

Übungsblatt 9

Willkommen zum Praktikum zu Programmieren in Python.

Aufgabe 1.

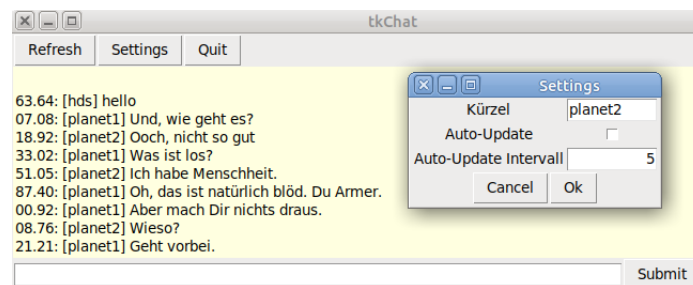
Bringen Sie das OO-GUI für den `gruss` aus der Vorlesung zum laufen als ein Programm `tkgruss.py`.



Hinweis 1. Für eine der nächsten Aufgaben bilden Sie Gruppen von 2-3 Teilnehmern. Jede Gruppe erhält ein eigenes Git-Repository mit dem Namen `19pythg[a-z]`. Sie machen ein kleines Projekt, um maximal weitere 10 Punkte zu erhalten. **Jeder** Gruppenteilnehmer soll sich einbringen und **muss** adäquat in der Git-Historie auftauchen. Bei Unklarheiten machen wir dazu Einzelgespräche am Rechner. Bewertet wird die Funktionalität (Umfang), Korrektheit und Eleganz der Lösung. Es muss nicht so aussehen wie in den Screenshots. Sie können mehr oder weniger oder andere Funktionalität realisieren. Der gezeigte Umfang entspricht der Erwartung.

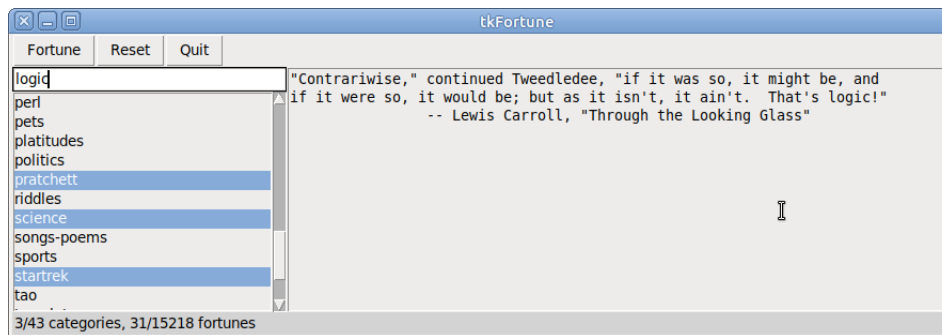
Hinweis 2. Gehen Sie bei der Realisierung der GUIs schrittweise vor. Fügen Sie immer eine Interaktion zum GUI hinzu und realisieren Sie und testen Sie diese. Wenn Sie merken, dass Sie Funktionalität wiederverwenden können, dann zögern Sie nicht diese auszulagern. Achten Sie darauf, dass das GUI auch gut skaliert. Dass also die Inhalte in den Fenstern mitwachsen und nichts komischen passiert, wenn man zum Beispiel Fullscreen skaliert. Oft sollten horizontale Knopfleisten oben oder Statusleisten unten in X-Richtung mitwachsen aber nicht in Y-Richtung. In einem zweigeteilten (links/rechts) oder dreigeteilten (links/rechts oben/rechts unten) bleibt die linke Seite meist gleich breit, füllt sich aber nach unten. Rechts nimmt den Rest des Platzes auf. Rechts oben und rechts unten teilen Sie den übrigen Platz.

Aufgabe 2. Schreiben Sie ein GUI für die `chat`-Aufgabe aus Blatt 7 als `tkchat.py`.



Initial könnte es nur der Gesprächsverlauf sein und eine Eingabezeile mit Submit-Knopf. Es könnte zum Beispiel ein Refresh-Knopf dazu kommen, um den aktuellen Gesprächsverlauf vom Server zu holen. Auf einer Toplevel-Settings-Seite kann man Einstellungen vornehmen. Ein Auto-Updater holt sich (in einem separaten Thread) auch ohne Benutzereingaben den aktuellen Stand alle x Sekunden. Mit einem Beenden-Knopf beenden Sie das Programm (und alle Threads).

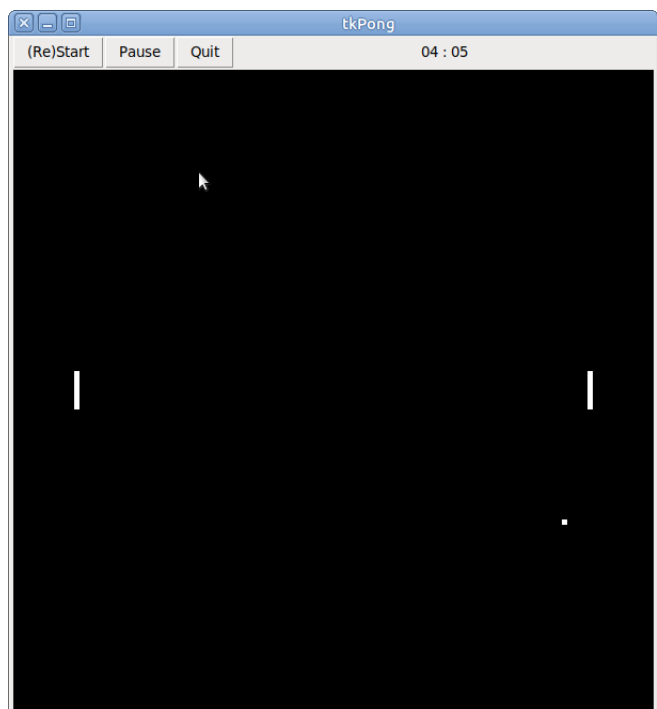
Aufgabe 3. Schreiben Sie ein GUI für die `fortune`-Aufgabe aus Blatt 3 als `tkfortune.py`. Auf jeden Fall gibt es einen Knopf, um ein beliebiges Fortune anzuzeigen und einen Knopf, um das Programm zu verlassen. Zusätzlich kann ein Text-Filter angegeben von einem Wort, das in dem Fortune enthalten sein muss.



Da die Fortunes aus mehreren Dateien kommen entspricht jede Datei einer Kategorie, die wir auswählen können. Ein Reset-Knopf setzt die Auswahl der Kategorien und der Filter zurück. Eine Statusleiste zeigt wie viel Fortunes in wie viel Kategorien wir zur Verfügung haben und aus wie vielen davon durch die aktuelle Auswahl und Filter wir eines wählen würden.

Aufgabe 4. Realisieren Sie Pong, ein sehr einfaches Arcade-Spiel, als `tkpong.py`.

Auf einem Spielfeld (Canvas) gibt es zwei Schläger, einen links und einen rechts, sowie einen Ball. Der Ball bewegt sich zwischen den Schlägern und prallt oben und unten ab. Der Ball bewegt sich von selbst initial ab der Mitte zufällig nach links oder rechts und zufällig in einem maximal 45 Grad Winkel. Die Schläger werden mit jeweils zwei Tasten (zum Beispiel links mit `w` und `s` und rechts `i` und `k`) bewegt. Wenn ein Schläger den Ball trifft, dann wird der Ball zurückgespielt. Vielleicht je nachdem wo ihn der Schläger trifft in einem anderen Winkel. Man kann das Spiel pausieren und beenden. Beim Beenden werden alle Threads beendet. Wenn der Schläger den Ball verfehlt erhält der Gegner einen Punkt. Die Punkte werden angezeigt. Wenn der erste Spieler 10 Punkte hat wird das Spiel pausiert. Mit Restart kann man ein neues Spiel starten.



Es steht in `tkgameutil.py` ein Hilfsmodul bereit, das Sie verwenden dürfen. Sie ermöglicht in einem Spiel kontinuierlich den Zustand von Tasten abzufragen. Realisieren Sie einen Gameloop mit fester Dauer, der 1) Eingaben (Tasten) abfragt, 2) einen Gametick (Fortschritt) macht 3) neu zeichnet und 4) den Rest der Zeit schläft. 25 FPS reicht völlig.

Hinweis 3. Sie dürfen die Templates `tktpl.py` oder `etkapp.py` verwenden, müssen dies aber nicht tun.

moodle Python_Barth