Eine OpenGL-Version von Asteroids soll zum Laufen gebracht und einige Änderungen vorgenommen werden.

1 Vorbereitungen

- Sie benötigen neben den bisherigen Paketen noch GLEW und OpenGL.
 - Windows: Die OpenGL-Bibliothek müsste schon vorhanden sein (C:\Windows\System32\openg132.d11), ebenso die zugehörigen Headerdateien. Installation von GLEW mit MinGW mit pacman -S mingw-w64-x86_64-glew (Systemarchitektur gegebenenfalls anpassen).
 - Linux: Die zu installierenden Pakete heißen libopengl-dev und libglew-dev. Eine freie OpenGL-Implementierung ist Mesa. Normalerweise sind Treiber für die Hardware GPU aber installiert.
- Erstellen Sie ein neues Verzeichnis für diese Aufgabe.
- Laden Sie sich die ZIP-Datei asteroids_opengl.zip aus Ilias herunter.
- Entpacken Sie die Datei. Es wird ein Unterverzeichnis asteroids_opengl erstellt.
- Kopieren Sie sich die C++-Dateien Ihrer Lösung und Tests der vorherigen Aufgabe (.h, .cc und .tcc) in dieses Verzeichnis. Kopieren Sie auch Ihre Lösung und Tests der geometry-Dateien darin.
- Erstellen Sie mit CMake das Makefile. Wenn Sie unter Windows arbeiten, dann müssen Sie vorher die entsprechende Stelle für die Bibliotheken einkommentieren (die für Unix auskommentieren).
- Übersetzten Sie das Programm und starten Sie das Spiel (bzw. beheben Sie vorher die auftretenden Fehler).
- Ändern Sie die Auflösung in main_game.cc. Da die Viewport-Transformation von OpenGL durchgeführt wird, sollte das Spiel in der geänderten Auflösung funktionieren. Es ist ein 4:3-Verhältnis der Auflösung nötig, damit keine Verzerrung auftritt.

2 Horizontales und Vertikales Scrolling, Teil A

- Die Vertex-Buffer-Objects werden am Spielanfang genau einmal initialisiert.
- Die Vertex-Array-Objects werden für jedes Spielobjekt neu erzeugt, wenn es von der Spiellogik neu erstellt wurde. Sie werden zerstört, wenn das Spielobjekt aus der Physik-Engine entfernt wurde. Beides geschieht in der render()-Methode.
- In der render()-Methode werden mit einer Transformationsmatrix, die Weltkoordinaten auf die kanonische Sicht transformiert. Dabei wird auch die y-Achse in ein links-händisches Koordinatensystem transformiert.
- Der View soll so geändert werden, dass das Raumschiff immer in der Mitte gezeichnet wird und die Objekte entsprechend verschoben werden. Dies darf nur gemacht werden, wenn das Raumschiff existiert. Mit game.ship_exists() können Sie abfragen, ob das Raumschiff existiert. Mit game.get_ship() bekommen Sie den Zeiger auf das Raumschiff. Wenn das Raumschiff nicht existiert, ist der Werte nullptr.

- Die Anzeige der Punkte und freien Raumschiffe dürfen nicht verschoben werden.
- Überlegen Sie sich eine geeignete Transformation (es handelt sich lediglich um eine Translation) und binden Sie diese zwischen den bestehenden Transformationsmatrizen ein.

3 Horizontales und Vertikales Scrolling, Teil B

Wenn das Raumschiff zu den Grenzen des Spielfelds hin bewegt, dann werde die entgegen kommenden Objekte nicht gezeichnet. Statt die Objekte nur einmal zu zeichnen, sollen sie neun-mal gekachelt an unterschiedlichen Positionen innerhalb der for (auto & view: views)-Schleife gezeichnet werden.



- Es dürfen keine zusätzlichen Vertex-Arrray-Objekte erzeugt werden. Die vorhanden View-Objekte müssen lediglich mehrfach gezeichnet werden.
- Überlegen Sie sich wieder geeignete Transformationen (es handelt sich auch hier lediglich um Translationen) und binden Sie diese zwischen den bestehenden Transformationsmatrizen ein.
- Das Raumschiff muss danach nahtlos durch den Spielbereich durchfliegen.
- Beachten Sie:
 - Die fliegende Untertasse taucht und verschwindet jetzt natürlich auch im sichtbaren Bereich und nicht mehr am Rand auf.
 - Sie selbst können nicht mehr wahrnehmen, ob sie am Rand sind. Hier tauchen die neuen Asteroiden auf. Also Vorsicht.
 - Bei Zerstörung des Raumschiffs oder Hyperspace springt die Darstellung.

4 Hinweise

- Es gibt debug-Ausgaben. Alle mit Level 0 werden immer ausgegeben (z.B. error). Der Debuglevel ist lediglich ein Präprozessorsymbol und kann auch beim Compileraufruf gesetzt werden. Es handelt sich lediglich um sehr einfache C-Makros.
- Sie können wie bei der ersten Aufgabe auch Animationen versuchen einzubauen, ein Ego-Shooter-Perspektive oder anderes. Das Spiel muss aber als Asteroids-Spiel erkennbar bleiben.