

Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel
Tim Schulte, Thorsten Engesser
Wintersemester 2017/2018

Universität Freiburg
Institut für Informatik

Übungsblatt 10

Abgabe: Freitag, 12. Januar 2018, 20:00 Uhr

WICHTIGE HINWEISE: Zur Bearbeitung der Übungsaufgaben legen Sie bitte ein neues Unterverzeichnis `sheet10` im Wurzelverzeichnis Ihrer Arbeitskopie des SVN-Repositories an. Ihre Lösungen werden dann in Dateien in diesem Unterverzeichnis erwartet. Beachten Sie bitte bei allen Aufgaben die *Hinweise zur Bearbeitung der Übungsaufgaben* unter der folgenden URL:

<http://gki.informatik.uni-freiburg.de/teaching/ws1718/info1/guide/hinweise.html>

Bewertet wird bei allen Aufgaben die letzte Version, die zur Deadline des Übungsblattes auf dem SVN-Server eingereicht ist.

Aufgabe 10.1 (Ethische Reflektion; Datei: `ethics.txt`; Punkte 5)

In dieser Aufgabe sollen Sie eine fiktive Situation unter ethischen Gesichtspunkten beurteilen. Wichtig ist, dass Sie Ihre Beurteilung ausreichend begründen. Ziehen Sie dazu die Leitlinien des VDI, die Leitlinien der GI oder den *Software Engineering Code of Ethics and Professional Practice* (<http://www.acm.org/about/se-code>) der ACM zu Rate.

Die IT-Abteilung des Weihnachtsmanns entwickelt eine Software, welche Geschenke klassifiziert und sie den Kindern mit übereinstimmenden Wunschzetteln zuweist. Weihnachten steht vor der Tür und die Software kurz vor ihrer Fertigstellung. Sie ist für den reibungslosen Weihnachtsbetrieb bereits fest eingeplant. Eine verzögerte Auslieferung der Software würde dazu führen, dass der geplante Geschenkedurchsatz reduziert werden müsste.

Trixi ist für die Qualitätssicherung zuständig, und kann auf jahrelange Erfahrung im Software-Testen zurückblicken. Obwohl die Software alle vom Weihnachtsmann angeordneten Tests bestanden hat, vermutet Trixi, dass die Software noch weiteren Tests unterzogen werden müsste um potentielle, unvorhergesehene Risiken ausschließen zu können. Jordi, der Leiter der IT-Abteilung, setzt Trixi unter Druck, die Software endlich freizugeben. Jordi erwähnt, dass der Weihnachtsmann (in der Regel gemütlich, freundlich und gerecht) durchblicken gelassen habe, dass er die gesamte Abteilung schließen werde, sollte die Software nicht rechtzeitig fertiggestellt werden. Trixi hält dagegen, dass ein Fehler in der Software weitreichende Folgen haben könnte. Kinder könnten vereinzelt unpassende Geschenke erhalten, was auch der Reputation des Weihnachtsmanns Schaden zufügen würde.

Wie sollte sich Trixi in der geschilderten Situation entscheiden?

Aufgabe 10.2 (Feuerwerk; Datei: `fireworks.pdf`, `fireworks.py`; Punkte 2+3+3+3+2)

2017 war ein Jahr mit Höhen und Tiefen. Zum Abschied wollen wir es noch einmal richtig krachen lassen. Um die Umwelt zu schonen (und nebenbei noch ein wenig Python zu lernen) natürlich rein virtuell. Angedacht ist ein Feuerwerk, dass jeden `tkinter.Canvas` zum

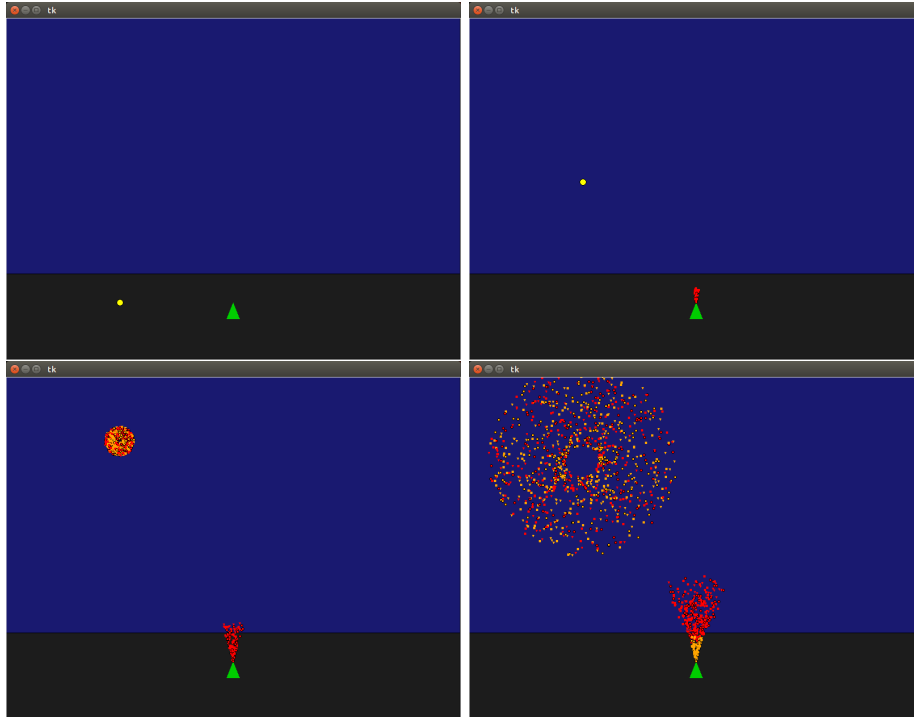


Abbildung 1: Funktionalität der `Rocket`-Klasse

Leuchten bringt. Um Ihnen ein wenig Arbeit abzunehmen, haben wir bereits einen Leuchtvulkan implementiert. Diesen finden Sie in der Datei `fireworks.py` auf der Vorlesungs-Website. Ihre Aufgabe besteht im Folgenden darin, das Modul um neue Klassen zu erweitern, damit weitere Typen von Feuerwerkskörpern erstellt und in die virtuelle Szenerie eingefügt werden können. Unter allen Lösungen wird eine Spezialjury das schönste und originellste Feuerwerk auswählen. Der Erschaffer darf fortan den Titel *Pythonista spectaculi pyrotechnici* tragen.

- (a) Machen Sie sich mit der Klassenhierarchie aus dem Modul `fireworks.py` vertraut. Zeichnen Sie anschließend das zugehörige Klassendiagramm (vgl. Foliensatz Nr. 17, Folie 26). Notieren Sie zu jeder Klasse zudem die neu definierten Attribute und Methoden.¹ Speichern Sie das fertige Diagramm in der Datei `fireworks.pdf` oder `fireworks.jpg`.
- (b) Leuchtvulkane haben zweifelsohne ihren eigenen Charme. Das Erreichen großer Höhen und das explosionsartige Hinterlassen leuchtender Partikel am Himmel gehören jedoch nicht dazu. Um Ihr Feuerwerk vielschichtiger zu gestalten, implementieren Sie nun eine Klasse `Rocket`. `Rocket`-Objekte sollen an einer beliebigen Stelle auf dem Canvas erzeugt werden können, mit einer bestimmten Geschwindigkeit aufsteigen, und nach einer bestimmten Flugzeit explodieren. Dabei soll ein beliebiges Muster aus Partikeln entstehen, welches sich nach einer vorgegebenen Lebensspanne wieder auflöst. Siehe Abbildung 1.
- (c) Implementieren Sie eine Klasse `RocketLauncher`. Instanzen dieser Klasse sollen in variablen Abständen `Rocket`-Objekte erzeugen und in den Himmel schießen. Variie-

¹Vererbte Attribute und Methode müssen nicht erneut angeführt werden

ren Sie dabei Abschusswinkel, Geschwindigkeit, und Explosion der `Rocket`-Objekte um den Effekt zu verstärken.

- (d) Überlegen Sie sich wenigstens einen weiteren Typ an Feuerwerkskörpern, und implementieren Sie diesen. Vergessen Sie nicht im Docstring die beabsichtigte Wirkung zu beschreiben.
- (e) Nutzen Sie Ihre neu erworbenen Pyro-Fähigkeiten zum Inszenieren des ultimativen Feuerwerks zum Jahreswechsel 2017/2018. Erstellen Sie dazu Instanzen der zuvor implementierten Feuerwerks-Klassen und zünden diese auf dem `tkinter.Canvas`.

Aufgabe 10.3 (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet10` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, benötigter Zeitaufwand nach Teilaufgabe, Bezug zur Vorlesung, Interessantes, etc.).