

Übung 3

Laden Sie zunächst von Moodle die Datei „Paint.zip“ herunter mit dem Codegerüst für diese Übung. Programmieren Sie auf dieser Basis mit C++ unter Verwendung der 2D-API von Qt ein kleines Malprogramm, das die folgenden graphischen Objekte darstellen kann (je als eigene Klasse implementiert):

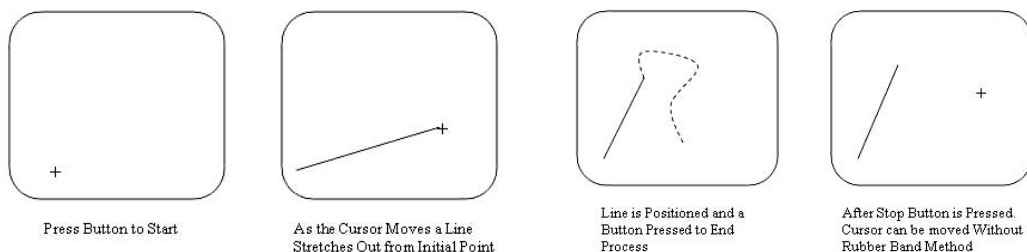
- Linie (geg. durch Anfangs- und Endpunkt)
- Rechteck (z.B. durch Angabe der linken oberen sowie rechten unteren Ecke)
- Kreis (geg. durch Mittelpunkt und Radius)
- Freihandlinie (lässt sich in Qt mit Hilfe einer sog. Polyline recht einfach abbilden)

Die o.g. Objekttypen sollen dabei von einer abstrakten Basisklasse namens `GraphObj` abgeleitet sein, welche insbes. virtuelle Methoden zum Zeichnen und Aktualisieren (beim Erstellen) des jeweiligen Objekts anbietet (analog zum Polymorphie-Beispiel aus der Vorlesung).

Die Zeichenmethode muss vom `paintEvent()` aus aufgerufen werden und könnte z.B. folgende Signatur haben, wobei der Parameter `p` die im `paintEvent()` angelegte `QPainter`-Instanz ist, da davon immer nur eine gleichzeitig verwendet werden kann: `void display(QPainter *p);`

Die Objekte sollen dabei graphisch-interaktiv mit Hilfe der Maus definiert werden. Bei Linien (bei Auswählen von „Line“) soll das Teilstück zwischen dem zuletzt durch Klick eingegebenen Punkt und der aktuellen Maus-Position während der Mausbewegung kontinuierlich angezeigt werden. Wird ein Mausbutton gedrückt, so wird der Startpunkt gesetzt.

Wird die Maus bei gedrückter Maustaste bewegt, so wird je die Linie bis zur aktuellen Mausposition gezeichnet (sog. Rubber-Band-Methode; siehe Abbildung). Wird der Mausbutton schließlich losgelassen, so ist die Linie fertig gezeichnet. Nutzen Sie zum Abspeichern der Punkte am besten direkt den Qt-Datentyp `QPoint`.



Kreis und Rechteck sollen je durch Auswahl eines Startpunktes (im „`mousePressEvent`“) und Aufziehen mit der Maus („`mouseMoveEvent`“) – genau wie schon bei der Linie – entsprechend einer sog. Gummiband-Visualisierung (Rubber Band) erzeugt werden.

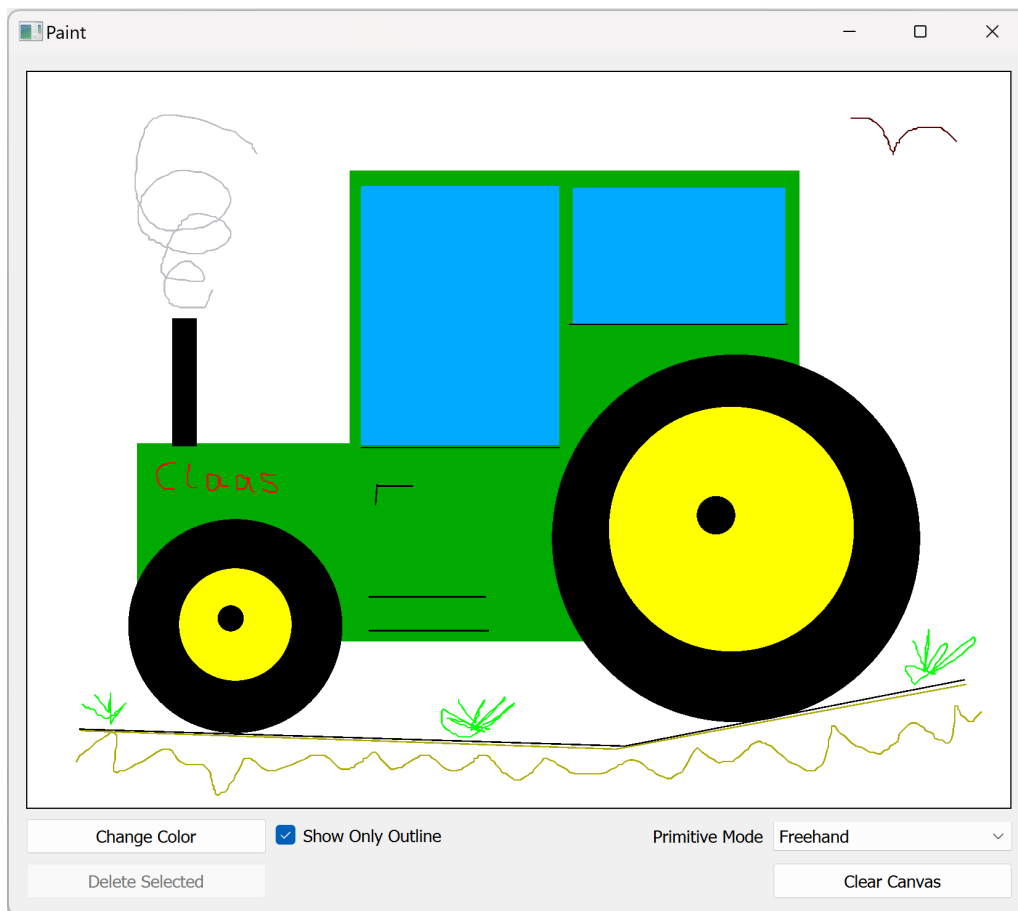
In ähnlicher Weise kann die Freihandlinie aus den Mauspositionen erzeugt werden, welche sich bei Bewegung mit gedrückt gehaltener Maustaste ergeben.

Dabei sollen –wie im Polymorphie-Beispiel aus der Vorlesung – alle zu zeichnenden Objekte von einem `Scene`-Objekt verwaltet werden, welches auch für die Darstellung zuständig ist.

Implementieren Sie außerdem noch eine Clear-Funktionalität (bei Klick auf den Button „Clear Canvas“ soll die Zeichnung wieder verschwinden).

Weiterhin sollen Objekte ohne nennenswerte Größe (z.B. < 2 Pixel), die durch versehentlichen Klick o.ä. entstanden sind, direkt wieder aus der Szene gelöscht werden.

Die folgende Abbildung zeigt das Malprogramm in Aktion (die Zeichenfläche nennt man in der Computergraphik übrigens typischerweise „Canvas“ bzw. „Viewport“):



Klickt man hier auf „Change Color“ kann man eine Farbe auswählen. Diese soll die Zeichenfarbe für das nächste zu erstellende graphische Objekt sein. Ist das Häkchen bei der Checkbox „Show only Outline“ nicht gesetzt, sollen die Objekte gefüllt gezeichnet werden (wie z.B. bei den Rädern zu sehen ist). Andernfalls soll nur der Umriss dargestellt werden (also ungefüllt).