

## Семинар 2. Компьютерная графика.

### 2.1. Добро пожаловать в мир компьютерной графики!

Как ты знаешь, компьютерная графика появилась в середине 20 века. Многие из вас знакомы с такими игровыми сериями, как Battlefield, GTA, Call of Duty и так далее.

Но знаешь ли ты, с чего началась компьютерная графика? Если хочешь узнать, то вперед по ссылке! [Это видео](#) покажет тебе краткую историю эволюции компьютерной графики.

Нашей целью будет понять, каким образом были созданы самые первые игры, и попытаться повторить некоторые примеры, такие как арканойд или тетрис.



1. Battlefield 4



2. GTA V

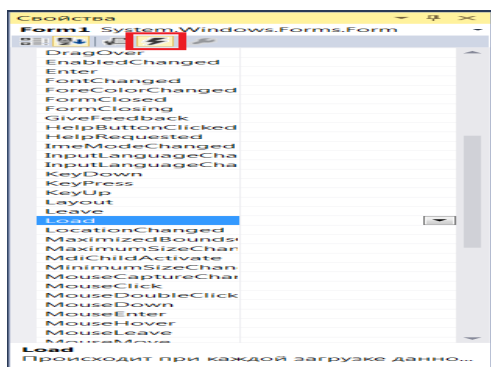
## 2.2. Основы основ. Готовим все необходимое.

Как ты мог увидеть, первые игры использовали лишь небольшой набор изображений. А чаще, это были вовсе не изображения, а просто какие-то линии или квадраты. Многие из вас уже изучали команды для рисования линий, а для тех, кто не изучал, мы сейчас о них расскажем.

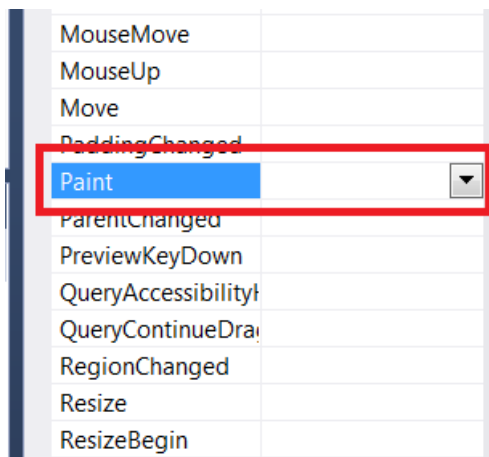
Итак, для начала давай подумаем, что необходимо художнику для рисования? Для начала, ему нужно место, где можно рисовать, или холст. Так же, ему потребуется кисть и краски.

C# - это тот же самый художник, только на компьютере. Ему тоже требуются краски, холст и кисть. Так что давай для начала разберемся, как предоставить ему все необходимое.

Начнем с создания нового консольного приложения. Его мы будем использовать в качестве основы. В Windows Forms приложении для рисования есть специальное событие, которое нам нужно обрабатывать, и в котором должен находиться весь код рисования. Это событие называется Paint. Для того, чтобы создать новый обработчик событий, нужно в свойствах формы выбрать вкладку события.



Далее, нужно найти событие Paint и сделать на нем двойной щелчок.



Ты перейдешь к обработчику событий Paint.

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
}
}
```

Здесь мы и будем давать все команды для рисования.

Итак, теперь давай создадим все необходимое для рисования (холст, кисти и краски).

Начнем с создания самого художника.

В нашем обработчике событий дай такую команду:

```
Graphics g = CreateGraphics();
```

Как ты помнишь из занятий первого семестра, эта команда создаст новый объект типа Graphics и сохранит его в переменную g.

Теперь приступим к созданию кисти. В роли кисти будет выступать объект Pen.

Для создания ручки дай следующую команду:

```
Pen myPen = new Pen(Color.Red);
```

Color.Red – это цвет, которым будет рисовать кисть. Ты можешь указать свой собственный. Чтобы увидеть список всех доступных цветов по умолчанию, напиши Color. и нажми Ctrl + Пробел.

Ты наверное уже догадался, что в качестве холста мы будем использовать саму форму, так что нужды в её создании нет.

Итак, у нас все готово к началу и давай, наконец, что-нибудь нарисуем!

### 2.3. Параллельные линии.

Мы начнем изучение графики с самой простой и основной команды – рисования линии на плоскости.

Нетрудно догадаться, что для того, чтобы нарисовать линию нам нужно всего лишь две точки. Любую линию можно задать с помощью двух точек – точки начала и точки конца. Без лишних слов, вот команда для рисования линии:

```
g.DrawLine(myPen, x1, y1, x2, y2);
```

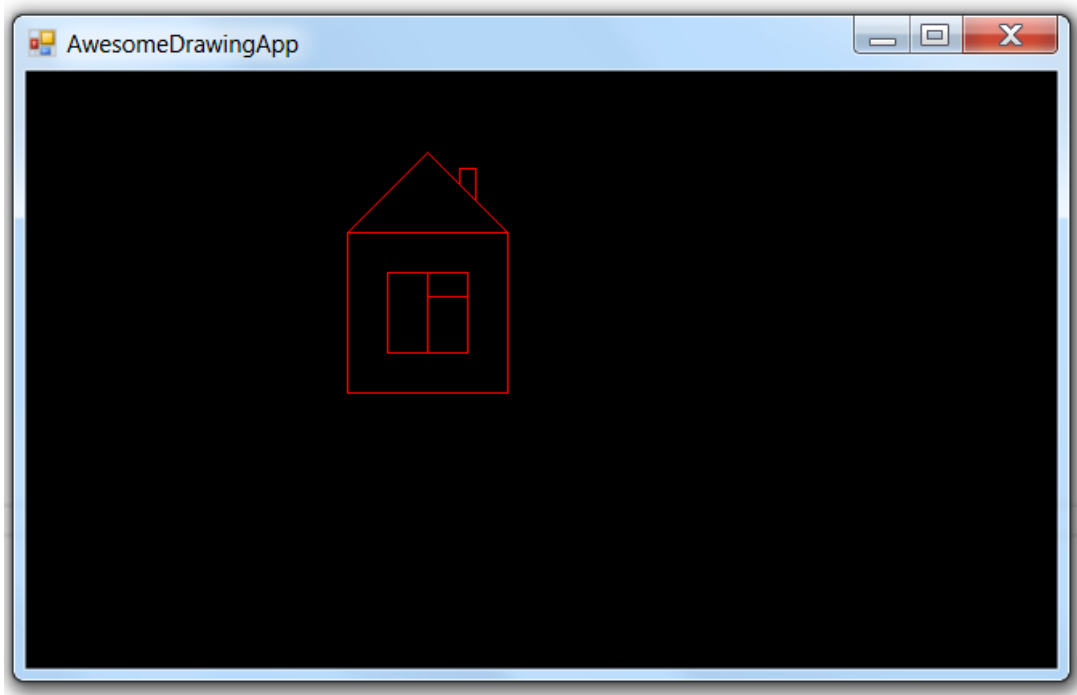
Здесь x1, y1, x2 и y2 – это координаты двух точек. Как видишь, мы используем ранее созданный объект g. Именно у него мы вызываем команду рисования линии. Как ты наверняка помнишь, g и есть переменная, в которую мы сохранили нашего «художника». Помимо этого, обрати внимание на то, что мы так же используем ранее созданную переменную myPen в качестве параметра. Это и есть «кисть», с помощью которой рисует наш «художник».

Можно указать любые значения вместо x1, y1, x2, y2. Например, чтобы нарисовать линию поперек экрана, можно дать такую команду:

```
g.DrawLine(myPen, 0, 0, Width, Height);
```

### Задание

А теперь попробуй нарисовать вот такой домик самостоятельно!



Ну как, сложно? Если нет, то давай переходить к следующей части – рисованию окружностей!

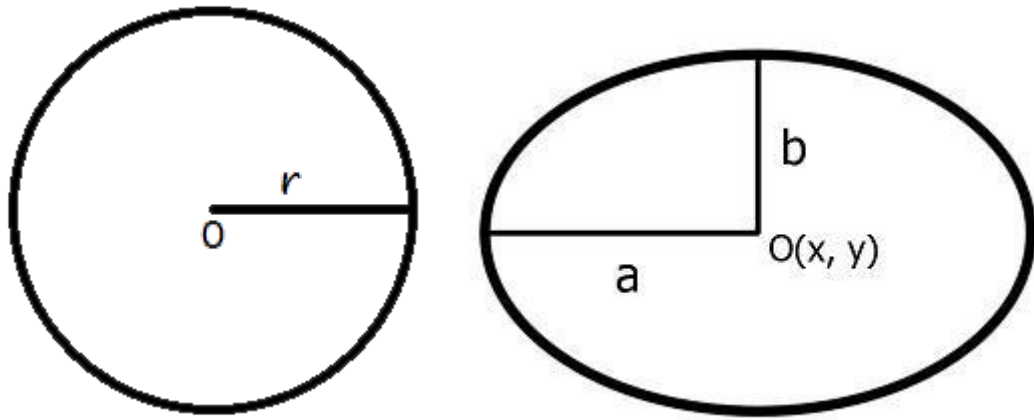
## 2.4. Окружности.

Если использовать только линии, то наша графика будет весьма скучная. Поэтому давай разберемся, как можно рисовать окружности!

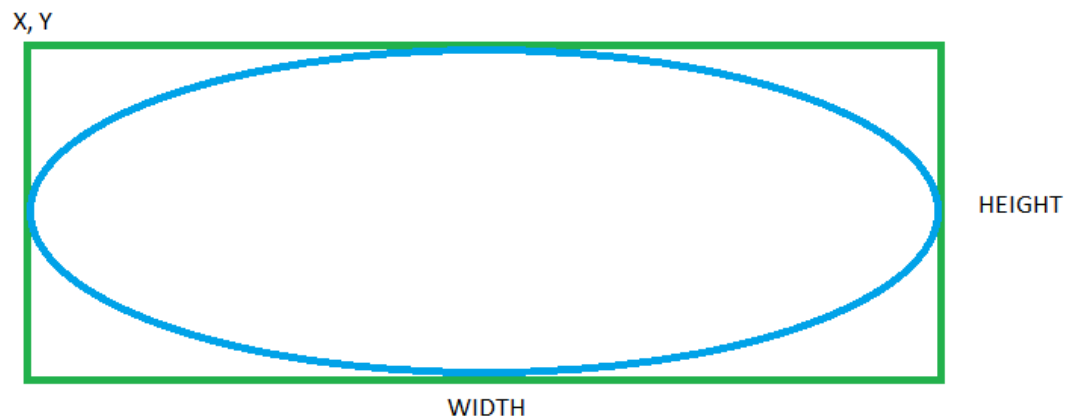
Вот такая команда используется для рисования окружностей

```
g.DrawEllipse(myPen, x, y, width, height);
```

Интересный набор параметров, не так ли? Логично было бы предположить, что для рисования окружности нам только потребуются координаты центра и радиус? А вот и нет, дело в том, что в C# используется более общий подход. Ведь круг – это разновидность эллипса, у которого радиус всегда равен одному и тому же значению.



Поэтому ребята из команды разработчиков C# решили, что можно использовать одну команду для рисования и эллипсов, и окружностей. А вот для описания эллипса только лишь координаты центра и радиуса уже не достаточно. Так что в C# для описания эллипса используется прямоугольник.



Как видишь, у нас получается эллипс, вписанный в прямоугольник. И  $x, y$  – это координаты левого верхнего угла, а  $width$  и  $height$  – ширина и высота соответственно. А теперь придумай сам, как можно нарисовать круг!

А теперь, вооружившись всеми этими навыками, создай какое-нибудь произведение искусства! Для начала, ты можешь добавить солнце и дым из трубы к нашей предыдущей работе с домиком.

Далее, единственным ограничением будет твоя фантазия! (Ну и не стоит забывать о нормах приличия и эстетики! ;) )