

Programmieren C++: File-Aufteilung, dyn. angelegte Objekte: Fortsetzung Grafik

Klaus Kusche

Dieses Beispiel ist eine Fortsetzung des vorigen Beispiels, nimm deine Lösung (oder meine Musterlösung) des vorigen Schrittes als Ausgangsbasis!

1. Trenne dein Programm sauber in **.h-Files und .cpp-Files** auf:

- Für **Color** reicht ein **.h-File**, ein **.cpp-File** ist nicht nötig (Warum?).
- Für die Rechteck-Klasse brauchst du beide Files.
- Und das **main** bekommt auch einen eigenen **.cpp-File**.

Vergiss nicht auf den Schutz gegen mehrfaches Include in deinen **.h-Files** und überlege genau, wo welche #include hingehören (und wo nicht)!)

2. Wir wollen beliebig viele Objekte herumfliegen lassen (deklariere eine Konstante für die Objekt-Anzahl, damit diese leicht geändert werden kann).

Da es nicht möglich ist, ein Array von Rechtecken anzulegen (Warum?), legen wir ein Array von Pointern auf Rechtecke in der nötigen Größe an.

Dann befüllen wir dieses Array in einer Schleife mit dynamisch angelegten Rechtecken (ich habe sie in die Fenstermitte gesetzt und ihnen eine zufällige Farbe, eine zufällige Größe bis **30**, und eine zufällige Geschwindigkeit -10 ... 10 gegeben).

Die eigentliche Hauptschleife habe ich ebenfalls umgebaut:

- Die äußere Schleife läuft, solange es noch Objekte gibt, und enthält das **sdlMilliSleep**, das **sdlUpdate**, und eine Schleife über alle Elemente des Arrays.
- Die innere Schleife prüft, ob das Array-Element ein **nullptr** ist (in diesem Fall tut sie nichts). Wenn nicht, ruft sie für das betreffende Objekt **fly** auf.
- Wenn das **fly** von der Wand abgeprallt ist, wird das Rechteck wie im vorigen Beispiel um 10 % verkleinert und geprüft, ob dadurch die Breite oder Höhe 0 geworden ist. Wenn ja, habe ich das Objekt vernichtet, einen **nullptr** im Array-Element gespeichert, und den Zähler der “lebenden” Objekte erniedrigt.