

Entwurfsentscheidungen für die Grafik-App BrainPool

Paketstruktur

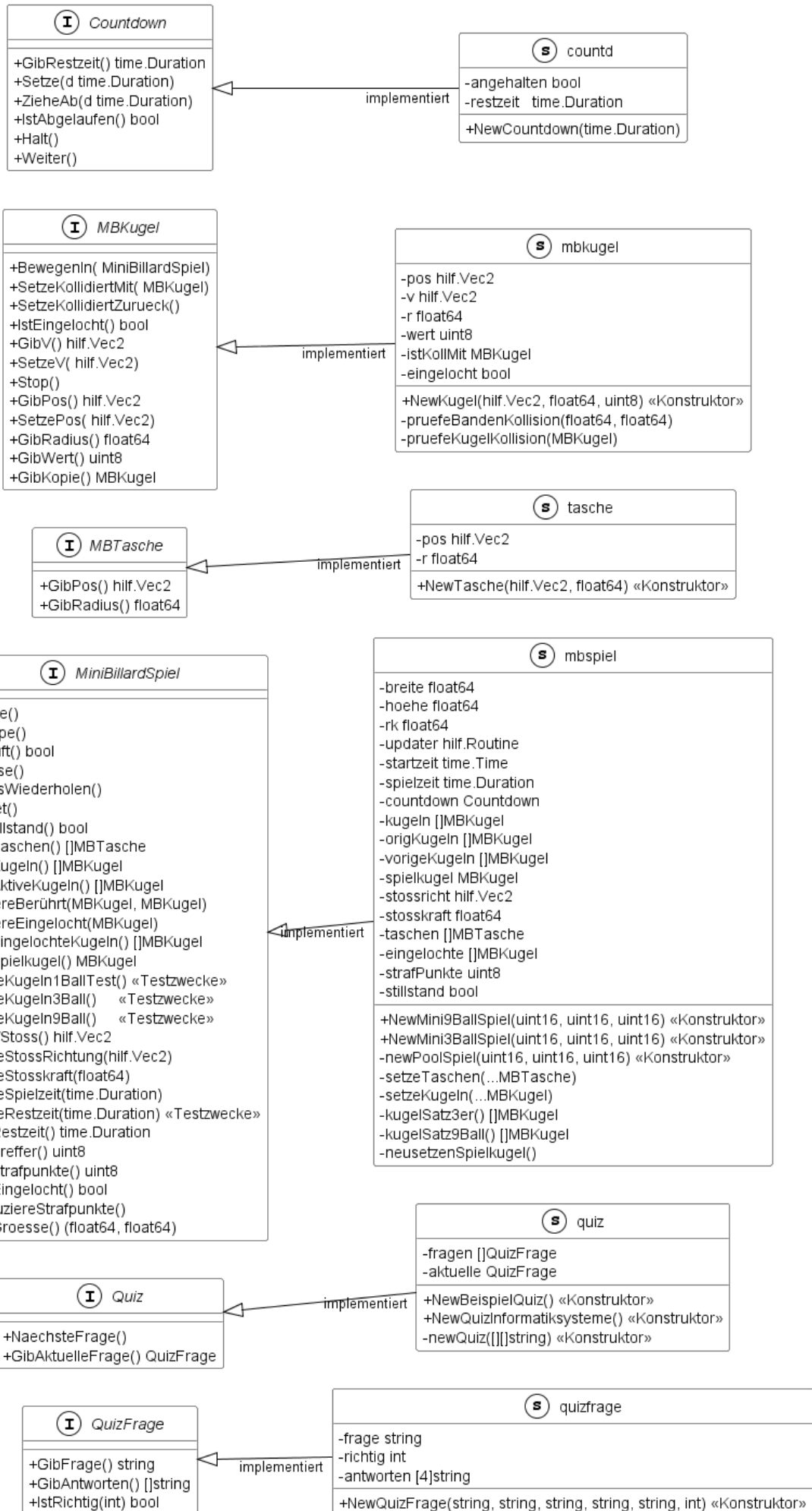
apps	das grundlegende Interface für eine App, eine Runner-Funktion und Hilfsroutinen für die Steuerung; fassen die gesamten Laufzeitelemente einer App zusammen. Hier liegen auch: eine Beispiel-App und die BrainPool-App selbst.
modelle	Simulationenobjekte und Quizzes; die "Logik"
views_controls	Die gesamte Darstellung/Das App-Layout über eine stark spezialisierte und gut informierte Widget-Hierarchie.
assets	Fonts, Klangdateien und nachladbare Quizfragen im Dateisystem
klaenge	spezialisierte Klassen zum Abspielen von Klangdateien aus den assets
hilf	Klassen zur Steuerung von Routinen und für die Vektorrechnung

Laufzeitverhalten

Die BrainPool App basiert auf gfx. Aufgrund des starken Spielcharakters mit vielen bewegten Elementen läuft die App in einem unmittelbaren Modus, bei dem in einer angemessen schnellen Wiederholung der gesamte Widget-Stapel mit allen Änderungen von unten nach oben neu gezeichnet wird. Das kommt der Struktur von gfx sehr entgegen und erleichtert die Implementierung und die Steuerung.

Die Zeichenschleife (Darstellung, App-Methode `Zeichne()`) ist dabei von der Spielschleife (Logik, Moduswechsel, App-Methode `Update()`) getrennt. Zudem wurde eine weitere nebenläufige Schleife für die Simulation des Billard-Spiels eingeführt, da deren Wiederholrate in der gegenwärtigen Implementierung grundlegend ist für das Geschwindigkeitsverhalten der Billard-Kugeln. Die Simulation wird komplett vom Modell gesteuert. Hier wäre zukünftig eine Steuerung über die `Update()`-Methode wünschenswert, um die Programmierung transparenter zu machen ...

Die Tastatur- und die Maussteuerung laufen in eigenen Abfrage-Schleifen und ändern den Zustand des Modells jeweils über Ihnen zugewiesene Methoden aus dem App-Interface. Dies erleichtert die Übersicht, da der Code für die Steuerung von der restlichen App getrennt ist.



views_controls

« nur Views und Zeichen-Helfer »

