ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ

“КАМЕНСК-УРАЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ”

|  |  | Специальность: “Информационные системы и программирование”.  Группа: ИСиП-201 |
| --- | --- | --- |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

по дисциплине: “Основы алгоритмизации и программирования”

Тема: “Творческий проект. Сборник игр.”

|  |  | Выполнила:  Устьянцева М.Е.  Проверила:  Дмитриева Я.Л. |
| --- | --- | --- |

г.Каменск-Уральский, 2022 г.

ОГЛАВЛЕНИЕ:

Добавьте заголовки ("Формат > Стили абзацев"), чтобы они появились в оглавлении.

# ВВЕДЕНИЕ

На данный момент одним из самых популярных и актуальных способов развлечения являются игры. Они берут свое начало еще задолго до появления современных гаджетов, и по мере развития самого человека, в мире рождались всё более интересные и узко направленные игры. Таким образом они стали классифицироваться, и со временем начали разделяться на такие группы как: физические и психологические, интеллектуально-творческие и социальные. Особое внимание стоит уделить интеллектуально-творческому виду, который внес немаловажный вклад в развитие познавательных процессов ребёнка и умственному развитию личности. Самыми распространенными, известными и легко узнаваемыми играми этого типа стали пятнашки, “memory” (или же игра на память, которая особенно популярна за рубежом), судоку и ребусы. Изначально, по игровому действию все они разделялись на настольные и словесные, но, идя нога в ногу со временем, технологиями и совершенствованиями, люди научились реализовывать игры на компьютере, применяя абсолютно разные способы и методы решения, дизайн, условия и формы, и при этом не меняя сути и предназначения самой игры.

Так, целью моего проекта стало создание собственного уникального сборника, состоящего из трех интеллектуально-творческих игр “Найди пару”, “Пятнашки” и “Квадраты внимания”, которые проверят игрока на уровень его логики, реакции, памяти и внимания, и, возможно, поспособствуют их развитию.

# АНАЛИТИЧЕСКИЙ ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

На создание моего творческого проекта понадобилось больше месяца. За это время я посмотрела некоторое количество обучающих роликов на YouTube, оценила и попробовала создать некоторые части моей работы, опираясь на полученную информацию из видео, изучила и посетила множество сайтов и форумов с вопросами, которые так или иначе касались способов реализации моей задумки. Наиболее часто посещаемыми мной сайтами стали metanit.com и cyberforum.ru., благодаря которым мне удалось выстроить фундамент для некоторых фрагментов моих игр.

Говоря о собственных подходах к решению поставленной цели, отмечу, что самым верным был путь проб и ошибок. Так мне удалось подобрать максимально понятный для восприятия и уникальный способ реализации игр. Несмотря на это, у меня всегда был план, которому я максимально старалась придерживаться. Таким образом мне удалось грамотно рассчитать время на создание каждой логики.

# ОПИСАНИЕ МЕТОДОВ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

Первостепенной задачей в моем творческом проекте стало создание максимального комфорта пользования для игрока. Для этого мной был проведен небольшой опрос среди знакомых, благодаря которому мне удалось выяснить, что многие хотя бы раз слышали о выбранных мной играх (или играли в них), но при этом правила и суть для знакомых оставались либо расплывчатыми, либо вовсе забытыми. Таким образом, мной было предусмотрено наличие похожей ситуации, и результаты данного опроса нашли отражение в интерфейсе моего проекта с помощью следующих элементов и компонентов:

* ToolTip - компонент, отображающий текст при наведении указателя мыши на элементы управления;
* MessageBox - элемент, выводящий на экран окно с сообщением.

На втором месте по значимости стояла задача разработки интересного и красочного дизайна, сопровождающегося соответствующей музыкой. Для решения этого вопроса я использовала предназначенные для обработки фотографий приложения, чтобы сделать мои игры визуально более запоминающимися и яркими. Говоря о музыкальном сопровождении, стоит отметить, что изначально эта тема была для меня новой. Ввиду отсутствия опыта работы с нужными музыкальными форматами внутри кода, эта задача стала реальной только после изучения специального пространства имён System.Media, которое позволяет реализовать звуковую составляющую игры, а также после работы с конвертером звуковых файлов, с помощью которого у меня получилось преобразовать формат mp3 в wav, который необходим для использования в Visual Studio.

# ОПИСАНИЕ СТРУКТУР ДАННЫХ И АЛГОРИТМОВ

Ввиду того, что мой творческий проект состоит не из одной игры, а из целого сборника, то я опишу структуры данных и алгоритмов, обращаясь к каждому звену как к отдельному.

* Главное меню:

Оно представляет собой объект, использующийся для связи с каждой из форм, на которой расположены игры, а также имеет музыкальное сопровождение:

Form2 newform= new Form2();

newform.Show();

//создаём фоновую музыку с помощью волнового файла формата wav

System.Media.SoundPlayer sp = new System.Media.SoundPlayer(@"G:\Новая папка\проект по проге\48bb90af8e1e401.wav");

sp.PlayLooping();//зацикливаем музыку

* “Найди пару”:

1. Пауза: если таймер равен 0, то таймер включается, кнопки доступны. Иначе - таймер останавливается и выключается, кнопки становятся недоступными.

if (timer == 0)

{

timer1.Start();

timer++;

button19.Enabled = true;

pictureBox2.Enabled = true;

pictureBox3.Enabled = true;

pictureBox4.Enabled = true;

pictureBox5.Enabled = true;

pictureBox6.Enabled = true;

pictureBox7.Enabled = true;

pictureBox8.Enabled = true;

pictureBox9.Enabled = true;

pictureBox10.Enabled = true;

pictureBox11.Enabled = true;

pictureBox12.Enabled = true;

pictureBox13.Enabled = true;

pictureBox14.Enabled = true;

pictureBox15.Enabled = true;

pictureBox16.Enabled = true;

pictureBox17.Enabled = true;

}

else

{

timer1.Stop();

timer--;

button19.Enabled = false;

pictureBox2.Enabled = false;

pictureBox3.Enabled = false;

pictureBox4.Enabled = false;

pictureBox5.Enabled = false;

pictureBox6.Enabled = false;

pictureBox7.Enabled = false;

pictureBox8.Enabled = false;

pictureBox9.Enabled = false;

pictureBox10.Enabled = false;

pictureBox11.Enabled = false;

pictureBox12.Enabled = false;

pictureBox13.Enabled = false;

pictureBox14.Enabled = false;

pictureBox15.Enabled = false;

pictureBox16.Enabled = false;

pictureBox17.Enabled = false;

}

1. Перемешать: кол-во открытых элементов равно 0. Кнопка “начать” доступна. Первый цикл рандомит 8 пар картинок из ImageList. Второй цикл учитывает, свободна ли ячейка. Изначальное кол-во использованных пар - 0, и пока оно не равно 8, функционирует цикл, отвечающий за расставление картинок на свободные места.

button1.Enabled = true;

open\_elem = 0;

//создаём случайные пары картинок

Random rnd = new Random();

for (int i = 0; i < 8; i++)

{

para[i] = rnd.Next(8);

}

for (int i = 0; i < 16; i++)

{

pic[i] = -1; /\*-1 - ячейка свободна\*/

}

int ispolz = 0;

while (ispolz != 8)

{

int n1 = rnd.Next(16); /\*первое место из пары\*/

int n2 = rnd.Next(16); /\*второе место из пары\*/

if (n1 == n2) continue; /\*continue-оператор, который принудительно запускает цикл заново\*/

if (pic[n1] == -1 && pic[n2] == -1)

{

pic[n1] = pic[n2] = para[ispolz];

ispolz++;/\*количество использованных пар увеличивается\*/

}

}

hide();

hide() - метод, в котором каждому pictureBox присваивается одна и та же картинка - обложка, под которой находятся картинки:

public void hide()

{

pictureBox2.BackgroundImage = imageList1.Images[8];

pictureBox3.BackgroundImage = imageList1.Images[8];

pictureBox4.BackgroundImage = imageList1.Images[8];

pictureBox5.BackgroundImage = imageList1.Images[8];

pictureBox6.BackgroundImage = imageList1.Images[8];

pictureBox7.BackgroundImage = imageList1.Images[8];

pictureBox8.BackgroundImage = imageList1.Images[8];

pictureBox9.BackgroundImage = imageList1.Images[8];

pictureBox10.BackgroundImage = imageList1.Images[8];

pictureBox11.BackgroundImage = imageList1.Images[8];

pictureBox12.BackgroundImage = imageList1.Images[8];

pictureBox13.BackgroundImage = imageList1.Images[8];

pictureBox14.BackgroundImage = imageList1.Images[8];

pictureBox15.BackgroundImage = imageList1.Images[8];

pictureBox16.BackgroundImage = imageList1.Images[8];

pictureBox17.BackgroundImage = imageList1.Images[8];

}

1. Событие Click: преобразовываю sender в PictureBox, чтобы узнать, на какие картинки нажал пользователь. Затем преобразовываю в int, что с помощью свойства Tag узнать индекс нажатой кнопки.

PictureBox p = (PictureBox)sender;

int index = Convert.ToInt32(p.Tag);

//если мы нажимаем на одну и ту же картинку два раза, то она совпадает со своим же индексом (тэгом), и тогда мы выходим из метода (т.е. картинка не исчезает)

if (open\_elem == 1)

{

if (open[0] == index)

{

return;

}

}

if (open\_elem == 2)

{

hide();

open\_elem = 0;

}

open[open\_elem] = index;

open\_elem++;

if (open\_elem == 2)

{

if (pic[open[0]] == pic[open[1]])

{

//прячем пикбоксы, которые открыл пользователь

picbox[open[0]].Visible = false;

picbox[open[1]].Visible = false;

open\_elem = 0;

hide();

}

}

p.BackgroundImage = imageList1.Images[pic[index]]; - для PictureBoxов подбираются картинки из imageList1, опираясь на индекс.

1. Условие выигрыша: button19\_Click(null, null) - с помощью этой строки осуществляется автоматическое заполнение поля картинками. В данном примере m-минуты;s-секунды. Если все pictureBox стали невидимыми, на экране появляется окно сообщения с текстом, который зависит от того, за какое время пользователь прошел игру. Если игрок захочет начать игру заново, то форма с игрой принимает первоначальный вид, и все начинается по новой. В ином случае форма закрывается вместе с соответствующей музыкой; пользователю открывается главное меню.  
   if ((m == 0 && s > 0) || (m > 0 && s == 0))

{

if (pictureBox2.Visible == false && pictureBox3.Visible == false && pictureBox4.Visible == false && pictureBox5.Visible == false &&

pictureBox6.Visible == false && pictureBox7.Visible == false && pictureBox8.Visible == false && pictureBox9.Visible == false &&

pictureBox10.Visible == false && pictureBox11.Visible == false && pictureBox12.Visible == false && pictureBox13.Visible == false &&

pictureBox14.Visible == false && pictureBox15.Visible == false && pictureBox16.Visible == false && pictureBox17.Visible == false)

{

timer1.Stop();

if (m ==0 && s>=18)

{

DialogResult res = MessageBox.Show("Отлично! Вы прошли игру за " + (s+1) + " секунд. Вы занимаете почетное 1 место! Начать заново?", "Поздравление!", MessageBoxButtons.YesNo, MessageBoxIcon.Information, MessageBoxDefaultButton.Button1);

if (res == DialogResult.Yes)

{

hide();

button19\_Click(null, null);

m = 0;

s = 30;

label1.Text = Convert.ToString(m);

label2.Text = Convert.ToString(s);

label1.ForeColor = Color.Black;

label2.ForeColor = Color.Black;

button1.Enabled = false;

button19.Enabled = true;

button20.Enabled = false;

button17.Enabled = true;

pictureBox2.Enabled = false;

pictureBox3.Enabled = false;

pictureBox4.Enabled = false;

pictureBox5.Enabled = false;

pictureBox6.Enabled = false;

pictureBox7.Enabled = false;

pictureBox8.Enabled = false;

pictureBox9.Enabled = false;

pictureBox10.Enabled = false;

pictureBox11.Enabled = false;

pictureBox12.Enabled = false;

pictureBox13.Enabled = false;

pictureBox14.Enabled = false;

pictureBox15.Enabled = false;

pictureBox16.Enabled = false;

pictureBox17.Enabled = false;

pictureBox2.Visible = true;

pictureBox3.Visible = true;

pictureBox4.Visible = true;

pictureBox5.Visible = true;

pictureBox6.Visible = true;

pictureBox7.Visible = true;

pictureBox8.Visible = true;

pictureBox9.Visible = true;

pictureBox10.Visible = true;

pictureBox11.Visible = true;

pictureBox12.Visible = true;

pictureBox13.Visible = true;

pictureBox14.Visible = true;

pictureBox15.Visible = true;

pictureBox16.Visible = true;

pictureBox17.Visible = true;

}

else

{

this.Close();

sp.Stop();

SoundPlayer sp1 = new SoundPlayer(@"G:\Новая папка\проект по проге\48bb90af8e1e401.wav");

sp1.PlayLooping();//зацикливаем музыку

}

}

* “Пятнашки”:

1. Размещение кнопок:

//размещаем кнопки с помощью цикла

int x;

int y = 1;

for (int i = 0; i < 16; i++)

{

x = i - y \* 4;

y = i / 4;

}

1. Событие Click:

//создаём событие click, с помощью которого мы узнаем, какую кнопку нажал пользователь

//преобразуем sender в button для той же самой цели. Затем преобразуем в int, чтобы получить номер кнопки.

Button position = (Button)sender;

int n = (int)Convert.ToInt32(position.Tag.ToString());

//уменьшаем позицию на 1, т.к. изначально Tag был задан с 1

n--;

int y = n / 4;

int x = n - y \* 4;

//создаём кнопки, окружающие нажатую пользователем кнопку

int verh = y - 1;

int niz = y + 1;

int pravo = x + 1;

int levo = x - 1;

//получим номер правой кнопки при условии, что она существует только тогда, когда она меньше 4

if (pravo < 4)

{

int nom1 = y \* 4 + pravo;

if (!B[nom1].Visible)

{

B[nom1].Visible = true;

B[n].Visible = false;

B[nom1].Text = B[n].Text;

}

}

if (levo >= 0)

{

int nom2 = y \* 4 + levo;

if (!B[nom2].Visible)

{

B[nom2].Visible = true;

B[n].Visible = false;

B[nom2].Text = B[n].Text;

}

}

if (verh >= 0)

{

int nom3 = verh \* 4 + x;

if (!B[nom3].Visible)

{

B[nom3].Visible = true;

B[n].Visible = false;

B[nom3].Text = B[n].Text;

}

}

if (niz < 4)

{

int nom4 = niz \* 4 + x;

if (!B[nom4].Visible)

{

B[nom4].Visible = true;

B[n].Visible = false;

B[nom4].Text = B[n].Text;

}

}

1. Перемешать:

for (int i = 0; i < 15; i++)

{

b = rnd.Next(1, 16);

if (i == 0) /\*проверка на первый элемент, который не будет повторяться\*/

{

number[i] = b;/\*присваивание рандомного числа к элементу массива\*/

}

else

{

for (int j = 0; j < i; j++)/\*если число не первое, то запускается 2-ой цикл, в котором проверяются уже сгенерированные числа на повторение\*/

{

if (number[j] != b)

{

a = 0;/\*если так, то повторений нет\*/

}

else { a = 1; break; }/\*если так, то повторения есть\*/

}

}

if (a == 0)

{

number[i] = b;

}

else { i--; } /\*если повторения есть, то генерация числа начинается сначала\*/

}

//присваиваем тексту кнопок значения массива

button2.Text = Convert.ToString(number[0]);

button5.Text = Convert.ToString(number[1]);

button6.Text = Convert.ToString(number[2]);

button7.Text = Convert.ToString(number[3]);

button8.Text = Convert.ToString(number[4]);

button9.Text = Convert.ToString(number[5]);

button10.Text = Convert.ToString(number[6]);

button11.Text = Convert.ToString(number[7]);

button12.Text = Convert.ToString(number[8]);

button13.Text = Convert.ToString(number[9]);

button14.Text = Convert.ToString(number[10]);

button15.Text = Convert.ToString(number[11]);

button16.Text = Convert.ToString(number[12]);

button17.Text = Convert.ToString(number[13]);

button18.Text = Convert.ToString(number[14]);

* “Квадраты внимания”:

1. Перемешать:кнопка начать доступна; во избежание совпадения цвета текста на кнопках и самого цвета кнопки, при рандоме задаю параметры 100,255, избегая черный. Для более удобной и понятной работы мной была выбрана коллекция List, в которую я занесла стринговые значения (текст на подсказке). Если этот лист содержит более 0 элементов, то происходит рандом индекса от 0 до кол-ва элементов в List. Массиву кнопок присваиваются значения из этого же листа по индексу, и после использования они удаляются, тем самым обходя повторение элементов листа.

Random rnd = new Random();

button39.Enabled = true;

this.BackColor = Color.FromArgb(rnd.Next(100, 255), rnd.Next(100, 255), rnd.Next(100, 255));

textBox1.BackColor = Color.FromArgb(rnd.Next(100, 255), rnd.Next(100, 255), rnd.Next(100, 255));

List<string> listofletter = new List<string>

{

"O","n","H","F","J","l.","x",

"I","i","z","P","y","M","v",

"G","j","q","Q","E","t","s",

"u","w","C","k","A","L","K","m",

"N","r","D","o","R","B","p"

};

for (int i=0;i<36;i++)

{

B[i].BackColor = Color.FromArgb(rnd.Next(100, 255), rnd.Next(100, 255), rnd.Next(100, 255));

if (listofletter.Count > 0) /\*свойство Count отвечает за кол-во элементов внутри нашего листа\*/

{

int fIndex = rnd.Next(0, listofletter.Count);

B[i].Text = listofletter[fIndex];

listofletter.RemoveAt(fIndex);

}

}

1. Код одной кнопки как пример всех последующих: если выполняется условие, то индекс увеличивается вместе с финишем. ТекстБоксу присваивается значение по индексу из стрингового листа.

if (button2.Text == textBox1.Text)

{

index++;

textBox1.Text = listofletter2[index];

finish++;

}

1. Начать:все кнопки доступны; кнопка перемешки блокируется.

for(int i = 0; i < 36; i++)

{

B[i].Enabled = true;

}

timer1.Start();

timer++;

button40.Enabled = false;

button37.Enabled = true;

textBox1.Enabled = true;

textBox1.Text = "O";

finish = 0;

# ОПИСАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Перед пользователем появляется главное меню, которое состоит из трех игр на выбор: “Найди пару”, “Пятнашки”, “Квадраты внимания”. Переходя на выбранную игру, перед игроком появляется окно сообщений, которое уведомляет о том, что с сутью игр пользователь может ознакомиться, наведя курсор на каждую из них. Таким образом всплывет подсказка, с помощью которой игрок поймет, как ему необходимо действовать в той или иной игре.

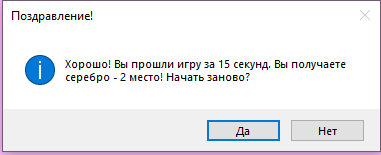
По наличию кнопок каждая игра идентична. Они имеют четыре команды: пауза, покинуть игру, перемешать и начать. Первоначально для пользователя доступны лишь две кнопки - перемешать и покинуть игру. После нажатия на первую из них разблокируется кнопка “начать”, но заблокируется предыдущая, и перед игроком появится функционирующее игровое поле, при нажатии на которое таймер начинает отсчитываться. После этого имеет смысл использовать кнопку “пауза”, которая становится также доступной.

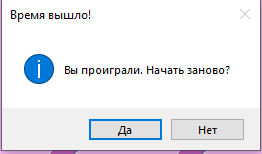
Главная задача для пользователя - как можно раньше пройти игру.

Пройдя её, перед пользователем выводится сообщение с результатом, за которое он прошел игру, в ином случае она будет не пройдена. Игрок может попробовать сыграть заново, а может покинуть игру.

# ОПИСАНИЕ РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММЫ

Чтобы оценить уровень памяти, внимания и логики, пользователю присваиваются места от одного до трех. В случае провала, игрок не занимает ни одного места.





# ДОСТИГНУТЫЕ РЕЗУЛЬТАТЫ ТВОРЧЕСКОГО ПРОЕКТИРОВАНИЯ

К достигнутым мной результатам в процессе разработки своего проекта я могу отнести:

* Создание красочного дизайна;
* Удобство пользования;
* Создание логики рейтинга;
* Использование новых элементов управления и компонентов;
* Работа со свойствами форм, с которыми раньше не приходилось работать;
* Использование нестандартных и непривычных методов решения проблем;
* Использование нового пространства имён.

# РЕКОМЕНДАЦИИ ПО ДАЛЬНЕЙШИМ ИССЛЕДОВАНИЯМ

Результаты моего творческого проектирования могут быть применены как в классных работах, так и в последующих будущих проектах.

Несмотря на хороший итог, полученный в ходе разработки сборника игр, так или иначе осталась одна нерешенная проблема: невозможность использования всех шестнадцати картинок в игре “Найди пару”, что повлекло за собой генерирование одинаковых фотографий в числе, превышающее два. Поэтому мне стоит продолжить исследования в направлении работы с ImageList`ами, Random`ом и свойствами.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://metanit.com/sharp/tutorial/4.5.php>
2. <https://devpractice.ru/c-sharp-lesson-10-generics/>
3. <https://shwanoff.ru/list-c-sharp/>
4. <https://www.youtube.com/watch?v=frDKP-A71W0&t=1932s>
5. <https://www.youtube.com/watch?v=1Vh_buDuyAo>
6. <https://www.youtube.com/watch?v=8v6ukIYLq8k>
7. <https://metanit.com/sharp/windowsforms/4.19.php>
8. <https://www.cyberforum.ru/csharp-beginners/thread194902.html>