

## 8. Übungsblatt zur Vorlesung Computergraphik im WS 2017/18

Abgabe am Mittwoch, 20.12.2017 12:00 Uhr

Besprechung am Montag, 08.01.2018 17:30 - 19:00 Uhr

### Aufgabe 1 *Polygon-Clipping* [2 Punkte]

Welche maximale Anzahl neuer Vertices kann entstehen, wenn

- ein n-seitiges konvexes Polygon an einer Geraden geclippt wird?
- ein n-seitiges konkaves Polygon an einer Geraden geclippt wird?
- ein n-seitiges konvexes Polygon an einem Rechteck geclippt wird?
- ein n-seitiges konkaves (möglicherweise sich selbst überdeckendes) Polygon an einem Rechteck geclippt wird?

### Aufgabe 2 *Hello, OpenGL* [8 Punkte]

Diese Aufgabe dient als Einstieg in die Programmierung mit OpenGL. Über ILIAS erhalten Sie die Visual-Studio-Solution HelloGL, in der die zu bearbeitenden Stellen mit `TODO` gekennzeichnet sind.

**Hinweis:** Unterstützt werden Visual Studio 2015 und 2017. Bei der Installation müssen Sie unter „Einzelne Komponenten“ den „NuGet-Paket-Manager“ aktivieren. Sollten Sie Visual Studio 2017 verwenden, brauchen Sie zusätzlich das „Toolset für VC++ 2015.3 v140“. Diese Features lassen sich auch im Nachhinein mit dem Visual Studio Installer hinzufügen. Achten Sie bitte auch darauf, beim ersten Öffnen des Projekts, die verwendete Toolset-Version *nicht* auf v141 umzustellen (kein Upgrade).

Das Programmskelett greift für die Verwendung von OpenGL auf externe Open-Source-Bibliotheken zu. In der mitgelieferten Visual-Studio-Solution sind die Bibliotheken bereits eingebunden und werden (sofern der NuGet-Paket-Manager installiert ist) automatisch nachgeladen. Falls Sie einen anderen Compiler verwenden wollen, müssen Sie sich selbst um die Einbindung folgender Bibliotheken kümmern:

- glfw: <http://www.glfw.org/>
- glad: <https://github.com/Dav1dde/glad>
- glm: <https://glm.g-truc.net>

**Hinweis:** Falls Sie das Projekt nicht aus Visual Studio starten, müssen Sie gegebenenfalls die Pfade zu den Shader-Dateien im Code anpassen (in der Funktion `InitRenderer` in der `main.cpp`).

Wenn Sie das Projekt öffnen, kompilieren und das Programm starten, sollte sich ein leeres Fenster öffnen.

1. **Shader-Setup** (2 Punkte)

Vervollständigen Sie das Shader-Setup, indem Sie die Funktionen `CompileShader` und `LinkProgram` in `main.cpp` ergänzen. Sollten die Shader korrekt kompiliert und an das erstellte Program angehängt sein, sowie dieses gelinkt sein, wird ein gelbes Dreieck angezeigt (vgl. Abbildung 1).

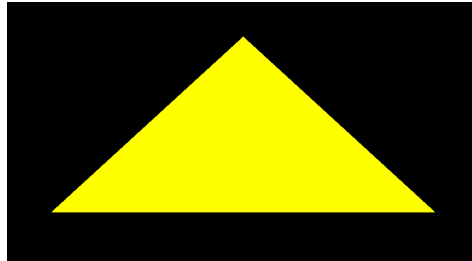


Abbildung 1: Ausgabe nach korrekter Bearbeitung von Teilaufgabe 1.

2. **Vertex-Array** (4 Punkte)

Ebenfalls in `main.cpp` finden Sie die Funktion `InitRenderer`, die das Fenster öffnet, den OpenGL-Kontext initialisiert und die Funktionen zum Shader-Setup aufruft. Ergänzen Sie den Code um die Erstellung eines Vertex-Buffers und eines zugehörigen Vertex-Attribute-Pointers, um einen Stern bestehend aus vier Dreiecken zu zeichnen (vgl. Abbildung 2). Die Vertex-Positionen liegen bereits im Array `vertices` vor. Stellen Sie außerdem sicher, dass im Vertex-Shader `vert.glsl` korrekt auf die Daten zugegriffen wird. Genauer können Sie den Kommentaren im vorgegebenen Quellcode entnehmen.

3. **Fenster-Transformation** (2 Punkte)

Im diesem Zustand führt ein Skalieren des Fensters dazu, dass der Fensterinhalt seine Form verändert. Das dargestellte Polygon soll zwar seine Größe proportional zur Höhe des Fensters anpassen, aber seine Form beibehalten.

Im Vertex-Shader `vert.glsl` ist eine Matrix namens `transform` deklariert. Diese wird aktuell jedoch weder vom Hauptprogramm an den Shader übergeben, noch im Shader verwendet. Fügen Sie in der Funktion `UpdateTransform` Code hinzu, der eine Matrix erstellt und unter Verwendung der OpenGL-Funktion `glUniformMatrix4fv` an den Shader übergibt (sodass der Vertex-Shader über `transform` darauf zugreifen kann). Modifizieren Sie außerdem den Vertex-Shader, damit er die übergebene Matrix verwendet. Die übergebene Matrix soll dabei den oben spezifizierten Effekt erzielen.

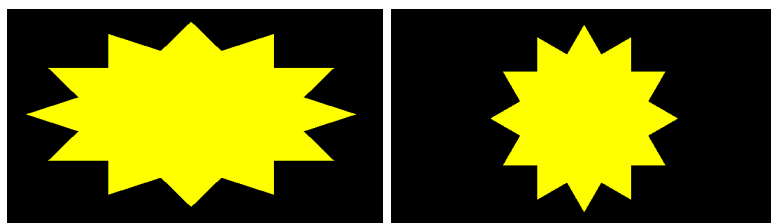


Abbildung 2: Ausgabe nach korrekter Bearbeitung von Teilaufgabe 2. (links) und 3. (rechts).

**Bitte laden Sie bei Ihrer Abgabe keine Binaries hoch!**