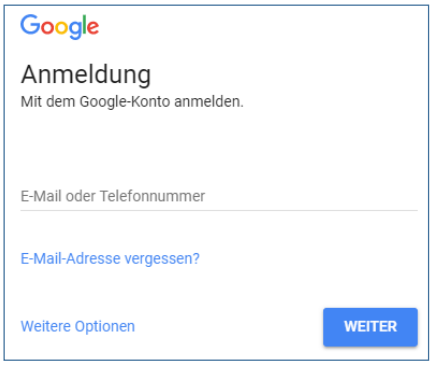
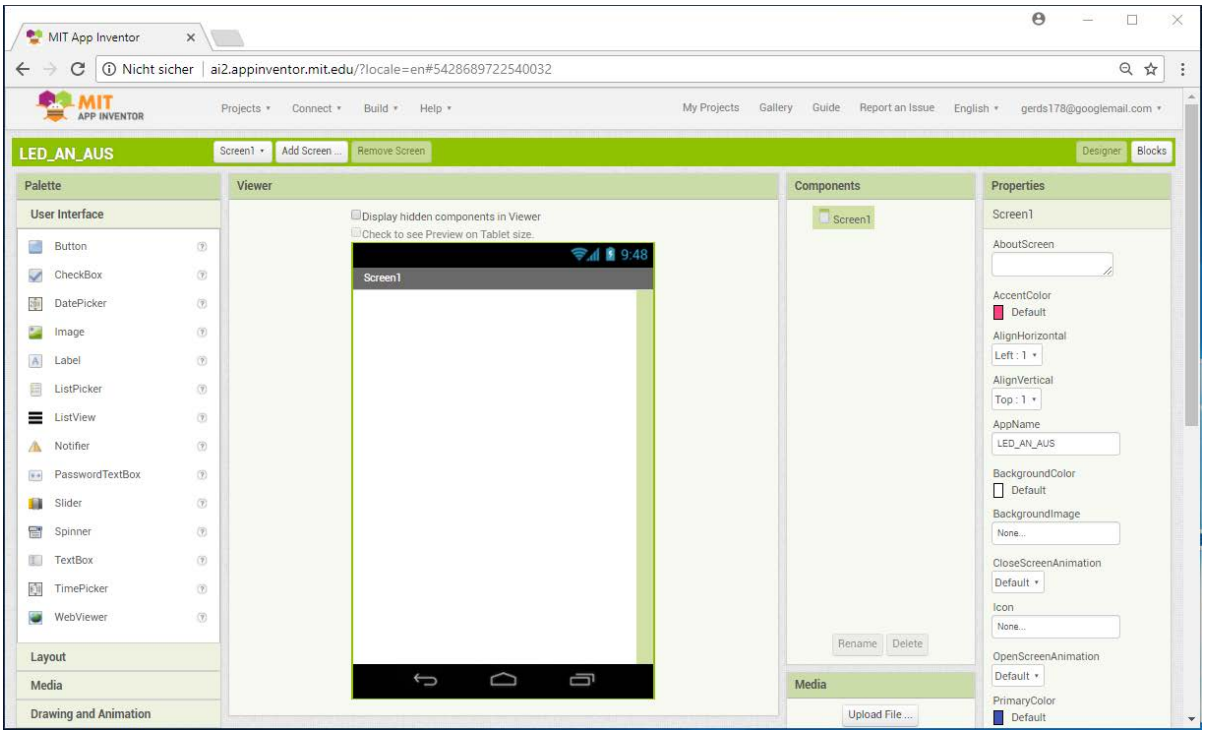
Schritt für Schritt Anleitung zum Erstellen einer App zum Ein- und Ausschalten einer LED

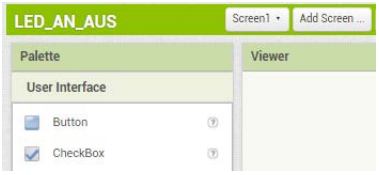
Durch Eingabe von http://ai2.appinventor.mit.edu/ in die Adresszeile eines Webbrowsers gelangt man zu seinem Google-Konto, über das der Zugang zum „MIT App Inventor“ geregelt wird. Als Neuling muss man sich registrieren und den weiteren Anweisungen folgen. Hilfen (auf Englisch) gibt es unter der Adresse http://appinventor.mit.edu/explore/get-started.html.



Nach erfolgreichem Start der Anwendung muss zum Entwickeln einer neuen App in der Titelzeile zunächst auf <Projects> und dann <Start new project> angewählt werden. In dem sich öffnenden Fenster einen Namen (hier: LED_AN_AUS) eingeben und mit <OK> bestätigen.

Daraufhin zeigt sich der App-Inventor in folgenden Gewand:

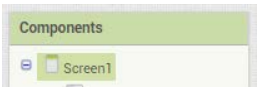




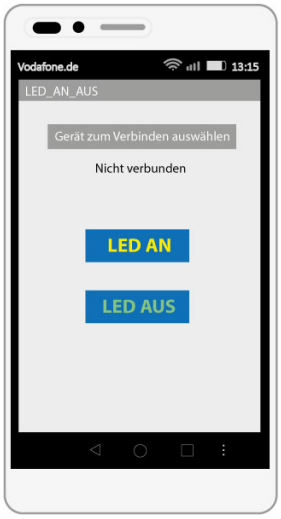
Auf der linken Seite finden sich unter <Palette> alle möglichen Komponenten zur Verwendung auf dem Handy.

In der Mitte wird die zunächst noch leere Oberfläche des Handys angezeigt.

Halbrechts werden in einer Spalte alle Komponenten aufgelistet, die auf dem Handy (im „Viewer“) platziert sind.



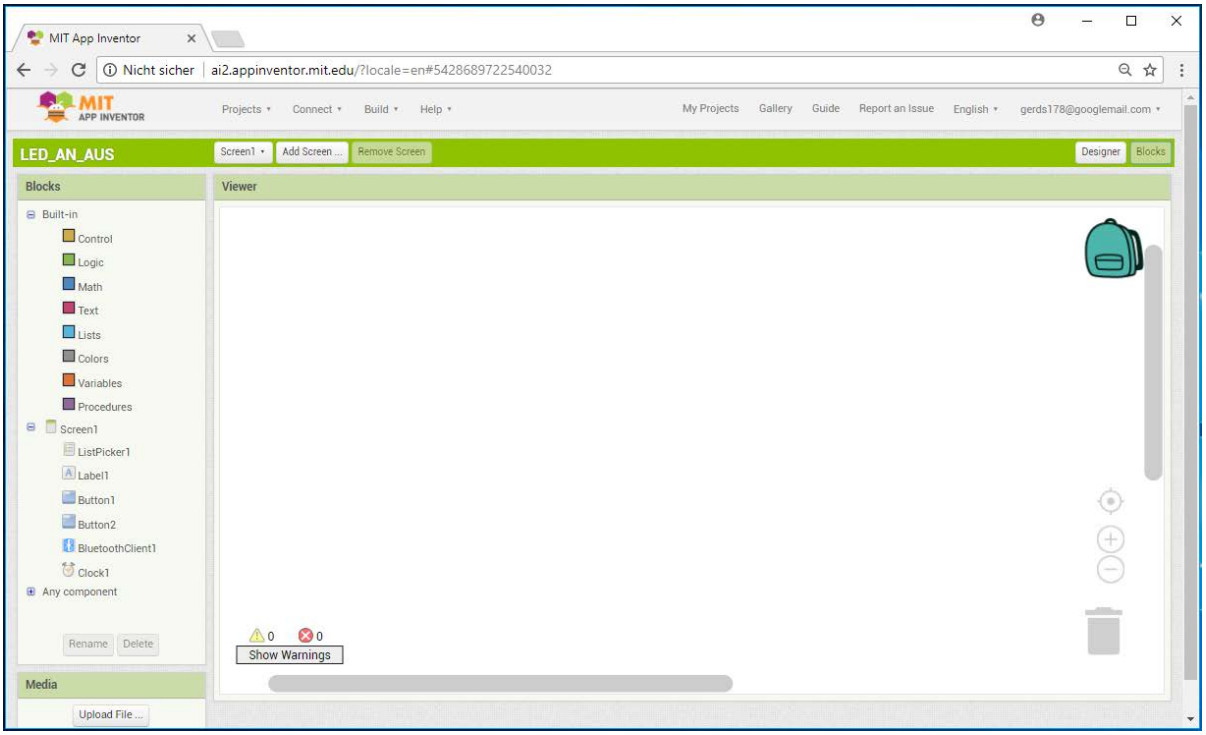
Die Spalte rechts außen zeigt – je nachdem welche Komponente angewählt ist – deren voreingestellten Details an. Durch Veränderung der Einstellungen dort kann deren Erscheinungsbild verändert werden.



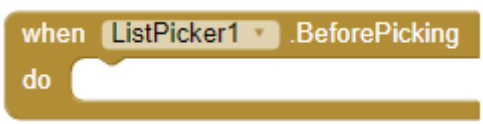
Um der selbst gestellten Aufgabe, also eine LED vom Handy an- und ausschalten zu können, müssen aus der Palette die benötigten Komponenten entweder auf den Bildschirm („Screen1“) oder die Oberfläche des „Viewers“ gezogen werden. Der Reihe nach sind das für unsere Zwecke:

1. Ein „ListPicker“, der eine Liste generiert, aus der das Bluetooth-Modul am Arduino gewählt werden kann.
2. Ein „Label“, das anzeigt, dass eine Verbindung zum Bluetooth-Modul hergestellt worden ist.
3. Zwei „Button“, um die Schaltvorgänge auslösen zu können.
4. Ein „BluetoothClient“. Er findet sich in der Palette unter „Connectivity“. Da er im Hintergrund arbeitet, bleibt er unsichtbar und zeigt sich – obwohl er ebenfalls auf „Screen1“ gezogen werden muss – unterhalb des Screens bei den „Non-visible components“.
5. Üblich ist auch, eine Uhr mit einzubinden. Sie findet sich als „Clock“ in der Palette unter „Sensors“. Auch sie bleibt unsichtbar.

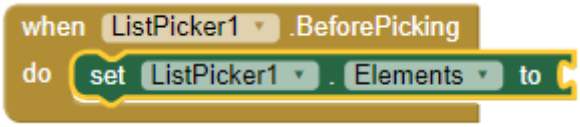
Damit sind alle Komponenten für die Programmierung vorhanden, die grafisch (mit Hilfe von Blöcken) erfolgt. Die dazu nötige Benutzeroberfläche öffnet sich bei einem Klick auf „Blocks“ in der grünen Zeile rechts oben.



Ein Klick auf „ListPicker1“ öffnet ein Auswahlmenü, bei dem folgender Baustein ausgewählt werden muss:



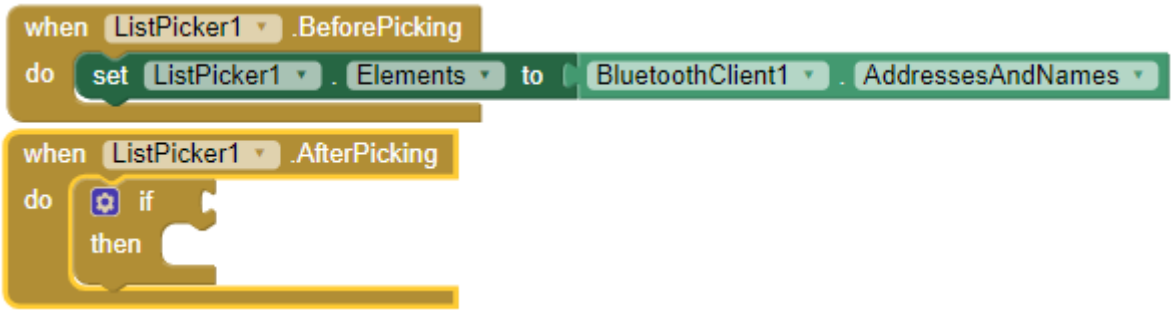
Ein zweiter Klick öffnet das Auswahlmenü erneut, bei dem „set ListPicker1.Elements to“ ausgewählt und in den Zwischenraum des ersten Elements eingefügt werden muss. Daraus ergibt sich folgendes Bild:



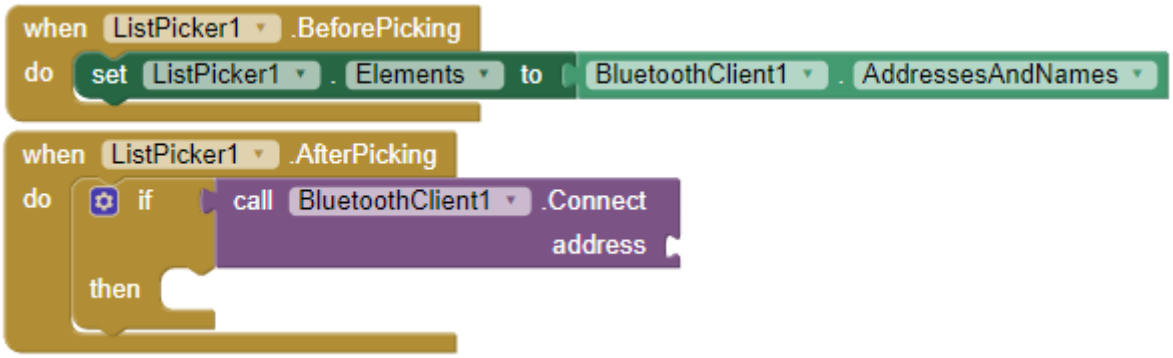
Danach wird unter „Blocks““BluetoothClient1 der „BluetoothClient1“ an-, der Baustein „BluetoothClient1.AdressesAndNames“ ausgewählt und mit dem vorigen Element verbunden. So ergibt sich das folgende Bild:

Damit ist ein erster Programmierabschnitt abgeschlossen: Die unter Bluetooth verfügbaren Geräte werden in einer Liste zusammengefasst und zur Anzeige gebracht. Das geschieht vor der Auswahl („BeforePicking“).

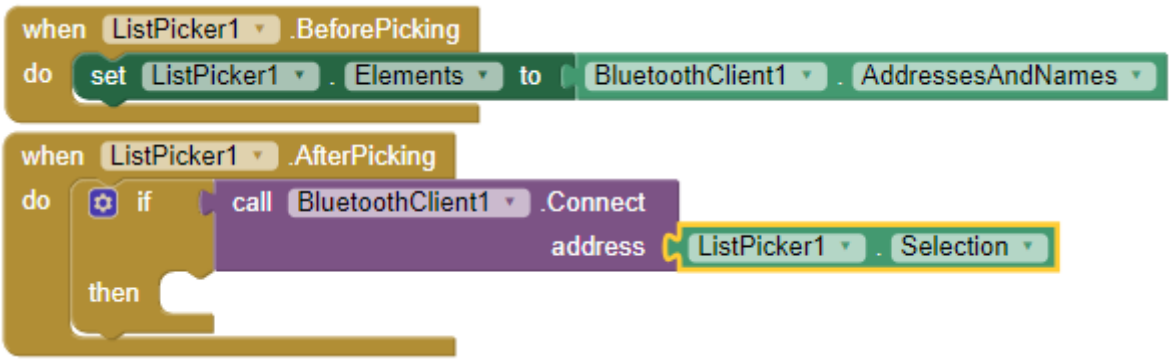
Dann muss durch festgelegt werden, was nach der Auswahl („AfterPicking“) passieren soll. Dabei handelt es sich um eine Wenn-dann-Beziehung. Der Baustein findet sich unter „Control“.



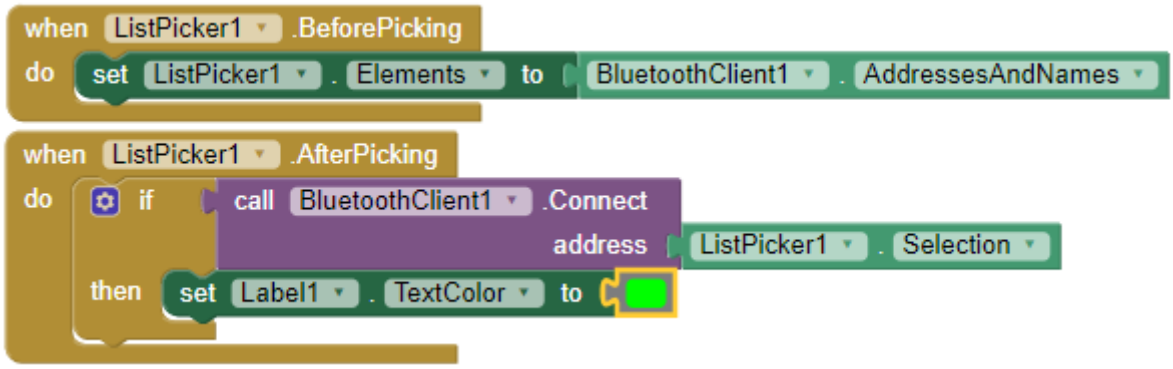
Wenn der „BluetoothClient1“ eine Verbindung anfragt („call BluetoothClient1.Connect“) …



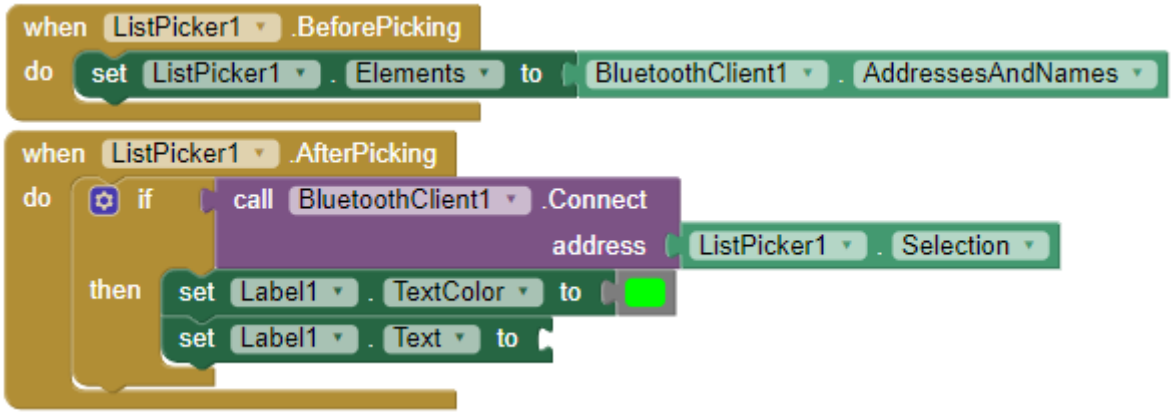
… soll er berücksichtigen, welche Wahl (“address ListPicker1.Selection“) getroffen worden ist …



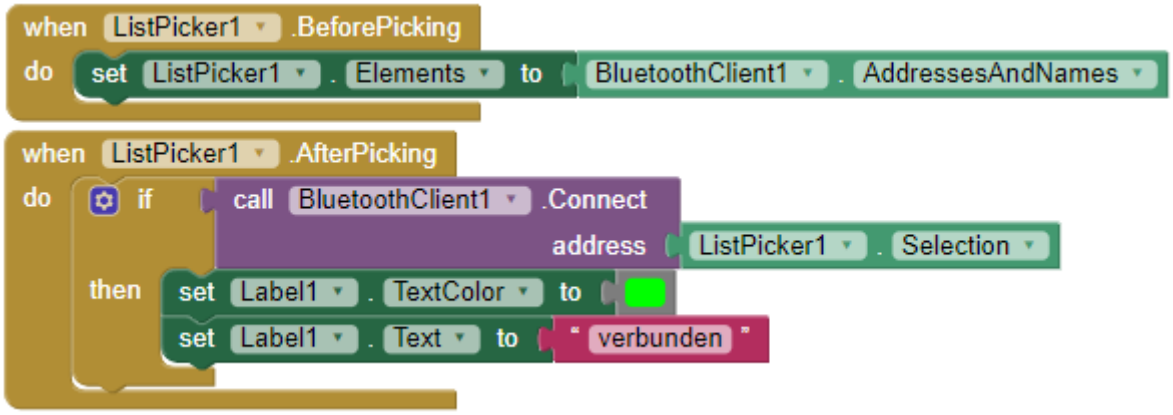
… dann die Textfarbe von „Label1“ grün (Baustein erreichbar über „Colors“) machen …



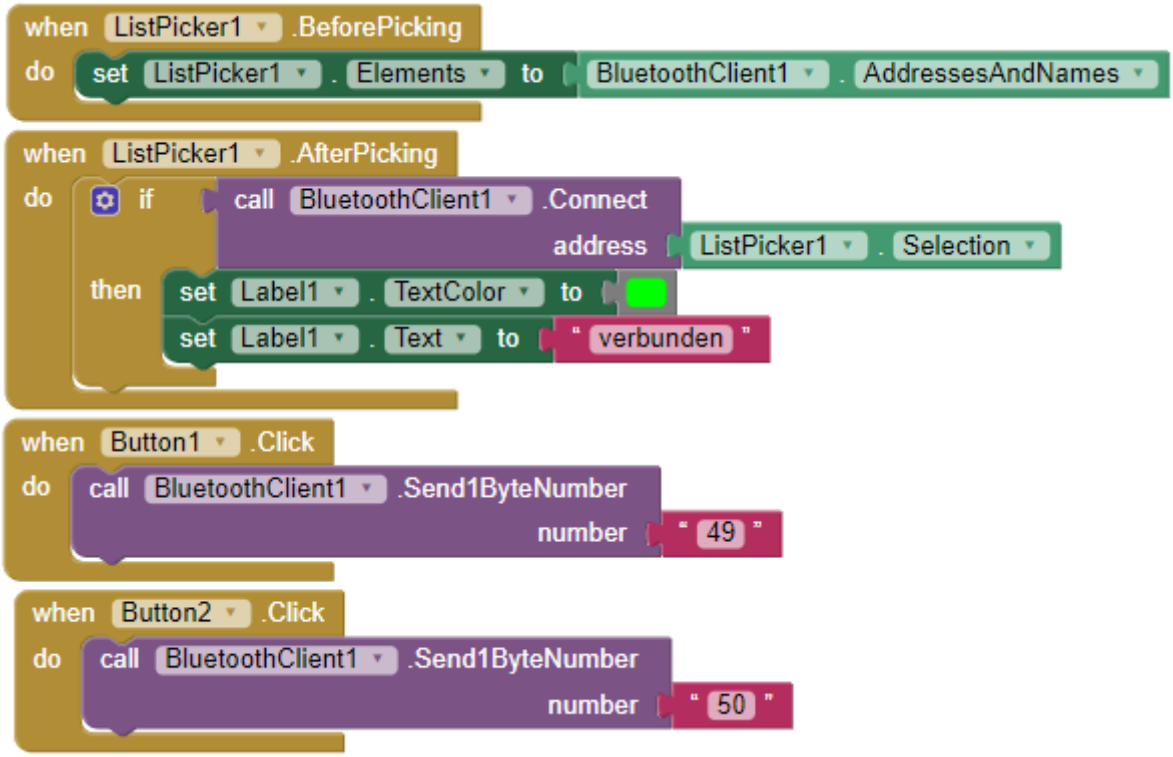
… und danach den Text von „Label1“ …



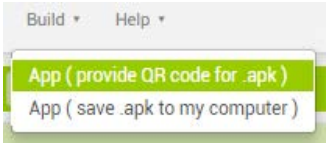
auf „verbunden“ ändern (Baustein erreichbar über „Text“. Das Feld zwischen den Anführungszeichen entsprechend beschriften).



Schließlich wird jeder „Button“ die der Funktion ausgestattet, bei einem Klick eine bestimmte Zahl über den Bluetooth Client zu übermitteln:



„Button1“ soll die Zahl 49 (Taste mit der Ziffer 1), „Button2“ die Zahl 50 (Taste mit der Ziffer 2) übermitteln.



Um zu prüfen, ob die Programmierung funktioniert, kann über „Build“ die App auf zwei Wegen an das Handy übermittelt werden. Am einfachsten ist es, die Funktion „save.apk to my computer“ zu wählen, wodurch die Installationsdatei in den Downloadordner des eigenen Rechners kopiert wird. Von dort sendet man sie beispielsweise als Anhang an eine Email zum Handy. Ist dort die Installation von Fremdsoftware erlaubt, erfolgt die Installation der App nach der Freigabe ohne weiteres Zutun.

Um das Erscheinungsbild auf dem Handy optisch aufzuwerten, sollte nun – wie weiter oben bereits erwähnt - der „Screen“ und die darauf platzierten „Components“ über die dazugehörigen „Properties“ anders angeordnet und im Erscheinungsbild verändert werden, also den Screen in horizontaler Richtung zentrieren, den Abstand zwischen den einzelnen Komponenten (Label1, ListPicker1, Button1 und 2) durch Einfügen von leeren und farblosen Textfeldern („TextBox“) vergrößern und die berührungsempfindlichen Flächen („Button“) sinnstiftend beschriften und die Flächen auffällig einfärben (vgl. dazu die Abbildung oben).

Viel Erfolg!