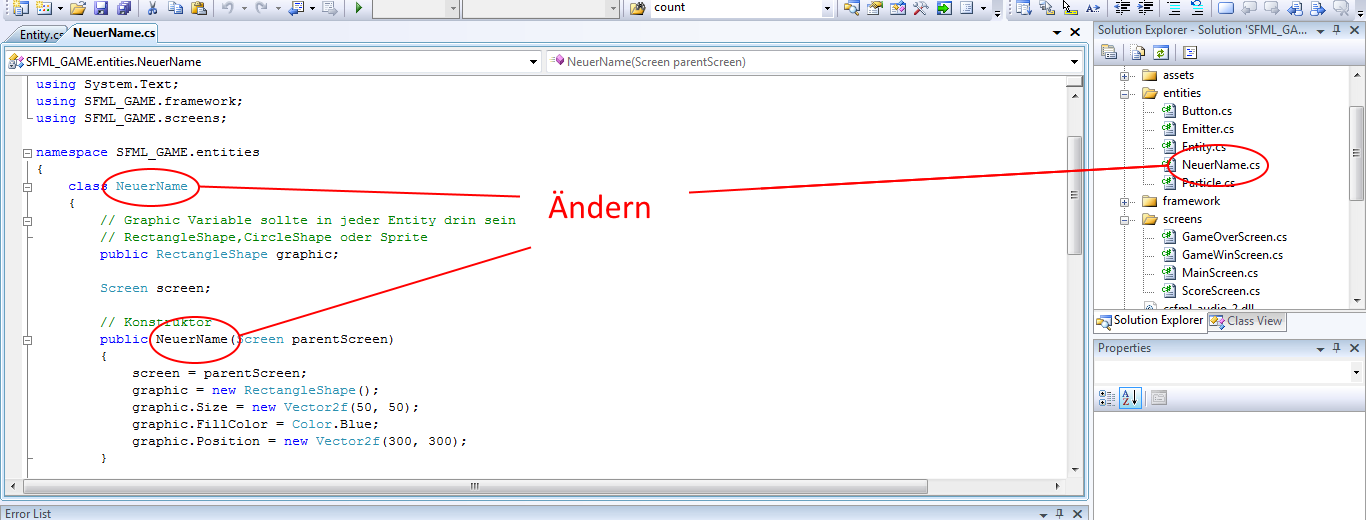
#10.2 Spieleobjekte(2)

Wie schon vorher schreiben wir neue Entities nicht komplett neu, sondern kopieren uns die "Entity.cs" im Ordner "entities". Sie dient als Vorlage für alle weiteren Entities um Zeit und Nerven zu sparen.  
  
Wir ändern die **Datei**, den Namen der **Klasse** und den Namen des **Konstruktors** alle zu dem Namen unserer neuen Entity.  
Außerdem sollte die Datei im **Ordner** "entities" liegen.  


In der Entity-Klasse ist Beispielhaft schon ein RectangleShape als Grafik. Dieses kann mit einem oder mehreren Shapes/Sprites ersetzt werden.

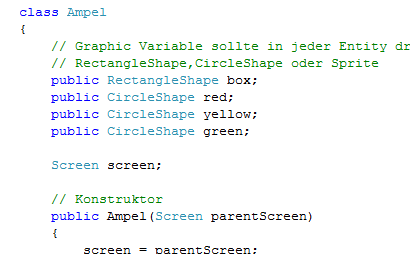
Beispielhaft wollen wir hier eine Ampel machen:

stop ready go

Die Ampel soll immer eine der 3 Zustände haben und zwischen ihnen in der angezeigten Reihenfolge wechseln. Das Wechseln soll über Tastendrücke ausgelöst werden.

Als erstes erstellen wir eine neue Entity Klasse, also kopieren wir Entity.cs, ändern in der Kopie den Dateinamen zu "Ampel.cs", den Klassen und Konstruktornamen zu "Ampel".(siehe Oben)

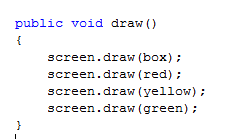
Als nächstes brauche wir die Shapes. Man kann sehen, dass die Ampel aus einem RectangleShape und 3 CircleShapes besteht. Also erstellen wir diese:

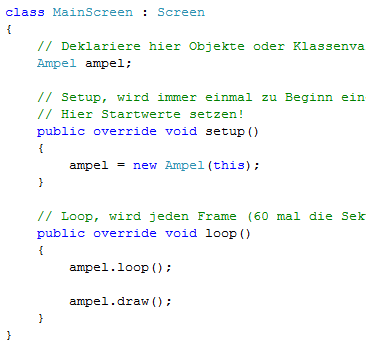


In unserem Konstruktor müssen wir diese natürlich erstmal erstellen.



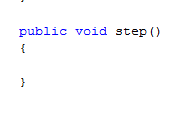
Damit wir unsere Shapes später auch sehen können, müssen diese in DrawAll() auch gezeichnet werden.



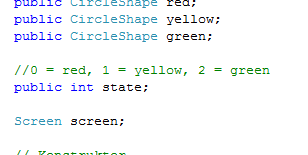
Jetzt können wir in der MainScreen unsere Entity einbauen. 

Was jetzt noch fehlt, ist das Schalten der Ampel.  
Dazu schreiben wir eine "step" Methode **in die Ampel Entity.**

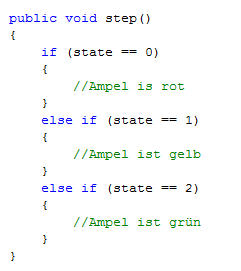
Diese soll den Zustand immer einen Schritt weiterwechseln.



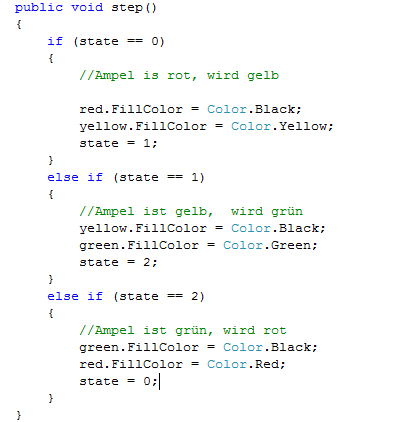
Bevor wir den Inhalt unserer Step-Methode schreiben, wollen wir vorher eine Variable schreiben, die den Zustand der Ampel speichert. Dazu erstellen wir ein int-Variable mit den Namen "state". Wobei state 0 ist, wenn die Ampel rot, 1 bei gelb und 2 bei grün.



In Step müssen wir durch If-Abfragen jetzt überprüfen, welchen Zustand die Ampel im Moment hat.



Jetzt müssen wir den Zustandswechsel der Ampel programmieren, in dem wir den entsprechenden nächsten Zustand einstellen. State sollte auch auf den neuen Zustand gesetzt werden.



Somit ist unsere Ampel schon fertig.

Jetzt müssen wir in der Mainscreen nurnoch step() aufrufen, immer wenn wir Space drücken.



