

In einer bestehenden Asteroidsimplementierung soll die visuelle Darstellung (View) geändert werden.

1 Vorbereitung

- Laden Sie sich aus Ilias die Quelltexte **asteroids.zip** hoch und entpacken Sie die Datei in dem Verzeichnis, in dem sich auch die Quelltexte der ersten Aufgabe entpackt haben. Es wird ein Unterverzeichnis **asteroids** erstellt, indem sich der Quelltextrahmen für die Aufgabe befindet.
- Kopieren Sie die Quelltexte Ihrer Lösung der zweiten Aufgabe in diesen Ordner (aber nicht CMakeLists.txt).

2 Makefile und Programm erstellen

Das Programm benötigt **SDL2** und **SDL2_Mixer**. Die entsprechenden Link-Libraries müssen vorhanden sein. Verwenden Sie **g++** der GNU-Compiler-Collection.

- Erstellen Sie mit **cmake** das Makefile und dann mit **make** die ausführbaren Programme. Es sollten keine Fehler auftreten. Lediglich einige Warnung über unbenutzte Variablen treten auf.
- Führen Sie die Testprogramme **physics_test** und **game_test** aus. Es dürfen keine Tests fehlschlagen. Falls doch, dann prüfen Sie erst, ob Ihre Implementierungen aus der ersten Aufgabe die Ursache sind, und beseitigen die Fehler.
- Den Ton des PCs im Labor bitte ausschalten (entsprechend markierte Funktionstaste).
- Führen Sie **main_game** aus. Es sollte sich ein Fenster öffnen und das Spiel gestartet werden. Sie können das Raumschiff mit den Pfeiltasten drehen und beschleunigen. Mit der Taste **D** wird geschossen. Die Leertaste löst einen Sprung in den Hyperspace aus.

3 View ändern

Die Anwendung ist nach dem Model-View-Controller (MVC) Paradigma aufgebaut. Die Darstellung und Steuerung wird mit Simple-Direct-Media-Layer Version 2 (SDL2) plattformunabhängig implementiert. Näheres zu SDL finden sie unter <https://www.libsdl.org/>.

Der View ist in der Klasse **sd12_renderer.cc** implementiert und **sd12_renderer.h** definiert.

- Ändern Sie Darstellung aller Spielobjekte nach Ihren eigenen Vorstellungen. Es dürfen aber weiterhin nur Striche oder Punkte gezeichnet werden.
- Verwenden Sie auch Farbe. Die Zeichenfarbe wird mit **SDL_SetRenderDrawColor(renderer, R, G, B, A)** geändert. Es wird ein ganzzahliges RGB-Farbmodell mit Alphakanal A verwendet. Jeder Farbwert wird mit 8-Bit codiert. In der Funktion **renderScore()** sind zwei Einzeilenkommentare, welche vor dem Zeichnen einer Linie die Farbe auf Gelb setzt und danach wieder für alle weiteren Zeichenoperationen auf Weiß. Probieren Sie dies aus.

- Das Spiel selbst (also die gesamte Modelllogik) darf nicht geändert oder erweitert werden. Lediglich `sld2_renderer.h` und `sd12_renderer.cc`! Den Controller sollten Sie auch nicht ändern.
- Beachten Sie, dass die bestehende Größe der Hitboxen (Kreise) nicht geändert werden darf, aber weiterhin mit der visuellen Darstellung sinnvoll korrespondieren muss.
- Ihre Änderungen können auch das Thema des Spiels ändern. Es muss kein Raumschiff, fliegende Untertasse oder Asteroid sein. Die Spiellogik bleibt allerdings so wie sie ist bestehen. Die Soundfiles im Ordner `sound` können Sie auch gegen eigene austauschen. Die Dateinamen müssen allerdings identisch zu den bestehenden sein.
- Sie können auch animierte Effekte in der Darstellung implementieren. Z.B. zeitgesteuert, so wie das bei den Explosionen der Fall ist. Sie können dazu im Renderer zusätzliche Klassen und Objekte programmieren. Animationen lassen sich auch über die Position der Objekte realisieren: Bei geraden X-Koordinaten des UFOs eine andere Darstellung zeichnen als bei ungeraden.