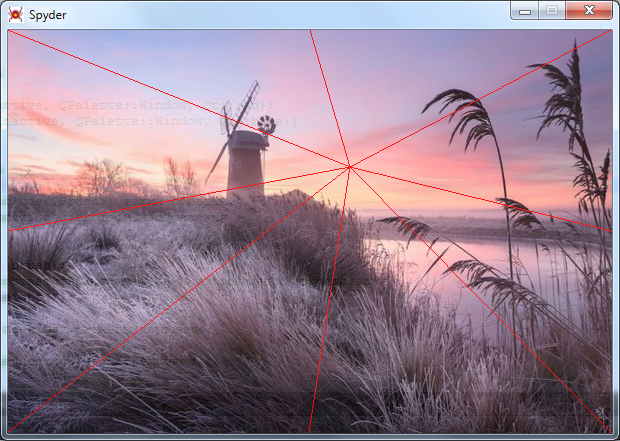
# Задание 1.

Добавить к последнему заданию прошлого занятия таймер, чтобы каждые 10 секунд появлялись новые 10 случайных фигур.

# Задание 2.

Рисовать на виджете паутину.



1. Рисование должно происходить при нажатой левой кнопке мыши;
2. Если левую кнопку отпустить – паутина пропадает;
3. Центр паутины – позиция курсора;
4. Также перерисовка осуществляется и при движении мыши в пределах виджета при нажатой левой кнопке;
5. Изменяйте вид курсора на паука **при рисовании**;

Доп задание (+20 баллов):

1. При нажатых клавишах Ctrl и правой кнопке мыши – на экране должны отобразиться текущие координаты курсора;
2. Добавьте фон виджета и сделайте его прозрачным на 10 процентов.

( void QWidget::setWindowOpacity(qreal level) ( level принимает значение от 0 до 1) )

## Порядок действий для рисования:

1. Так как паутину нужно рисовать далеко не всегда (а только при нажатой левой кнопке мыши)=> нам нужен признак =>в классе объявите булеву переменную, например, isDrawing. Как и где нужно проинициализировать эту переменную?
2. При нажатии левой кнопки мыши (в обработчике mousePressEvent()) флажок isDrawing нужно устанавливать
3. При отпускании, соответственно, сбрасывать
4. При перемещении мыши  
   void QWidget:: mouseMoveEvent ([QMouseEvent](file:///C:\Users\user\AppData\Roaming\Microsoft\Word\qmouseevent.html) \* *event* ) [virtual protected]  
   если требуется рисовать, вызываем перерисовку окна

## Рисование линий

Напоминание: рисовать стоит только в том случае, когда установлен флаг IsDrawing

* Получаем размеры клиентской области
  + QRect QWidget::rect() const **или через**
  + Знакомые нам методы int QWidget::width() и int QWidget::height()
* Рисуем паутину, используя текущую точку курсора и размеры клиентской области
* Текущая точка курсора извлекается из методов обработчика событий мыши. Где удобно хранить текущее положение курсора?
* Подсказка: у класса QRect есть полезные методы для получения левого верхнего угла (topLeft()), и других углов
* Цвет паутины тоже можно задавать по желанию

## Порядок действий для отображения подсказки : (п.5)

1. В обработчике mousePressEvent() анализируем, какая кнопка была нажата (нас интересует правая). Для этого вызываем метод QMouseEvent::button();
2. Если нажата правая кнопка, из полученного параметра извлекаем экранные координаты мыши (например, с помощью метода pos());
3. Формируем строку подсказки вида «x=<значение>, y=<значение>» c помощью объекта класса QString
   1. Подсказки:
      1. для конкатенации строк можно использовать перегруженные операторы “+” и “+=”класса QString
      2. для перевода целого значения в строковое представление можно пользоваться статическим методом

QString QString::number(int value) [static];

1. Не забудьте #include<QToolTip>
2. Вывод подсказки осуществляется методом:

void QToolTip::showText ( const [QPoint](file:///C:\\Users\\user\\AppData\\Roaming\\Microsoft\\Word\\qpoint.html) & *pos*,const [QString](file:///C:\\Users\\user\\AppData\\Roaming\\Microsoft\\Word\\qstring.html) & *text*, [QWidget](file:///C:\Users\user\AppData\Roaming\Microsoft\Word\qwidget.html) \* *w=0* ) [static], где pos – координаты вывода подсказки, text –текст подсказки

А теперь будем выводить tooltip при нажатии правой кнопки мыши, **только** если нажата клавиша Ctrl (с помощью метода modifiers()).

void MainWindow::*keyPressEvent*(QKeyEvent \* event)

{

if(event->modifiers() && <константа Qt::Key для клавиши Ctrl>)

…

}