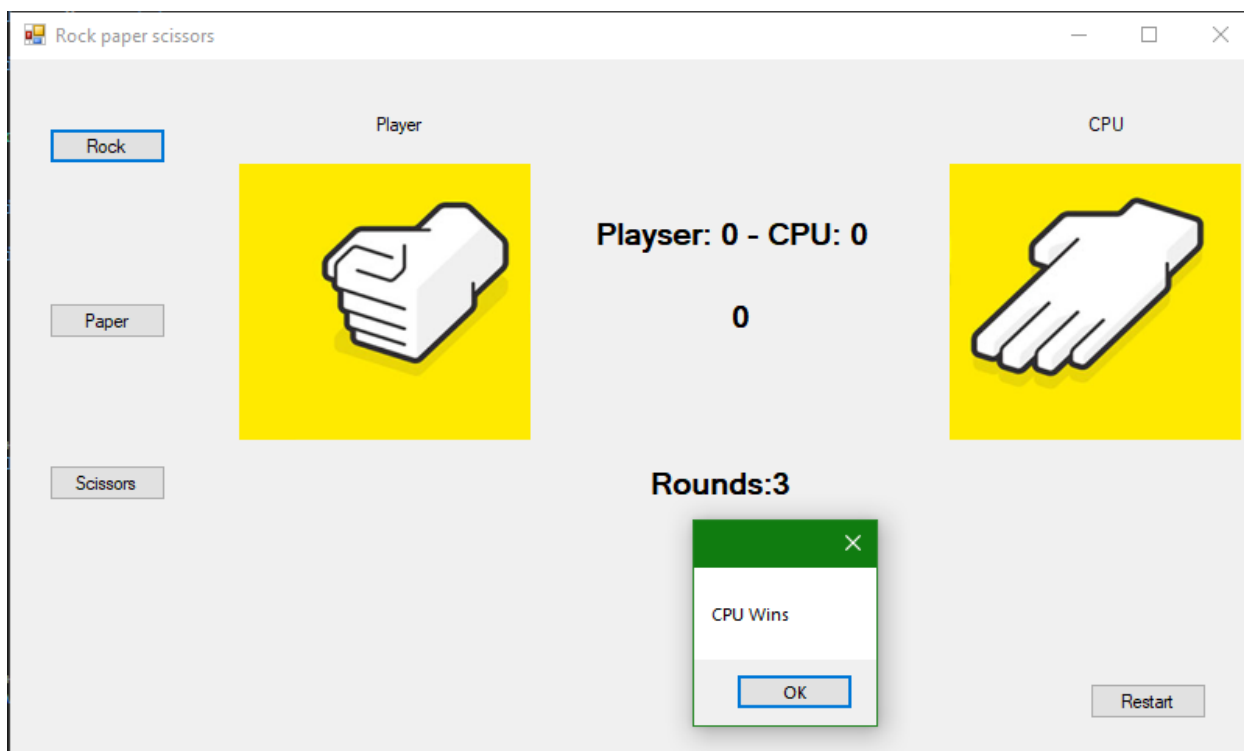


Рис. 11. Победа компьютера в раунде

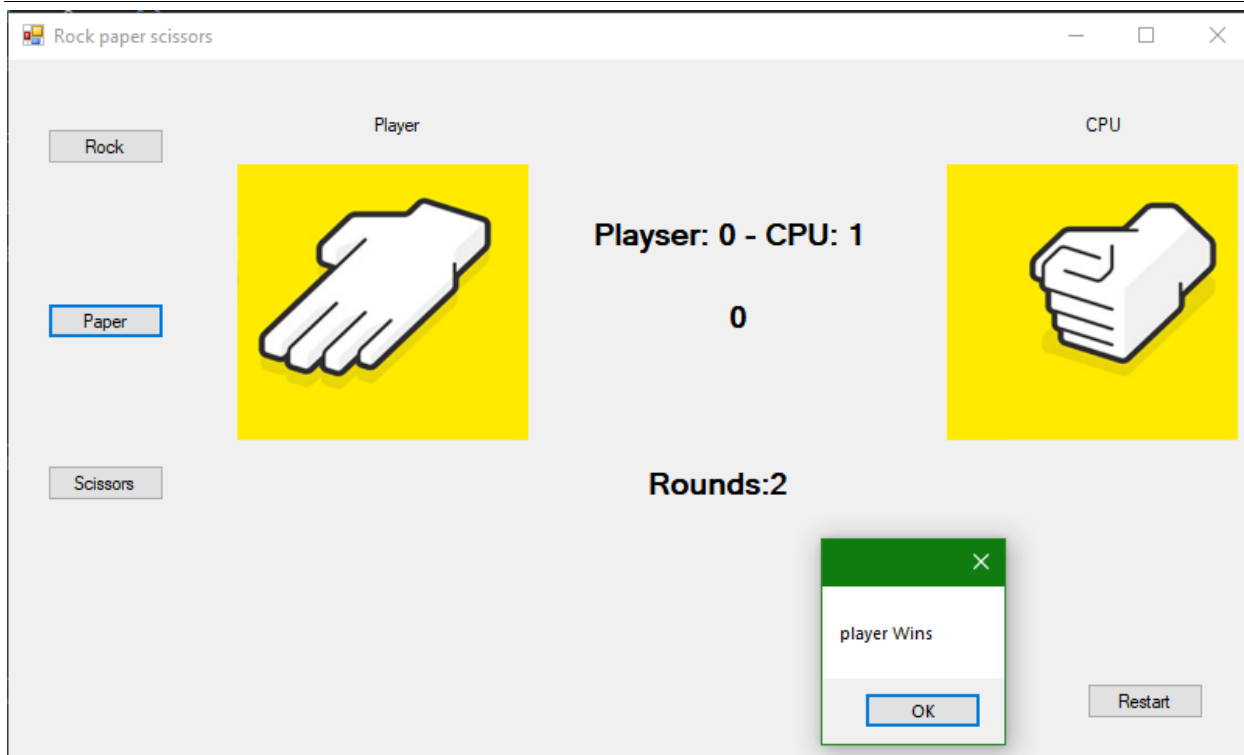


Рис. 12. Победа пользователя в раунде

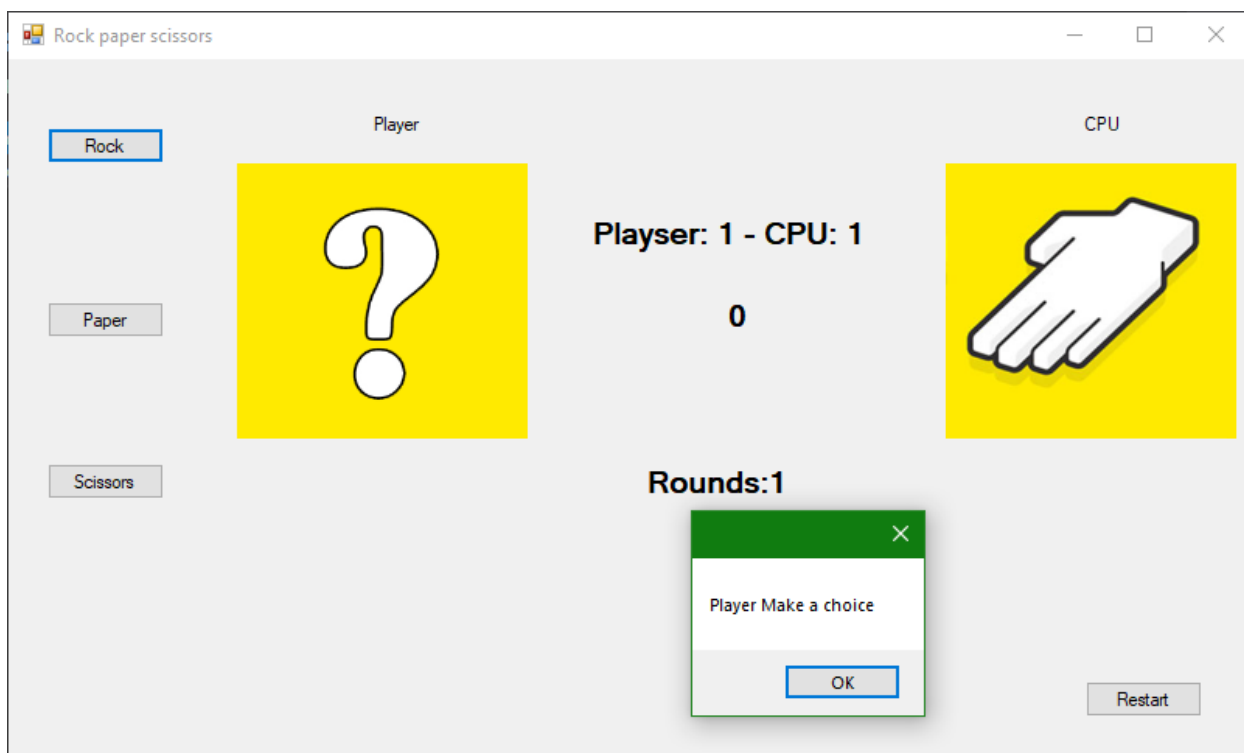


Рис. 13. Пользователь не сделал выбор

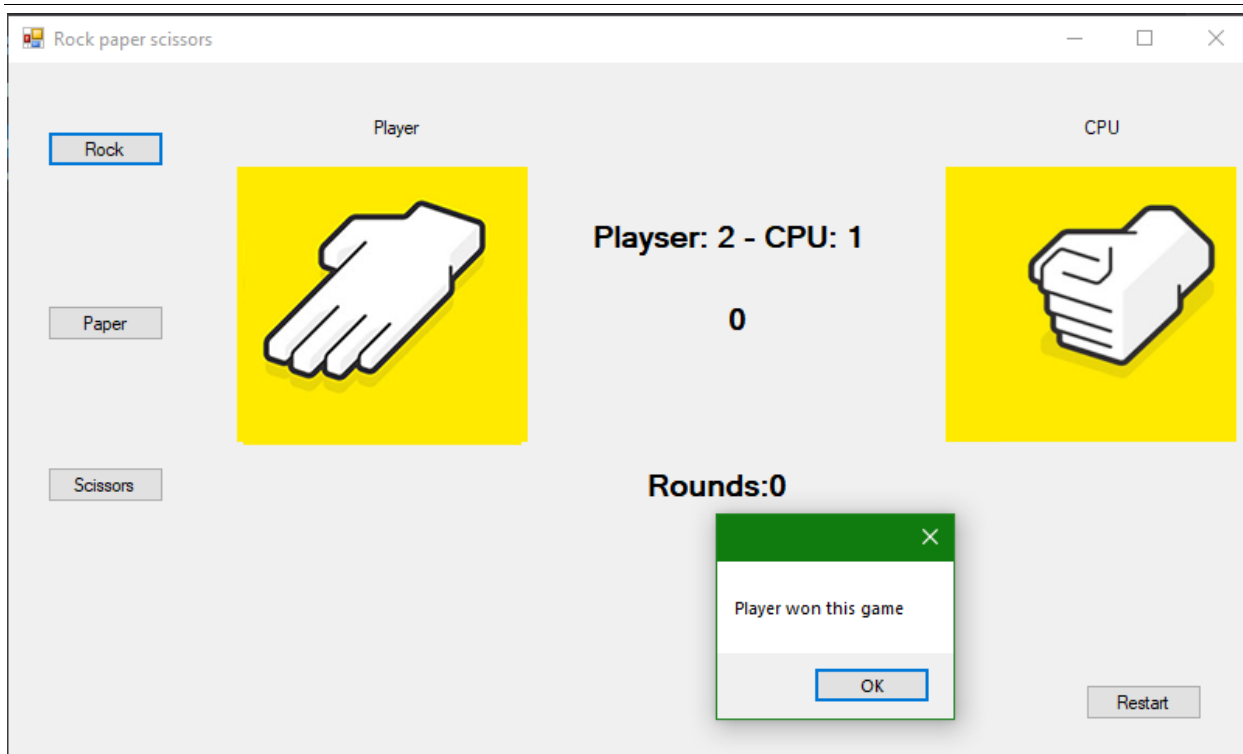


Рис. 14. Победа пользователя в игре

Таким образом, была написана игра камень, ножницы, бумага в среде разработки «Visual Studio» на языке программирования C#.

Библиографический список

1. Додобоев Н. Н., Кукарцева О. И., Тынченко Я. А. Современные языки программирования // Современные технологии: актуальные вопросы, достижения и инновации. 2014. №5. С. 81-85.
2. Магомадова З. С. Языки программирования высокого уровня // Разработка и применение наукоёмких технологий в эпоху глобальных трансформаций. 2020. №8. С. 94-96.
3. Просвирнина И. Ю., Егунова А. И., Аббакумов А. А. Среда разработки Microsoft Visual Studio на примере создания игры "морской бой" // Интеграционные процессы в науке в современных условиях. 2017. С. 123-125.
4. Базеева Н. А., Лебедев Д. С. Языки программирования для создания игр // E-Scio. 2019. №4. С. 31-39.