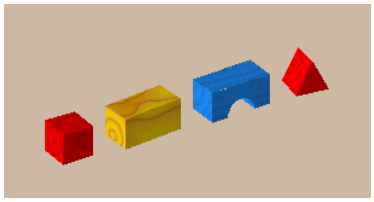
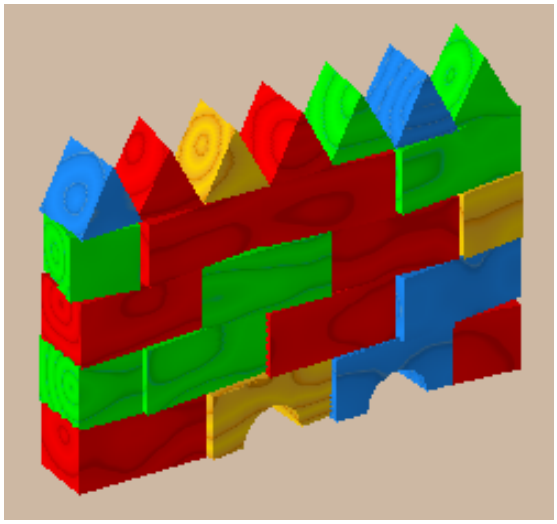
Vorlesung Softwaretest und -debugging

Dr. Carsten Gnörlich cg@techfak.uni-bielefeld.de

Übungsblatt 3 11. Mai. 2012



Zum Aufbau einer Mauer stehen die rechts gezeigten Bausteine zur Verfügung. Dies sind von links nach rechts halbe Klötze (half), volle Klötze (full), Bogenelemente (arc) sowie Dachelemente (roof). Die Bauelemente vom Typ half und roof haben die Größe 1x1x1; die übrigen beiden die Größe 2x1x1, sind also doppelt so breit wie hoch bzw. tief.



Mit Hilfe dieser Steine soll eine Mauer derart gebaut werden, daß übereinanderliegende Bausteine vom Typ „full“ jeweils um die Hälfte überlappen und somit einen stabilen Verband bilden. Außerdem soll gelten:

- Die oberste Reihe soll nur aus Bausteinen vom Typ „roof“ bestehen
- Eine Mauer der Höhe 1 besteht nur aus der obersten Reihe
- Die unterste Reihe soll aus Bausteinen vom Typ „arc“ bestehen, aber am Rand soll jeweils ein Klotz vom Typ „full“ oder „half“ stehen
- Der am weitesten links unten stehende Baustein ist immer vom Typ „full“.

Aufgabe 1

Vervollständigen Sie die Funktion *render_them()* im Programmgerüst „wall.c“ um eine Mauer für den Wertebereich 0..16 für Höhe und Breite zu erzeugen (der Wertebereich wird bereits im Hauptprogramm geprüft). Zum Darstellen der Bausteine sind die Funktionen *put_half(), put_full(), put_arc()* und *put_roof()* vorgegeben. Sie plazieren die Bausteinen innerhalb eines (x,y)--Koordinatensystems, wobei x und y Vielfache der Größe des halben Klotzes sind.

Beispielaufruf: *wall 5 7* würde die obige Mauer der Höhe 5 und Breite 7 erzeugen.

Aufgabe 2

a) Geben Sie den Kontrollflußgraphen für die von Ihnen implementierte Funktion *render_them()* an.
b) Finden Sie einen EE-Weg, der einen Vereinigungsknoten enthält. Geben Sie die Segmente dieses EE-Weges an.

Aufgabe 3

Finden Sie eine passende Belegung für *x* und *y* so daß der Aufruf *wall x y* einen Wert von $TWM_1 = 100\%$ erreicht.
Weisen Sie dies empirisch durch Instrumentierung des Programms nach.

Aufgabe 4

Geben Sie eine Testmenge für Aufrufe von *wall x y* an, so daß $C_i(2)$-Überdeckung (starke C_{GI}-Überdeckung) erreicht wird.

Aufgabe 5

Geben Sie alle sensitivierenden Testfälle für den Entscheidungsausdruck an, mit dem Sie ermitteln ob ein Baustein vom Typ „Full“ oder „Arc“ in der untersten Reihe plaziert wird. Ist es möglich diese Fälle direkt über einen Aufruf des wall-Programmes zu generieren? Wie läßt sich dieses Problem gegebenfalls lösen (nur beschreiben, nicht machen :-)?