|  |
| --- |
| Министерство образования и науки Российской Федерации |
| федеральное государственное бюджетное образовательное учреждение высшего образования |
| "Российский экономический университет имени Г.В. Плеханова" |
| **МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ** |

УТВЕРЖДАЮ

Заместитель директора по учебной работе

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Д.А. Клопов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 года

**Лабораторная работа № 10**

ОП 14 Объектно-ориентированное программирование

Тема: Реализация собственного игрового приложения. Выбор темы. Настройка. Работа с формой и компонентами. Разработка основных функций. Разработка дополнительных функций.

**Специальность: 09.02.05 «Прикладная информатика (по отраслям)»**

Квалификация: техник-программист

Разработчик: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Ю.А. Русакова

Рассмотрено ЦМК

«Профессиональных модулей 09.02.05»

«31» августа 2018 года. Протокол № 1

Председатель ЦМК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Л.А.Соколова

**Цель работы:** Изучив различные способы работы с графикой в C# и технологию Drag&Drop необходимо реализовать собственное игровое приложение на 2D графике.

**Количество часов на выполнение работы:** Продолжительность выполнения данной практической работы составляет 2 академических часа.

**Программное обеспечение:** Windows 7 и выше, Microsoft Office Word 2010 и выше, Visual Studio 2015 и выше.

**Теоретические сведения к выполнению лабораторной работы:**

2D-графика делится, как вы знаете, на растровую и векторную. Растровое изображение  — это набор цветных пикселей, заданных в прямоугольной области, хранящихся в файлах  \*.bmp, \*.jpg, \*.png и т.п.  Самый простой растровый редактор — программа Paint. Векторная графика намного экономнее (по объемам памяти) растровой. Так для рисования прямоугольника достаточно задать координаты двух точек (левого верхнего и правого нижнего углов) и цвет и толщину линии. В этом разделе в основном рассмотрим методы векторной графики.

Пространство имен **System.Drawing (Рисование)** обеспечивает доступ к функциональным возможностям графического интерфейса GDI+ , используя около 50 (!) классов, в том числе класс Graphics. Чуть позже мы будем использовать дополнительные пространства имен System.Drawing.Drawing2D,  System.Drawing.Imaging, System.Drawing.Printing, System.Drawing.Text, расширяющие функциональные возможности  библиотеки System.Drawing.

Класс **Graphics** предоставляет методы рисования на устройстве отображения(другие термины — графический контекст, «холст»). Определимся сразу, на чем мы хотим рисовать. Далее в примерах он обозначается как объект g.

Такие классы, как Rectangle (прямоугольник) и Point (точка) инкапсулируют элементы GDI+. Отметим, что Point вообще то является структурой (struct) с полями x,y. Это уточнение несущественно, так как в C# структуры похожи на классы,  a инициализация объекта-структуры point может выглядеть так же, как инициализация объекта-класса:  
Point point= new Point();

Класс **Pen** (перо) используется для рисования линий и кривых, а классы, производные от класса Brush (кисть) используются для закраски замкнутых контуров (см. ниже).

Класс **GraphicsPath** позволяет задавать последовательность соединенных линий и кривых, класс Region описывает внутреннюю часть графической формы, состоящей из многоугольников и контуров.

Класс Image – абстрактный базовый класс предоставляет функциональные возможности для производных классов Bitmap и Metafile. Bitmap используется для работы с пиксельными изображениями (см. выше пример). Metafile определяет графический метафайл, содержащий записи, описывающие последовательность графических операций, которые могут быть записаны (созданы) и воспроизведены (отображается). Этот класс не наследуется.

**Класс Graphics**

Он инкапсулирует поверхность рисования GDI+. Этот класс не наследуется. Методов в этом классе огромное количество, поэтому сначала представим их в таблице, а затем рассмотрим некоторые из них с примерами и пояснениями.  
В третьем столбце таблицы указывается число перегрузок метода, различающихся набором параметров (используйте интеллектуальную подсказку  IntelliSense для выбора нужного Вам варианта метода).

|  |  |  |
| --- | --- | --- |
| Имя метода | Описание | Число перегрузок |
| Clear(Color) | Очищает всю поверхность рисования и выполняет заливку поверхности указанным цветом фона. | 1 |
| CopyFromScreen(Point, Point, Size) | Выполняет передачу данных о цвете, соответствующих прямоугольной области пикселей, блоками битов с экрана на поверхность рисования объекта Graphics. | 4 |
| Dispose() | Освобождает все ресурсы, используемые данным объектом Graphics. | 1 |
| DrawArc(Pen, Rectangle, Single, Single) | Рисует дугу, которая является частью эллипса, заданного структурой Rectangle. | 4 |
| DrawBezier(Pen, Point, Point, Point, Point) | Рисует кривую Безье, определяемую четырьмя структурами Point. | 3 |
| DrawBeziers(Pen, Point[]) | Рисует несколько (N) кривых Безье, определяемых массивом из (3N+1) структур Point. | 2 |
| DrawCloseCurve(Pen, Point[ ]) | Рисует замкнутый фундаментальный сплайн | 4 |
| DrawEllipse(Pen, Rectangle) | Рисует эллипс | 4 |
| DrawIcon(Icon, Rectangle) | Рисует значок | 2 |
| DrawImage(Image image, int x, int y) | Рисует заданное изображение image, используя его фактический размер в месте с координатами (x,y) | 30 |
| DrawLine(Pen, Point, Point) | Проводит линию, соединяющую две структуры Point. | 4 |
| DrawLines(Pen, Point[ ]) | Рисует набор сегментов линий, которые соединяют массив структур Point. | 2 |
| DrawPath(Pen, gp) | Рисует пером Pen объект GraphicsPath gp. | 1 |
| DrawPie(Pen, Rectangle, Single, Single) | Рисует сектор, который определяется эллипсом, заданным структурой Rectangle и двумя радиалtьными линиями. | 4 |
| DrawPolygon(Pen, Point[]) | Рисует многоугольник, определяемый массивом структур Point. | 2 |
| DrawRectangle(Pen, Rectangle) | Рисует прямоугольник, определяемый структурой Rectangle. | 3 |
| DrawRectangles(Pen, Rectangle[]) | Рисует набор прямоугольников, определяемых структурами Rectangle. | 2 |
| DrawString(String, Font, Brush, PointF) | Создает указываемую текстовую строку в заданном месте с помощью определяемых объектов Brush и Font. | 6 |
| Equals(Object) | Определяет, равен ли заданный объект текущему объекту. (Унаследовано от Object.) | 1 |
| ExcludeClip(Rectangle) | Обновляет вырезанную область данного объекта Graphics, чтобы исключить из нее часть, определяемую структурой Rectangle. | 1 |
| ExcludeClip(Region) | Обновляет вырезанную область данного объекта Graphics, чтобы исключить из нее часть, определяемую структурой Region. | 1 |
| FillClosedCurve(Brush, Point[]) | Заполняет внутреннюю часть замкнутой фундаментальной кривой, определяемой массивом структур Point. | 6 |
| FillEllipse(Brush, Rectangle) | Заполняет внутреннюю часть эллипса, определяемого ограничивающим прямоугольником, который задан структурой Rectangle. | 4 |
| FillPath(Brush, GraphicsPath) | Заполняет внутреннюю часть объекта GraphicsPath. | 1 |
| FillPie(Brush, Rectangle, Single, Single) | Заполняет внутреннюю часть сектора, определяемого эллипсом, который задан структурой RectangleF, и двумя радиальными линиями. | 3 |
| FillPolygon(Brush, Point[]) | Заполняет внутреннюю часть многоугольника, определяемого массивом точек, заданных структурами Point. | 4 |
| FillRectangle(Brush, Rectangle) | Заполняет внутреннюю часть прямоугольника, определяемого структурой Rectangle. | 4 |
| FillRegion(Brush, Region) | Заполняет внутреннюю часть объекта Region. | 1 |
| Flush() | Вызывает принудительное выполнение всех отложенных графических операций и немедленно возвращается, не дожидаясь их окончания. | 2 |
| IntersectClip(Region) | Обновляет вырезанную область данного объекта,  включая в нее пересечение текущей вырезанной области  и указанной структуры | 3 |
| ResetClip() | Сбрасывает выделенную область g, делая ее бесконечной | 1 |

Класс Pen

Класс Pen определяет объект, используемый для рисования прямых линий и кривых. Этот класс не наследуется. Конструкторы класса (тоже пример перегрузки методов):  
1) Pen(Color)  инициализирует новый экземпляр класса Pen с указанным цветом.  
2) Pen(Color, Single) инициализирует новый экземпляр класса Pen с указанными свойствами Color и Width. (Width — устанавливает ширину пера Pen, в единицах объекта Graphics, используемого для рисования  
Например:

Pen redPen = new Pen(Color.Red);   // толщина пера по умолчанию 1 пиксельили используя промежуточную переменную green

Color green = Color.Green;

Pen greenPen = new Pen(green, 4.5f);

Можно вводить новый объект без указания явного имени пера (пример создания динамического объекта — пера):

g.DrawRectangle(new Pen(Color.Red, 3), r);

Например, для рисования прямоугольника r красным пером толщиной 3 пикселя, используя графический контекст («холст») g.

**Класс Brush**

Класс Brush определяет объекты, которые используются для заливки внутри графических фигур, таких как прямоугольники, эллипсы, круги, многоугольники и дорожки. Это абстрактный базовый класс, который не может быть реализован. Для создания объекта «кисть» используются классы, производные от Brush, такие как SolidBrush, TextureBrush и LinearGradientBrush,  например:

SolidBrush br = new SolidBrush(Color.Aqua);

g.FillRectangle(br, r); // закраска прямоугольника r на «холсте» g

Или без явного объявления объекта «кисть» br:  
g.FillRectangle(new SolidBrush(Color.Red), r);

**Этапы выполнения:**

1. Тема игры и её небольшое описание;
2. Правила игры;
3. Реализовать внешний вид игры исходя из правил и необходимых компонентов;
4. Реализовать основные функции:
   1. Реализация игры исходя из правил;
   2. Реализация настроек пользовательского интерфейса.
5. Реализовать дополнительные функции:
   1. Таблица рекордов;
   2. Ведение статистики по игре в виде графики;
   3. Добавить окно правил, для ознакомления пользователя;
   4. Добавить несколько уровней сложности (если это позволяют правила игры);
   5. Прокомментировать код программы;
   6. Любое логическое дополнение программы в рамках темы.

**Вывод:** При выполнении лабораторной работы, в которой …описываете что делали… были использованы следующие компоненты: …, из них дополнительно изучения требовали: …; для реализации функционала программы необходимо было применить классы: …; дополнительно в лабораторной работе было разобрано (применено, изучено) …. Основные проблемы при выполнении возникли с ….

**Контрольная работа:**

1. Доступ к каким возможностям предоставляет пространство имён System.Drawing?
2. Для чего необходимо событие DragDrop?
3. Укажите два основных события обрабатываемые при работе с DragDrop.
4. Дайте определение классу Graphics.
5. Дайте определение классу Pen