

# Übungsblatt 5

## Interaktive Systeme – SoSe 24

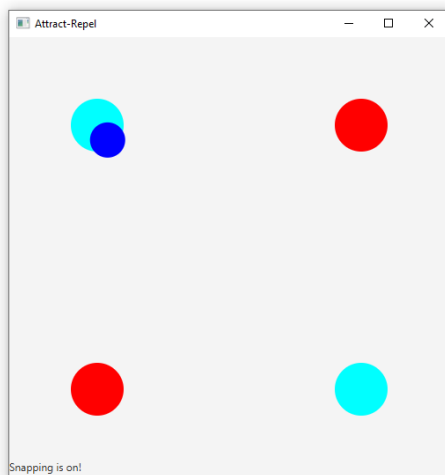
Prof. Dr. Michael Rohs, Jan Feuchter, M.Sc.

Alle Übungen, die nicht explizit als Gruppenaufgabe deklariert sind, müssen in Einzelarbeit geleistet und individuell abgegeben werden. Identische Abgaben werden als Plagiat behandelt.

Abgabe bis Montag den 06.05. um 23:59 Uhr über <https://assignments.hci.uni-hannover.de/SoSe2024/ISy>. Die Abgabe muss aus einer einzelnen zip-Datei bestehen, die alle nötigen Dateien enthält. Lösen Sie Umlaute in Dateinamen bitte auf.

Verpacken Sie Ihre Lösung (pdf-Datei und zip-Datei des exportierten Projekts) zum Hochladen in eine zip-Datei.

### Aufgabe 1: Semantic Snapping (15 Punkte)



Beim Semantic Snapping können Ziele („snap sites“) ein gezogenes Objekt anziehen oder abstoßen, abhängig davon, ob die Positionierung in der Semantik der Anwendung sinnvoll ist oder nicht. Implementieren Sie unter Verwendung des Templates `SemanticSnap.zip` (siehe Stud.IP) folgendes Verhalten: Es gibt anziehende Ziele (attracting targets) in Hellblau und abstoßende Ziele (repelling targets) in Rot. Weiterhin gibt es ein dunkelblaues Objekt, das sich mit dem Mauscursor mitbewegt (wenn es nicht durch snapping angezogen bzw. abgestoßen wird). Für die anziehenden Ziele soll symmetrisches Snapping (Folien: „Snapping with Symmetrical Catch-Up Region“) implementiert werden. Für abstoßende Ziele soll das dunkelblaue Objekt das jeweilige Ziel nicht überfahren, sondern an dessen Rand bleiben. Dieses Verhalten ist im Video `SemanticSnap.mov` (siehe Stud.IP) illustriert.

- Implementieren Sie die Methode `snap` in der Klasse `RepellingTarget`. Der dunkelblaue Kreis darf sich nicht mit dem roten Ziel überschneiden.
- Implementieren Sie die Methode `snap` in der Klasse `AttractingTarget`. Das Verhalten soll so sein, wie im Video illustriert (symmetrisches Snapping). Wenn sich der dunkelblaue Kreis (1) und der hellblaue Kreis (2) nicht überschneiden bleibt (1) zentriert auf der Position des Mauscursors. Ist der Mauscursor innerhalb von (2), wird (1) in (2) zentriert. Ansonsten ist die symmetrische run-ahead bzw. catch-up Region aktiv.

- c) Implementieren Sie das Ein- bzw. Ausschalten des Snappings. Dies soll mit einem Tastendruck auf die Taste `s` möglich sein. Der Zustand soll in einem Label dargestellt werden. Nutzen Sie dafür Bindings.

Hinweis: Es ist hilfreich, die Methoden der Klasse `Point2D` zu verwenden (z.B. `distance`, `add`, `subtract`, `multiply`, `normalize`).

## Aufgabe 2: Pointing Techniques – BubbleCursor (8 Punkte)

In der Vorlesung wurde der BubbleCursor besprochen. Weitere Informationen finden Sie im nachfolgenden Paper.

Tovi Grossman and Ravin Balakrishnan. 2005. The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 05). ACM, New York, NY, USA, 281-290.

<http://www.tovigrossman.com/BubbleCursor/>

- a) Beschreiben Sie ein Szenario für das der Bubble Cursor besonders gut geeignet ist. Begründen Sie kurz, warum dies der Fall ist.
- b) Beschreiben Sie ein Szenario für das der Bubble Cursor weniger gut geeignet ist. Begründen Sie kurz, warum dies der Fall ist.