

Gebt bitte die Bearbeitung als .pdf oder .doc Datei an Tobias Hammer (hammer@informatik.uni-bremen.de) ab. Bitte stellt die Abgaben zu allen Aufgaben in einer Datei zusammen, die wir drucken und korrigieren können. Der Dateiname soll die Nummer des Übungszettels und der Gruppe beinhalten, z.B. `ifguebung507.pdf` für Übung Nr.5 Gruppe 7. Gebt Namen und Email aller Gruppenmitglieder an. Gebt ausserdem die Zeit (Maximum aller Gruppenmitglieder) an, die Ihr für den Zettel benötigt habt. Diese Angabe ist nur als Rückmeldung für uns und geht nicht in die Bewertung ein.

Aufgabe 16 Blockwelten (10 + 5 Kreativpunkte)

Dieses mal sollt ihr wieder ein eigenes kleines Spiel programmieren. Mittlerweile hab ihr so viele Grundlagen gelernt, dass ihr immer komplexere und anspruchsvollere Spiele erstellen könnt. Jetzt ist es an der Zeit ein Spiel mit einem Level zu erstellen. Nicht in 3D sondern ganz klassisch in 2D und auf Blöcken basierend. Viele der älteren Spiele verwendeten diese Technik, z.B. Klassiker wie Pacman, Super Mario Land, usw.

Damit ihr es etwas einfacher habt haben wir euch die Klasse `Map` vorgefertigt und ein kleines Demoprogramm erstellt. Darin könnt ihr praktisch sehen, wie man die Klasse verwenden kann. Die Klasse macht ein paar Annahmen, wie ein Level aufgebaut sein muss und woher die Grafiken kommen:

- Eine Levelbeschreibung ist eine einfache Textdatei.
- In der Datei sind Zeilen mit Buchstaben von A-Z, wobei jede Zeile exakt gleich lang sein muss.
- Jeder Buchstabe in der Datei steht für einen Block, der nachher auf dem Bild sichtbar ist. Die Bedeutung der Buchstaben ist der Map-Klasse egal, aber natürlich für euer eigenes Programm wichtig.
- Um die Blöcke auf den Bildschirm zu zeichnen, lädt die Map-Klasse zu jedem Blocktyp ein Bild mit dem Namen `Buchstabe.png`. Wenn eure Map-Datei nur aus 'A' und 'B' besteht müssen folglich die Dateien `A.png` und `B.png` existieren. Die Bilder sollten, wenn möglich alle gleich groß und quadratisch sein.
- Bilder müssen alle im Unterordner `images/` liegen, sonst gibt es einen Fehler.
- Da es sein kann, dass eure Blockwelt nicht immer den gesamten Bildschirm ausfüllt, kann man zusätzlich ein Bild hinterlegen, das für alle Blöcke außerhalb der in der Textdatei definierten Welt verwendet wird. Es hat den Namen `_.png`. Wenn es nicht vorhanden ist (wie im Demospiel), wird einfach nur die definierte Welt gezeichnet.

Aufgabe ist es nun ein Spiel in so einer Blockwelt zu kreieren. Die Spieleranzahl ist euch überlassen. Erstellt dafür ein Processing-Programm, die benötigten map-Dateien und sucht oder erstellt euch passende Bilder. Das Spiel sollte außer dem Map-Objekt als Spielwelt mindestens noch zwei Typen von Figuren (z.B. Spielfigur und Gegner) beinhalten. Von mindestens einem Figurtyp (z.B. dem Gegner) sollen mehrere Objekte im Spiel vorkommen. Die Anzahl dieser Objekte soll sich im Spielverlauf ändern.

Programmtechnisch soll für den Level die vorgegebene Map Klasse verwendet werden. Die Figurtypen sollen über Klassen realisiert sein. Achtet auf eine sinnvolle Aufteilung der Funktionalität zwischen Hauptprogramm und Klassen. Eine Klasse sollte möglichst eine abgeschlossene Aufgabe haben und nichts oder möglichst wenig vom Rest des Programms “wissen” müssen. Der Figurtyp von dem es variabel viele gibt soll mit einem Array realisiert sein in das Objekte eingefügt und gelöscht werden.

Gebt bei der Abgabe der Lösung alle Dateien ab, die das Spiel benötigt, also auch die Bilder, map-Dateien, ggf. Soundeffekte oder was immer nötig ist, um das Spiel zu starten. Bitte entweder direkt an die Mail anhängen oder einfach alles als zip-Archiv packen und anhängen.

Aufgabe 17 Wozu ist der doppelte Saugheber gut? (2 Bonuspunkte)

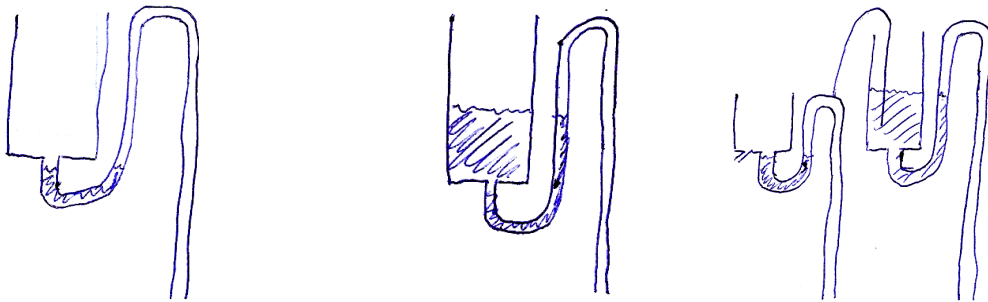


Abbildung 1: Saugheber als zentrales Element des Wasserschwall-Computers. *Links*: Der leere Saugheber stellt eine 0 dar. *Rechts*: Wasser im Saugheber stellt eine 1 dar. Fließt ein weiterer Schwall Wasser zu, übersteigt der Pegel das obere Ende des U-Rohrs. Durch den Saugeffekt wird der ganze Behälter leer gesaugt ($1 \rightarrow 0$). *Rechts*: Eine Kombination aus zwei Saughebern. Der linke Saugheber hat einen kleinen Ablauf und ein niedrigeres U-Rohr. Das obere Ende des U-Rohrs ist mit einem dünnen Rohr verbunden, dessen Ende im Behälter des rechten Saughebers endet (Bonusaufgabe).

Man kann das technologische Gedankenspiel aus Aufgabe 15 weiter bis zu einer Differenzmaschine treiben. In dieser Maschine treten viele Saugheber als Paar auf (Abb.1, rechts). Was tun die Saugheber Paare? Wozu sind sie wohl nötig?