

## Computergrafik 2 / Aufgabe 1: Warmup

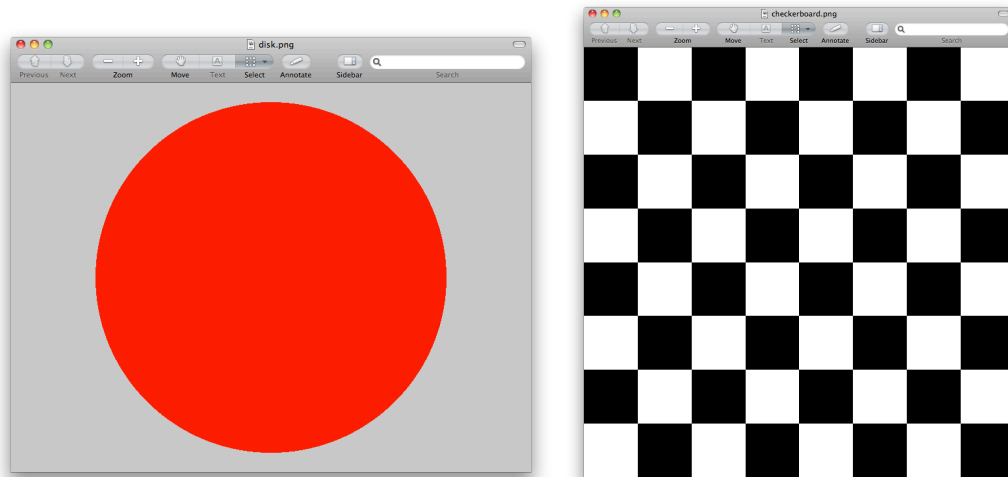


Abbildung: Erzeugen Sie Pixel für Pixel ein 8x8-Schachbrettmuster und eine 90%-Kreisscheibe.

In den kommenden Wochen werden Sie einen einfachen Raytracer implementieren. Dieser soll die erzeugten Bilder im PNG-Format auf der Festplatte ablegen können. Die erste Aufgabe besteht aus einigen vorbereitenden Arbeiten zur Eingewöhnung in die Entwicklungswerkzeuge und die Arbeitsweise des Raytracers.

### Gruppeneinteilung

Bilden Sie Arbeitsgruppen mit jeweils 2 Personen (nach Möglichkeit auch für die weiteren Aufgaben). Teilen Sie mir Namen, Emailadresse und Matrikelnummern der Gruppenmitglieder per Email an [hschirmacher@beuth-hochschule.de](mailto:hschirmacher@beuth-hochschule.de) **bis Mittwoch den 12.10.2011, 10:00 Uhr** mit.

### Auswahl der Programmiersprache

Wählen Sie die Programmiersprache, in der Sie den Raytracer implementieren wollen. Prinzipiell ist jede Programmiersprache geeignet; übersetzte Sprachen haben gegenüber interpretierten Sprachen aber meist einen deutlichen Geschwindigkeitsvorteil. Ich empfehle eine Implementierung in Java oder C++. Um den Start zu erleichtern, stelle ich für die Programmiersprachen Java und C++ jeweils ein minimales Framework zu Verfügung (siehe unten: Material).

Da der Fokus in dieser Vorlesung auf der Computergrafik und nicht auf der Programmiersprache liegt, kann ich nur in begrenztem Umfang Hilfestellung bei sprachbezogenen Problemen leisten. Sollten Sie also bisher gar nicht mit C/C++ und der

Verwendung von Zeigern und Templates vertraut und stundenplanmäßig ausgelastet sein, rate ich von der Verwendung von C++ für dieses Projekt ab.

## Die Aufgabe

Schreiben Sie ein Programm, das die folgenden beiden Bilder (siehe Abbildung) in beliebiger Größe erzeugt und diese im PNG-Format auf der Festplatte ablegt:

1. Ein Schachbrettmuster mit 64 schwarzen und weißen Feldern.
2. Eine zentrierte rote Kreisscheibe auf grauem Hintergrund. Der Durchmesser des Kreises entspricht 90% der kürzesten Kante des Bildes.

Benutzen Sie nicht eventuell vorhandene Zeichenroutinen zur Darstellung von Kreisen und Rechtecken sondern setzen Sie das Bild Pixel für Pixel zusammen. Veranschaulichen Sie sich den Aufbau des Beispiel-Frameworks:

- Die Klasse `ImageGenerator` geht alle Pixel des zu erzeugenden Bildes in einer festen Reihenfolge durch und ruft dann die Methode `getPixelAt()` des Interfaces `Painter` auf.
- Hinter dem `Painter`-Interface verbirgt sich entweder ein `Checkerboard`-Objekt oder ein `Disk`-Objekt – je nachdem mit welchem Objekt der `ImageGenerator` aufgerufen wurde (das `Disk`-Objekt ist im Beispiel noch nicht implementiert; dies gehört zu Ihrer Aufgabe).
- Ein `Checkerboard`-Klassenrumpf wird in dem Beispiel mitgeliefert. Das resultierende Bild ist zwar ein Schachbrettmuster, stellt aber noch nicht die gewünschte Lösung dar.
- Die Klasse `Disk` fehlt in dem Beispiel – diese müssen Sie analog zu `Checkerboard` selbst entwerfen.

Achten Sie darauf, dass die Bilder für alle möglichen Bildseitenverhältnisse und Auflösungen korrekt sind und z.B. keine unerwünschten Randpixel in der falschen Farbe auftauchen.

## Abgabe

Diese Aufgabe muss nicht abgegeben werden. Sie dient zur Einarbeitung in die Entwicklungswerkzeuge und zur Vorbereitung der folgenden Aufgaben und sollte leicht innerhalb von einer Woche zu erledigen sein.

## Material / Projektsetup

Bitte laden Sie das kleine Projektframework in Java oder C++ von der Webseite zur Vorlesung auf <http://schirmacher.beuth-hochschule.de> herunter.

### Einrichten des Projekts in Java / Eclipse

- Entpacken Sie die Datei `CG2-Warmup-Java.zip` in ein beliebiges Verzeichnis
- Starten Sie *Eclipse* und importieren Sie das Projekt in Ihren Workspace, indem Sie „Import / General / Existing Projects into Workspace“ auswählen und damit das entpackte Verzeichnis komplett in den Workspace kopieren.

### Einrichten des Projektes in C++ mit Qt

- Für den Fall einer Implementierung in C++ empfehle ich, das Qt-Framework von Nokia (vormals Trolltech) zu verwenden. Das Warmup-C++-Beispielprojekt ist ebenfalls gegen Qt entworfen. Qt ist ein Cross-Plattform-Framework, welches in C++ viele Funktionen ähnlich den Java-Klassenbibliotheken für Windows, Mac OS, Linux und viele Mobile Plattformen verfügbar macht.
- Von Qt können Sie eine C++-Klassenbibliothek (*Qt Library*) sowie eine Entwicklungsumgebung (*SDK*) erhalten. Auf den Laborrechnern ist beides bereits vorinstalliert.
- Um Qt auf Ihren eigenen Rechnern zu installieren, laden Sie die entsprechende Version des SDK von Nokia herunter und folgen Sie den Anweisungen. Bei Installation des SDK wird auch die Library automatisch mitinstalliert.
- Nach der Installation starten Sie den *Qt Creator*; dieser ist eine einfach zu bedienende Entwicklungsumgebung ähnlich dem *Eclipse*-GUI. Selbstverständlich können Sie auch andere Umgebungen wie *Visual Studio* verwenden; dort müssen Sie Qt allerdings manuell als Library eintragen etc. – Hilfe hierzu finden Sie im Netz.
- Wählen Sie “Create Project...” / “Other Project” / “Empty Qt Project”
- Wählen Sie als Projektnamen “Warmup” und als Root-Verzeichnis z.B. `~/Dev/CG2`. In diesem Beispiel würde Qt ein Verzeichnis `~/Dev/CG2/Warmup` anlegen. Der gewählte Projektname bestimmt automatisch den Standardnamen des Executables, d.h. es wird ein ausführbares Programm namens „Warmup“ erwartet.
- Im nächsten Schritt des “Create Project” Wizards wählen Sie “Desktop” als Zielplattform, wenn Sie eine Applikation für einen PC erstellen wollen. Folgen Sie den weiteren Instruktionen.
- Nachdem Sie das Projekt erstellt haben, entpacken Sie die `CG2-Warmup-CPP.zip` von der Webseite und importieren Sie die darin enthaltenen Quelldateien, indem Sie oben links im Creator auf das Projekt rechtsklicken, „Add Existing Files...“ auswählen, und alle `.cpp` und `.h` Dateien aus dem `.zip` auswählen.