

C# und OpenCV ausprobieren

- Installiere VisualStudio
 - Lade <http://www.github.com/gewitterwolkeS/OpenCV-Testen> herunter (Code->Download ZIP)
 - Öffne das enthaltene VisualStudio-Projekt.
 - Erledige folgende Aufgaben. Für jede Aufgabe erstellst du am besten eine Kopie des heruntergeladenen Projekts.
1. Das Fenster soll nur einen Button mit dem Namen „myButton“ und der Beschriftung „Ich bin ein Button“ enthalten.
 - a. Sobald du auf den Button drückst, soll sich der Fenstertitel zu „Button wurde gedrückt“ ändern.
 - b. Sobald du auf den Button drückst, soll sich der Fenstertitel zu „Button wurde ? mal gedrückt“ ändern, wobei das ? für die Anzahl der Klicks steht.
 - c. Wurde öfter als 5-mal geklickt, soll sich der Buttontext zu „es ist genug“ ändern. Außerdem soll er nun rückwärts bis 0 zählen.
 2. Das Fenster soll ein Dockpanel mit dem Namen „myDock“ enthalten. In diesem Dockpanel soll links eine Stackpanel mit dem Namen „myStack“ angedockt sein, der Rest soll durch ein Image mit dem Namen „myImage“ aufgefüllt sein. Du kannst aus dem Paket *wpfHelp* für Buttons die Klasse „AutoActionButton“ benutzen.
 - a. Im Stackpanel soll ein Button mit dem Namen „openButton“ und der Beschriftung „öffnen“ sein. Sobald er gedrückt wird, soll ein „OpenFileDialog“ erscheinen. Falls dieser Dialog mit „OK“ beendet wird, soll im Image das ausgewählte Bild erscheinen.
 - b. Wird nun irgendwo auf das Bild geklickt, soll an der Klickposition ein Kreis gezeichnet werden.
 - c. An diese Position soll außerdem die Koordinate des Punktes geschrieben werden.
 - d. Im Stackpanel soll außerdem ein Button mit dem Namen „refreshButton“ sein. Wird auf ihn geklickt, soll im Image wieder das ursprünglich ausgewählte Bild erscheinen
 - e. Beim Start wird von Haus aus ein rotes Bild mit den Abmessungen „600x400“, (MatType.CV_8UC3) dargestellt. Dieses rote Bild wird beim Refreshing auch angezeigt, wenn überhaupt nie ein Bild geöffnet wurde.

- f. Wird mehr als einmal geklickt, sollen die Punkte durch Linien verbunden werden.
 - g. Außerdem soll bei der Mitte der Linie der Winkel geschrieben werden, den die Linie gegenüber der positiven x-Achse hat (von -180° bis $+180^\circ$). Dazu schreibt man eine Methode, die den Winkel als Rückgabewert hat und als Argumente den Startpunkt und den Endpunkt erwartet.
 - h. Im Stackpanel soll außerdem ein `NamedIntUpDown`-Element aus dem Paket `wpfHelp` sein. Die gewählte Zahl soll den Radius der Kreise repräsentieren.
 - i. Im Stackpanel sollen mehrere `EnumCombo`-Elemente aus dem Paket `wpfHelp` sein. Damit soll man die Farbe des Kreises und der verschiedenen Texte auswählen können. Dazu erstellst du am besten eine Enum mit einigen Farben und schreibst eine Methode, die die Enum-Werte in die entsprechenden Scalar-Werte umwandelt.
 - j. Wird öfter als zweimal geklickt, soll der Winkel, den zwei aufeinanderfolgende Linien haben, an der Verbindungsstelle eingezeichnet werden (Winkel >0 bedeutet: Linkskurve, Winkel <0 bedeutet Rechtskurve). Dazu ist auch eine Methode brauchbar, welche die drei Punkte als Argumente braucht und den Winkel zurückgibt.
3. Das Fenster soll nur ein Image enthalten. Es soll bereits beim Starten des Programms ein Dialog erscheinen, indem man ein Bild auswählt (bzw. alternativ ein rotes Bild (600x400, `MatType.CV_8UC3`)). Es soll auch gleich ein Kreis in die Mitte des Bildes gemalt werden.
- a. Drückt man auf die Cursortasten, soll der Kreis nach rechts, links, oben oder unten wandern (bis er den Bildrand erreicht).
 - b. Der Kreis soll sich nun automatisch über den Bildschirm bewegen. Die Bewegungsrichtung soll über die links/rechts Tasten verändert werden, die Geschwindigkeit mit den oben/unten Tasten. Erreicht er den Bildrand, soll er zurückprallen (Einfallswinkel = Reflexionswinkel)