



5Code als Beispiel für eine best practice in E-Learning

Ein Schwerpunkt der studienunterstützenden Angebote des Fachbereichs liegt auf der frühen Studienphase, also den ersten beiden Semestern. Hier zeigt sich auf der Grundlage statistischer Daten, dass viele Studierende Prüfungen in den Grundlagenfächern hinauszögern und damit den planmäßigen Studienverlauf verlassen. Neben durchgängigen Mathematik-Sprechstunden mit Hilfe von QVM werden daher u.a. Maßnahmen finanziert, die auf die Entwicklung besserer Lernumgebungen abzielen.

Im Rahmen des von Studierenden mit erstellten und gewarteten Projekts 5Code ist im Berichtszeitraum eine integrierte Entwicklungsumgebung entstanden, die speziell Programmieranfängern in den ersten Semestern schnellere Lernerfolge ermöglicht. Das Ergebnis wurde auf der E-Learning Fachtagung Informatik DeLFI 2015 mit dem Best Paper Award ausgezeichnet.

Zunächst wurde eine einfach verständliche Darstellung erarbeitet, wie man in fünf Schritten vom Problem zum Programm kommt: Lesen > Verstehen > Überlegen > Aufschreiben > Codieren.

Um die kognitive Belastung der Lernenden bei der Bearbeitung dieser fünf Schritte wirksam zu vermindern und so den Lernerfolg zu erhöhen wurde die Entwicklungsumgebung 5Code entwickelt, die über alle fünf Schritte den gesamten Kontext integriert – von der Aufgabenstellung über eigene Notizen bis zur Codierung.

Zur Unterstützung des Verstehens können Aufgabenteile markiert und mit eigenen Überlegungen annotiert werden. Diese Notizen können in den Code übernommen und synchronisiert werden. Durch die Integration aller Schritte in einer einzigen Anwendung kann der **Flow** beim Lösen des Problems erhalten bleiben. 5Code wurde als Web-Applikation implementiert und wurde in mehreren Hochschul-Programmierpraktika evaluiert.

Mehr Informationen finden Sie auf der Webseite des Projekts:

24.03.2017

5Code als Beispiel für eine best
practice in E-Learning

<https://medien.hs-duesseldorf.de/personen/dahm/Projekte/Seiten/5Code.aspx>

Dort finden Sie neben Vortragsfolien und Zeitschriften-Artikeln in Deutsch und Englisch zum Projekt auch einen Demo-Zugang, mit dem man 5Code selber ausprobieren kann.

Bei Fragen können Sie sich an den Projektleiter Prof. Dr. Markus Dahm wenden: markus.dahm@hs-duesseldorf.de

Ein Beispiel-Screenshot:

The screenshot displays the 5Code web application interface. The top navigation bar includes the 5Code logo and tabs for 'Projekt', 'Bearbeiten', 'Tools', 'Sprache', 'Info', 'OOP1 - 13 - Abschlussaufgabe v14demo', 'Markus Dahm', and 'Offline'. Below this, a sub-navigation bar shows 'Standard', 'Erweitert', and 'Markierung' (selected). The main content area is divided into two panels. The left panel, titled 'Aufgabe 1 Kollisionen testen und löschen', contains text describing a game task and a list of questions. The right panel, titled 'Position inklusive Drehung', contains a code editor with two code snippets. The first snippet is for the 'Block' class, and the second is for the 'Spielfeld' class. The code editor also shows a 'Navi' (Navigation) pane on the right side.

Das Spiel 5Tris stellen wir nun soweit fertig, dass man es auf der Konsole spielen kann.

Aufgabe 1 Kollisionen testen und löschen

Im Spiel soll ein Block in unterschiedlichen Aktionen bewegt und gedreht werden. Dabei kann es zu einer Kollision kommen.

Eine Kollision tritt dann auf, wenn mindestens ein übertragbares Zeichen eines Blocks auf ein bereits vorhandenes übertragbares Zeichen im Spielfeld übertragen würde. Beispiele: (' ' steht dabei für ein Leerzeichen):

'z' trifft auf ' ' ergibt keine Kollision, 'z' trifft auf 'o' ergibt eine Kollision, ' ' trifft auf 'o' ergibt keine Kollision

Das Testen auf Kollision für einen Block im Spielfeld ist nicht ganz einfach, da mehrere Parameter berücksichtigt werden müssen

- ist das betrachtete Zeichen übertragbar?
- wohin ist der Block im Spielfeld verschoben?
- wie ist der Block im Spielfeld gedreht?

Alle diese Parameter sind bereits in einer Methodenstruktur berücksichtigt worden: In den Methoden, die alle Pixel eines Blocks mit diesen Randbedingungen in das Spielfeld abbildet. Dabei gehören diese Methoden verschiedenen Klassen an, je nachdem, auf welche Variablen sie zugreifen dürfen.

a) Welche Methoden sind das?

Überlegen Sie erst selber, kontrollieren Sie dann Ihre Antwort hier.

Position inklusive Drehung

Da ein Block auch nicht-quadratisch sein kann, müssen wir noch an einer Stelle eine Methode ergänzen: Die Methode, die feststellt, ob die **aktuelle Position gültig** ist, hatten wir im letzten Termin erweitert auf den **Test der Rand-Koordinaten**. Nun müssen wir noch mit einbeziehen, dass der Block auch **gedreht** sein kann. Bei einem *nicht-quadratischen* Block ändert sich dadurch was?

```
///                                     Klasse: Block
1 /** Methoden zur Detektion von Kollisionen in der Klasse Block
2  * Finde Kollisionen aller Pixel im Block b
3  */
4 boolean kollisionGefunden(){
5     // Finde Kollisionen aller Pixel im Block b
6 }

///                                     Klasse: Spielfeld
1
2 /** Methoden zur Detektion von Kollisionen in der Klasse Spielfeld
3  * Finde Kollisionen an der Stelle x,y i Spielfeld
4  */
5 boolean kollisionGefunden(int x, int y){
6     // Finde Kollisionen an der Stelle x,y im Spielfeld
7 }
```