

## Übungsblatt 2

### Makefile, Grafikausgabe

(Ausgabe 17.04.2014)

---

#### 1. Aufgabe *Grafikausgabe im X-Fenster und einfaches Makefile* ★★

Um unsere Simulationen nachher grafisch als Animationen darstellen zu können, werden wir eine X11-basierte Bibliothek namens **Xgraphics**<sup>1</sup> verwenden.

- a) Versuchen Sie das im Abschnitt 5 der Xgraphics-Anleitung gezeigte *erste* Demo-Programm zum Laufen zu bringen. Benutzen Sie aber statt `my...` ein anderes Präfix (z.B. `our`).

Versuchen Sie mithilfe der Anleitung die wesentlichen Elemente zu verstehen.

Zum Kompilieren und Linken können Sie folgende Syntax verwenden:

```
g++ -c Xgraphics.c
g++ -o program Xgraphics.o program.cpp -lX11
```

- b) Schreiben Sie ein **Makefile**, dass die o.g. Befehle beim Aufruf von `make` ausführt. Implementieren Sie auch `make clean`.

Wichtig: Um sinnvoll und effizient mit `make` arbeiten zu können, sollten Sie sich angewöhnen, jedes *Projekt* in einem separaten Verzeichnis abzulegen.

#### 2. Aufgabe *Einfaches Modell einer rotierenden Galaxie* ★★

Wir wollen uns nun veranschaulichen, warum sich die Spiralstruktur einer Spiralgalaxie nicht mit derselben Geschwindigkeit wie die um das Zentrum der Galaxie rotierende Materie (also z.B. Sterne) bewegen kann.

- a) Erstellen Sie mithilfe von **Xgraphics** ein Programm, das eine rotierende Galaxie aus Punkten/kleinen Kreisen als Animation darstellt (siehe Abb. 1). Nehmen Sie z.B. einen Radius der galaktischen Scheibe von `int nring = 20`, und setzen Sie bei jedem Radiusring `6 * i` Punkte (`i <= nring`).

Erzeugen Sie hierfür in einem hinreichend großem Window (globale Variable, z.B.  $600 \times 600$  Pixel) eine **World** (globale Variable). Beachten Sie, dass **World** ein kartesisches Koordinatensystem verwendet, die polaren Koordinaten des Galaxienmodells also immer erst umgerechnet werden müssen.

---

<sup>1</sup>Sie finden **Xgraphics.h**, **Xgraphics.c** und die Anleitung **Xgraphics.pdf** unter `htodt`.

- b) Galaxien rotieren nicht als starre Körper (also  $\omega \neq \text{const.}$ ), sondern wegen der Materieverteilung mit  $v_{\text{circ}} \approx \text{const.}$
- c) Um die “Zerstörung” eines initialen Spiralmusters durch differentielle Rotation zu untersuchen, können Sie z.B. jeden  $i$ -ten Punkt bei einem Radius  $i$  mit rot einfärben.
- d) Schreiben Sie nun eine **event**-Loop für die Animation analog zu den Beispielen in Abschnitt 5 der **Xgraphics**-Anleitung, allerdings mit dem Wert 0 für den Parameter **waitflag** in **GetEvent()**. D.h. die Loop läuft auch dann weiter, wenn keine Taste gedrückt wird, soll aber beendet werden (**done = 1 ;**), sobald eine Taste gedrückt wird. Bei jedem Schleifendurchlauf wird die Zeit **t** um **dt** hochgezählt und alle Punkte bei einem Radius **i** bewegen sich entsprechend dem Bewegungsgesetz für die gleichförmige Rotation weiter. Teilen Sie Ihr Programm hierfür in sinnvolle Funktionen auf.  
*Wichtig:* Um eine vernünftige Animation zu ermöglichen, sollte das Programm nach jedem Zeichnen eine kurze Pause einlegen, z.B. mittels **system ("sleep 0.1") ;** und vor jedem Zeichnen, mittels **ClearWorld(ourworld) ;** den Zeichenbereich löschen.
- e) Testen Sie auch ein Keplersches Geschwindigkeitsgesetz und den Fall  $\omega = \text{const.}$
- f) *Zusatz:* Sie können Ihr **Window**, in dem Sie zeichnen, auch noch mit **Buttons** oder **KeyPress**-Events versehen um mit entsprechenden Tastendrücken die laufende Simulation zu beeinflussen. Z.B. sollten mittels Taste **+** die Zeitschritte größer gemacht werden können.

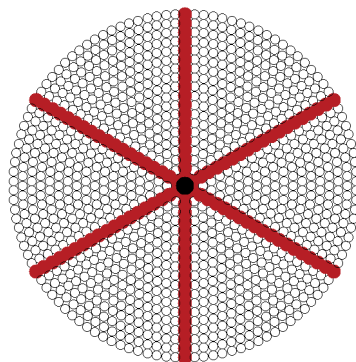


Abbildung 1: Rotationsmodell einer Galaxie mit initialem Muster