



VEKTORIA-MANUAL

EINFUEHRUNG &

INSTALLATION



Game Design

Inhalt



||||**VEKTORIA BEDEUTUNG**|||

||||**VEKTORIA GRUNDLAGEN**|||

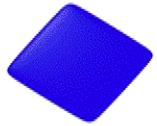
||||**VEKTORIA PHILOSOPHIE**|||

||||**VEKTORIA HISTORIE**|||

||||**VEKTORIA INSTALLAT.**|||



///VEKTORIA ///BEDEUTUNG



2 // / /

3 // / /

4 // / /

5 // / /



Vektoria Bedeutung

Namensursprung Vektoria

Vektor + Viktoria = Vektoria

1 // / / /

2 // / / /

3 // / / /

4 // / / /

5 // / / /

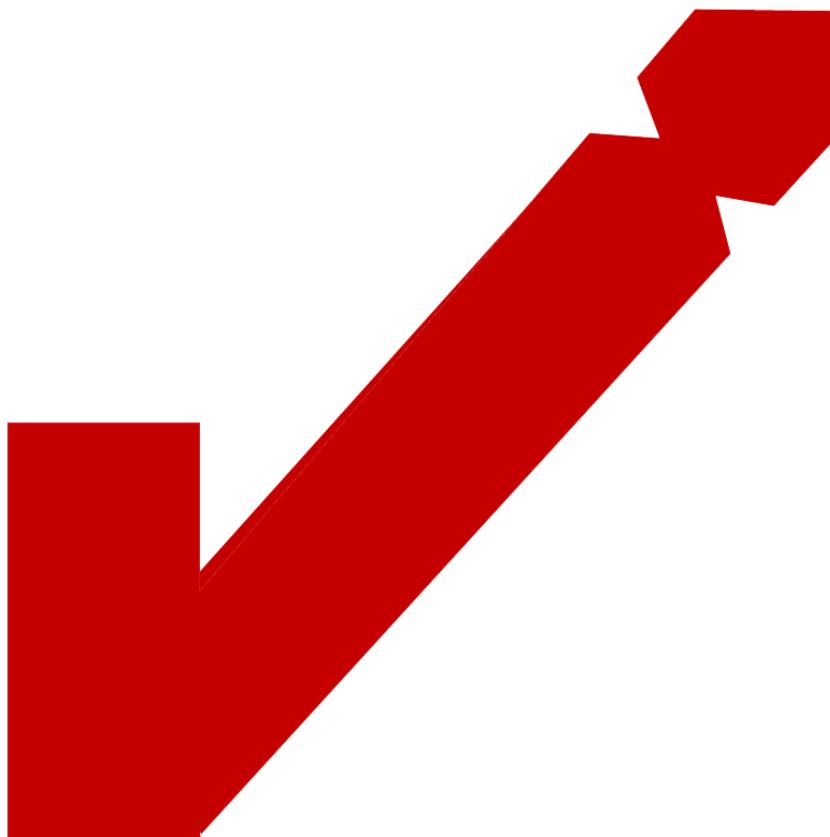
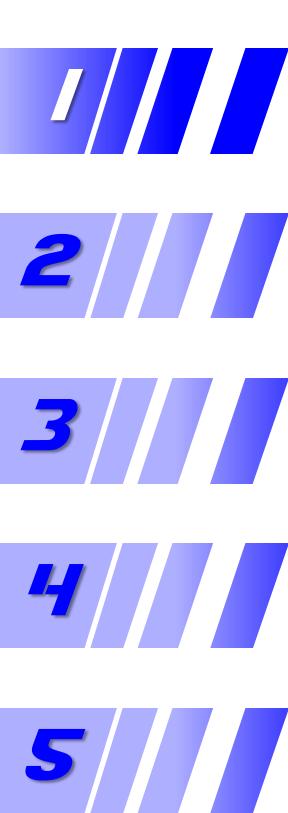
Vektor:
lat.
Träger des

Viktoria:
röm. Göttin des
Sieges und des
Erfolges



Vektoria Bedeutung

Vektoria Symbol



Konnotationen:

Vektorpfeil

V-Buchstabe

Victoryzeichen

OK-Häkchen

Aufwärtsstreben

Phallische Kraft

Steigende Kurse

Erfolg

Schmunzelmund

...



Warum Vektoria benutzen?

- Vektoria ist kostenfrei für die Lehre und nichtkommerzielle Zwecke.
- Durch optionales State-Sorting, hierarchisches Frustum Culling und Smart Allocating werden beste Bildwiederholraten bei komplexen Szenen erzielt.
- Durch einheitliche klar objektorientierte Struktur ist eine schnellere Erstellung von Games und Simulationen möglich.
- Vekoria enthält den zurzeit mächtigsten Szenegrafen, so sind u.a. zusammenlaufende Hierarchien möglich.
- Vekoria wird zurzeit rasant weiterentwickelt.
- Vekoria enthält sehr viele Features, die von anderen Szenegrafen bzw. GameEngines nicht angeboten werden. Siehe nächste Seiten!



Vektoria Bedeutung

Features von Vektoria



Sky Cubes

GlowMapping

Bump Mapping

Billboards

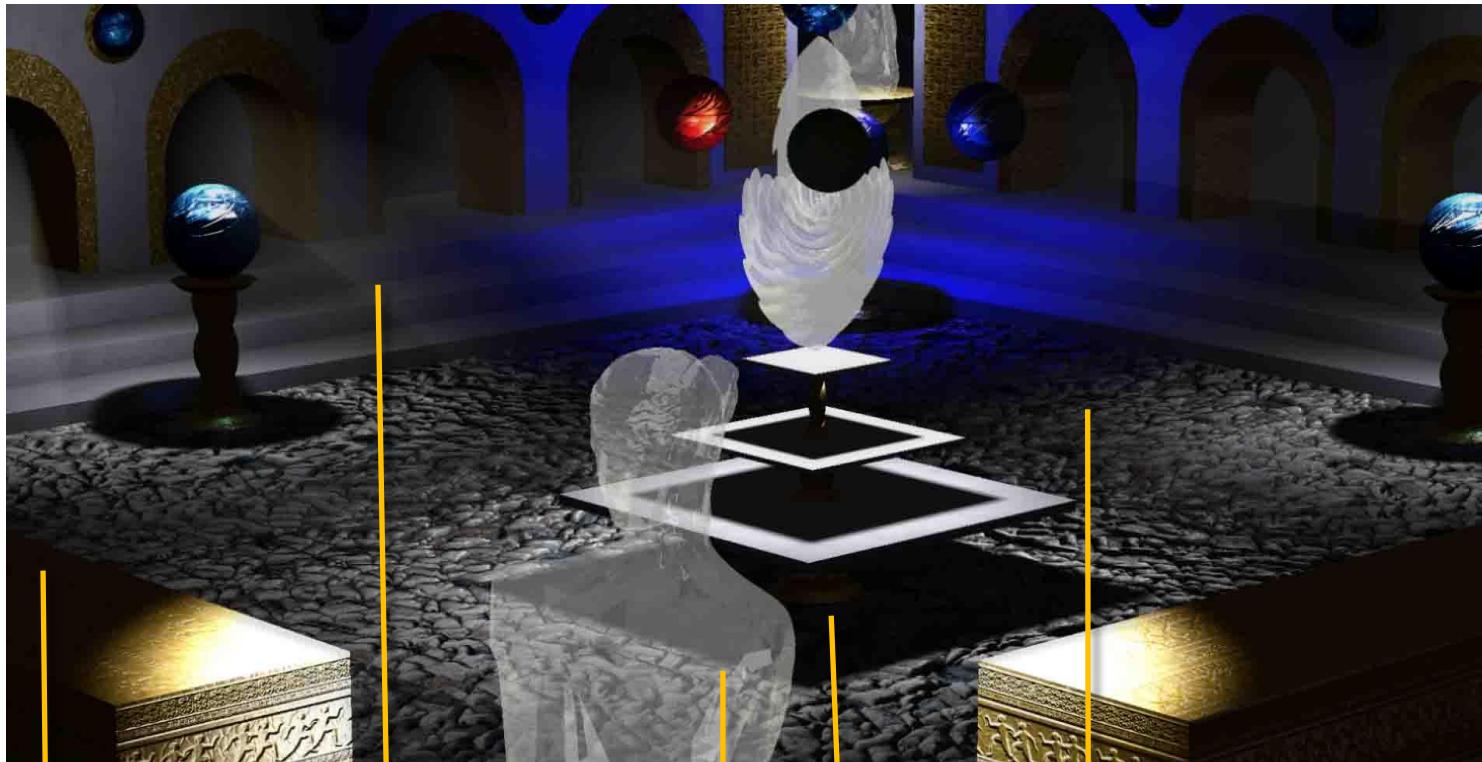
Fog

Alpha Textures

Mighty Scenegraph



Features von Vektoria



1 // / / /

Bloom

2 // / / /

God Rays

3 // / / /

Shadows

4 // / / /

Parallax Occlusion
Mapping

5 // / / /

Forward Plus Rendering



Vektoria Bedeutung

Features von Vektoria



Multi-Skydomes

Hierachitecture Modelling

Environmental
Mapping

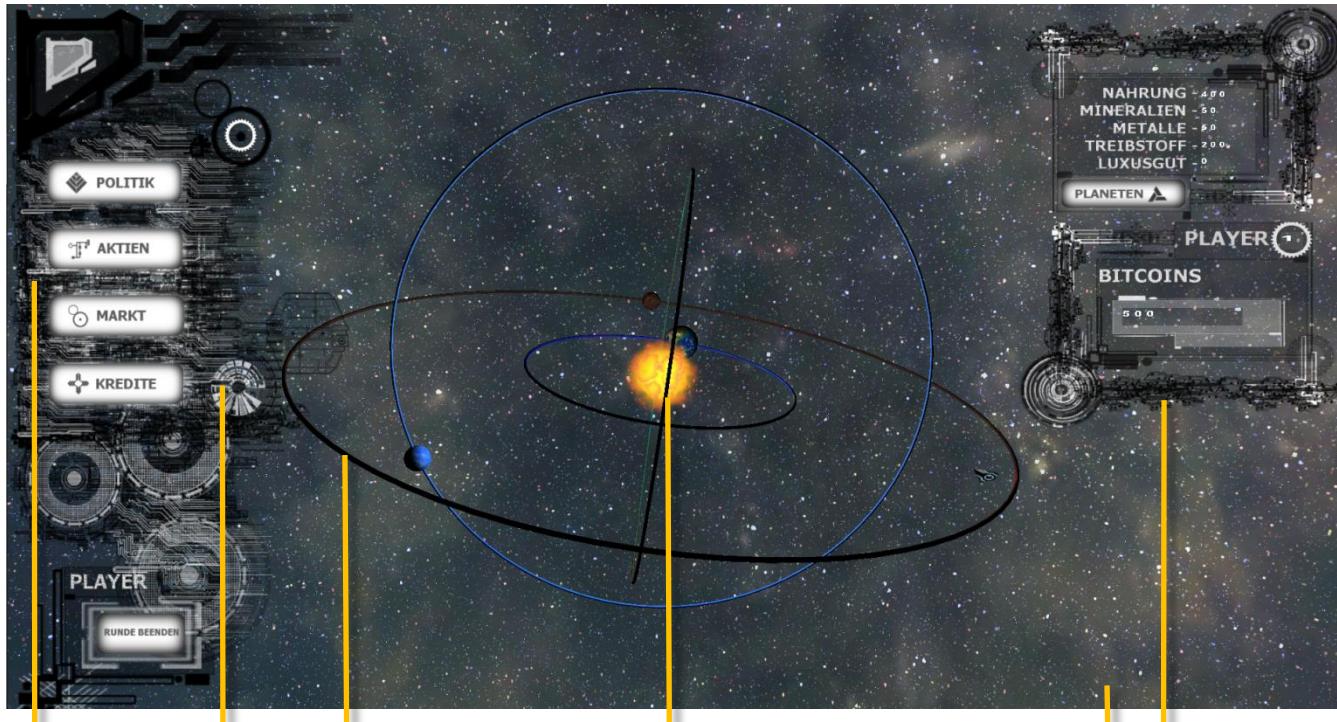
High Frame Rate by State Sorting

Specular Mapping



Vektoria Bedeutung

Features von Vektoria



Sprites

3D Sound

GUIs

Doppler Effect

Animated Textures

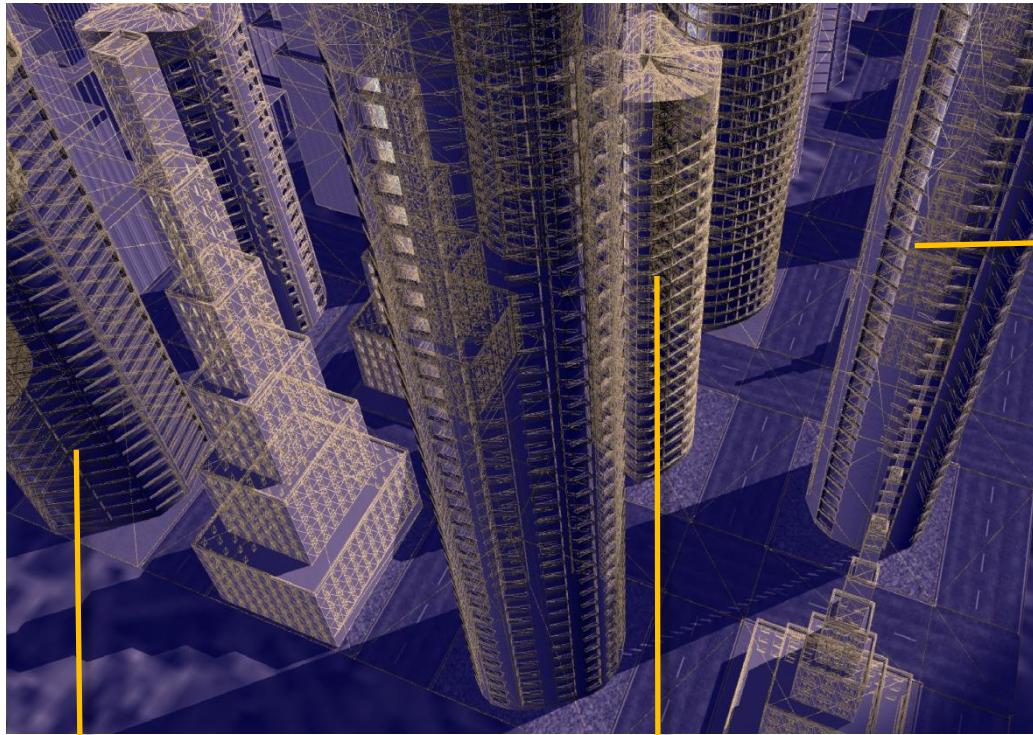
Cursor



Vektoria Bedeutung

Features von Vektoria

- 1 // / / /
- 2 // / / /
- 3 // / / /
- 4 // / / /
- 5 // / / /

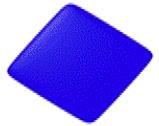


Kapitel 2

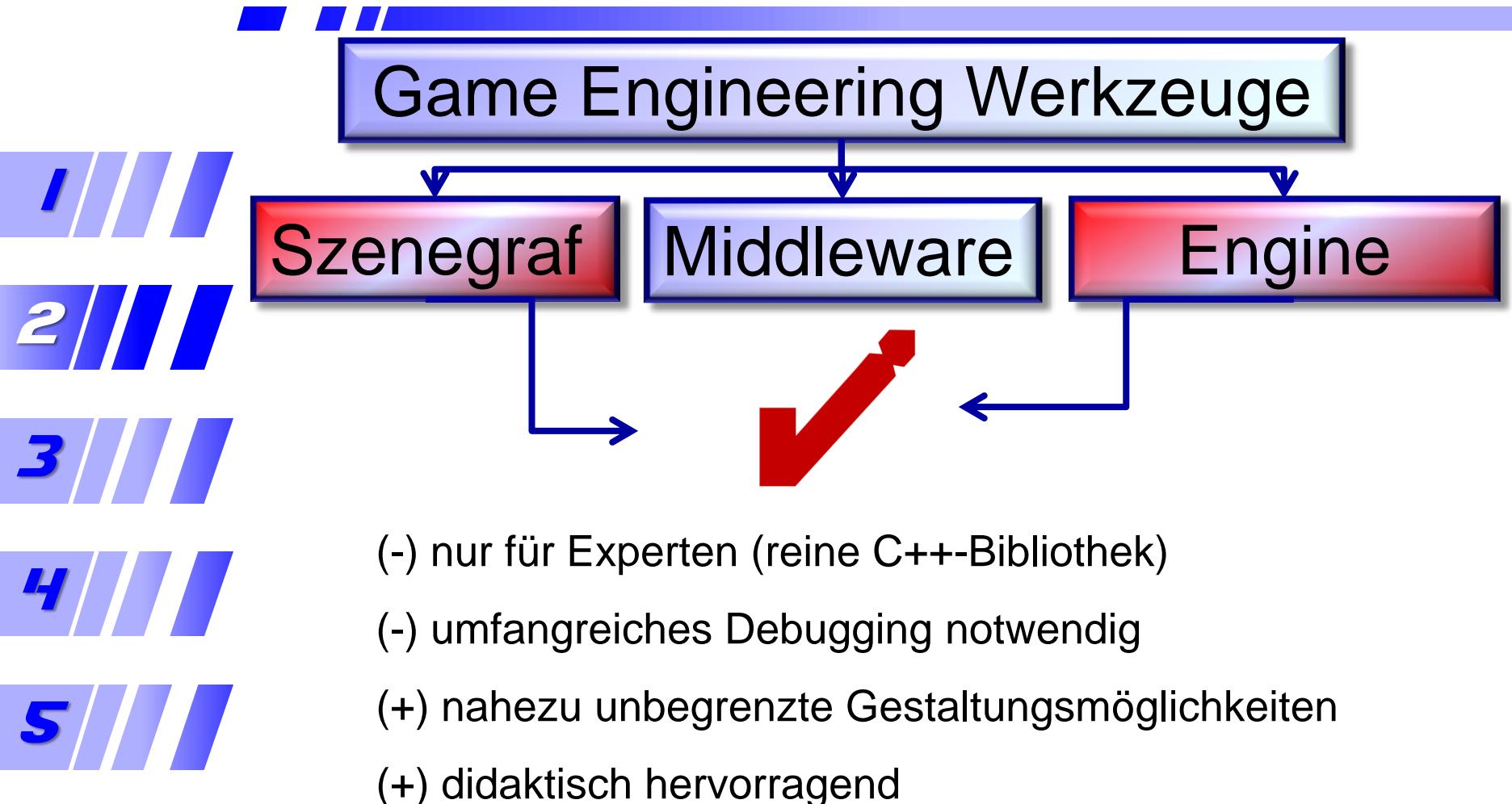
Kapitel 2



///VEKTORIA ///GRUNDLAGEN ///



Ansiedelung von Vektoria



Was ist Vektoria?

1 // / / / Bibliotheke für

- Games und
- Virtuelle 3D-Echtzeitsimulationen

2 // / / / basierend auf

- DirectX 11 und
- Visual C++

3 // / / / Also „nur“ eine C++-Bibliothek, aber:

- hat einen Szenegrafen integriert
- besitzt typische Methoden einer Game Engine

4 // / / / 5 // / / /



Was ist Vektoria *nicht*?

Es ist

- kein Modellierungsprogramm
- kein Animationsprogramm
- kein Leveleditor
- keine Middleware
- keine Point-and-Klick-Gameengine



Aufbau eines Vektoria-Spiels



Kapitel 3

Kapitel 3



1 // / / /

2 // / / /

/// VEKTORIA /// PHILOSOPHIE ///



4 // / / /

5 // / / /



Philosophie von Vektoria

Motto 1:

Vektoria ist klar und schlicht

2 Jedes Objekt hat eine genormte einheitliche Schnittstelle.
Vektoria ist frei von „Schnickschnack“-Features, welche die Handhabung unnötig erschweren.

3 Vektoria kommt mit einem „virtuellen Bauhausstil“ daher –
einfach und elegant zugleich.

4

5



Philosophie von Vektoria

Motto 2:

Vektoria lässt Mündigkeiten

1 // / / / Es gibt z.B. keine Überprüfungen der Ein- und Ausgabeparameter, welche die Codeausführung unnötig verlangsamen.

2 // / / / Der Programmierer wird für mündig genug gehalten, die Methoden mit den richtigen Parameter aufzurufen.

3 // / / / Es gibt fast keine protected- oder private-Membervariablen. Der Programmierer wird für mündig gehalten, auch in fremde Klassen sinnvoll einzugreifen.

Philosophie von Vektoria

Motto 3:

Vektoria abstrahiert

1 // / / / Die Engine abstrahiert von den technischen Gegebenheiten
Der Anwender braucht also für die Verwendung nichts von „Shadern“, „DirektX“ etc. zu verstehen.

2 // / / / Wenn man trotzdem die Shadern manipulieren will, ist dies dennoch möglich.

3 // / / / Der Aufbau einer 3D-Szene ist schnell möglich.
Trotzdem ist Vektoria sehr mächtig.

4 // / / / Der Aufbau einer 3D-Szene ist schnell möglich.
Trotzdem ist Vektoria sehr mächtig.

5 // / / / Der Aufbau einer 3D-Szene ist schnell möglich.
Trotzdem ist Vektoria sehr mächtig.



Philosophie von Vektoria

Motto 4:

Vektoria ist versteckt und offen



Vektoria kann für die Lehre und nichtkommerzielle Zwecke frei verwendet werden.

Kommerzielle Projekte müssen 5% des Umsatzes an Vektoria ausbezahlen.

Die Einsicht und die Entwicklung des Source Codes bleibt nur den Mitwirkenden vorbehalten. Die Library und die Schnittstellen sind dagegen offen.



Philosophie von Vektoria

Motto 5:

Vektoria ist panokratisch

1 // / / / Vektoria lässt den Mitwirkenden alle Freiheiten.

2 // / / / Es gibt keine Top-Down-Organisation.

3 // / / / Alle Entscheidungen werden direktdemokratisch per Elescheid von unten nach oben gelöst.

4 // / / / Vektoria ist somit vom Herstellungsprozess her strukturierte Anarchie.

5



Kapitel 4

Kapitel 4



1 // / / /

2 // / / /

3 // / / /

VEKTORIA HISTORIE



5 // / / /



PROF. DR. TOBIAS BREINER
VEKTORIA MANUAL

23 VON 49
VEKTORIA EINFUEHRUNG



Anfänge



- Erste Ideen, Klassendiagramm
- Evaluationen von Engines, ErsterCode



- Erste Programmierarbeiten am Szenegraph
- Entwicklung des Code-Dokumentes



- Finale Namensgebung, Logo, Abstimmung
- Erste Erfolge mit Dreieck



Erste Erfolge



- Primitive (Kugel, Tetraeder, Cube)
- Texturierung, Szenegraph



- Kamera-Bewegung (World-View)
- Umstellung auf Visual C++ 2010



- Komplettumbau (Beschleunigung)
- Erste Alpha-Version (V3)



Panokratische Mozelle, V4 & V5



- Gründung der panokratischen Mozelle Vektoria



- Version V4 (u.a. mit Lichern, Quaternionen, X3D-Importer)



- Version V5 (u.a. mit Stilen, Multiviewports, Schatten)



V6, V7 und GamesCom



- Version V6 (Frames, Overlays, Backgrounds)



- Version V7 (u.a. mit SMD-Modellen)



- Präsentation des ersten Vektoria-Spiels auf der GamesCom



Panokratische Elescheide



- Erste panokratische Elescheide



- Entwicklung des Vektoriportals
www.vektoria.de - Vektoria wird öffentlich



- Version V8 (u.a. mit hierarchischen Overlays, Architekturklassen.)

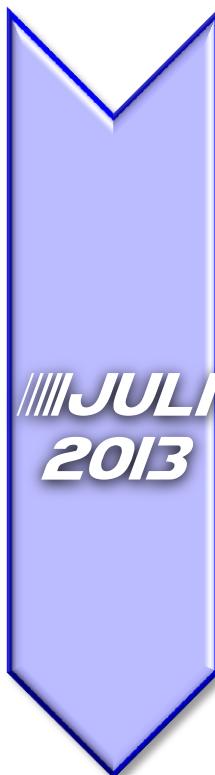


Geschichte von Vektoria

V9



- Version V9, u.a mit:
- **Modellierungsfunktionen** (Tapering, Twisting, Bending, Extruding, Sweeping, Subdivision, ...),
- **Mappingfunktionen** (planar, zylindrisch, bizylindrisch, orthografisch, kubisch, ...)
- **unlimitierter Hierarchie** (Es können nun beliebig viele Knotenobjekte an Placements oder Szenen an- und abgehängen werden, nun auch während der Laufzeit)
- **geometrische Architekturfunktionen** (Wände, Türme, Kuppeln, Fenster, Zinnen, Gärungen, ...)
- **optionales State Sorting** (Beschleunigt statische Hierarchieteile)



Geschichte von Vektoria

V10

- Version V10, u.a mit:
- Umstellung auf VS 2013,
- Parallax Occlusion Mapping, Environmental Mapping, Sky Mapping
- Mathematik für Strahlen und Ebenen
- Subs bei Materials
- erweiterete Architekturfunktionen
- Partikelsysteme



Geschichte von Vektoria

V11

- Version V11, u.a mit:
- Beschleunigtem Laden
- Writings, Wribels, und TextLabels
- Animierten Texturen und Materialien
- Vereinfachten Texturerzeugungsroutinen
- Switches für alle Knoten
- erweiterete Hierarchiefunktionen
- erweiterte Partikelsysteme
- Vektoria Intro



1 // / / /

2 // / / /

3 // / / /

4 // / / /

5 // / / /



Geschichte von Vektoria

V12

- Version V12, u.a mit:
- Modularisierung des Codes
- Picking
- Environmental Mapping
- Subsurface Scattering
- Parallax Occlusion Mapping
- Sky-, Sprite- und BRDF-Texturen
- vielen kleineren Verbesserungen



Der vektorianische Horrorherbst

- Durch einen panokratischen Elescheid wird die panokratische Struktur von Vektoria selbst abgewählt.
- Danach treten viele fähige Mitglieder aus dem Projekt aus.
- Eine zeitlang ist es unklar, ob und wie es mit Vektoria überhaupt weitergeht.
- Um 4 Jahre Softwareentwicklung nicht zu gefährden, wird letzten Endes gemeinsam entschieden, dass unter der Leitung von Tobias Breiner eine UG (haftungsbeschränkt) gegründet wird, im der unter anderem einige panokratische Elemente bestehen bleiben.

Geschichte von Vektoria

V13

- Version V13, u.a mit:
 - ForwardPlus-Renderer
 - BVH-Frustum Culling
 - Postprocessing
 - Blurring
 - Godrays
 - Layer für Overlays
 - erweitertem Picking
 - erweiterter Kollisionserkennung
 - vielen kleineren Verbesserungen

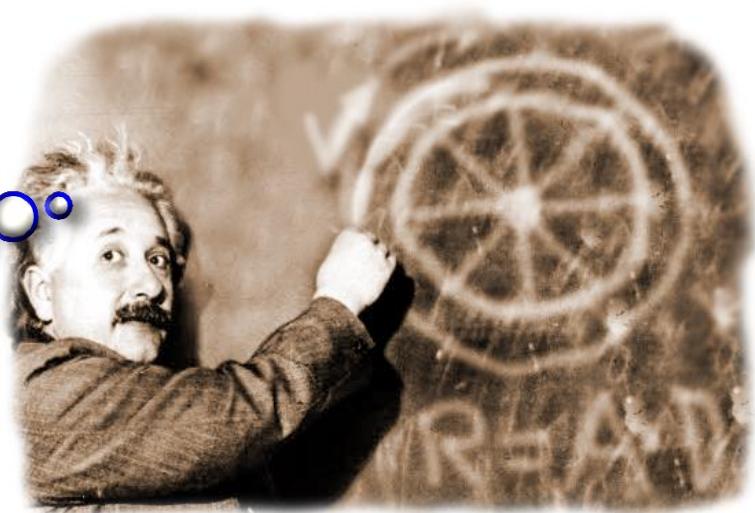


Warum das Rad neu erfinden?

Bisher existiert keine Engine, die ...

- direkte Manipulationen erlaubt.
- eine durchdachte, objektorientierte Struktur hat.
- verteiltes Rendering im LAN ermöglicht.
- Direct X11 voll einbindet.
- richtig schnell ist.

Zudem ist die
Wiedererfindung des
Rades für Studierende
didaktisch sinnvoll und
macht Spaß!



Kapitel 5

Kapitel 5



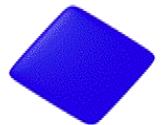
1 // / / /

2 // / / /

3 // / / /

4 // / / /

///VEKTORIA ///INSTALLATION ///

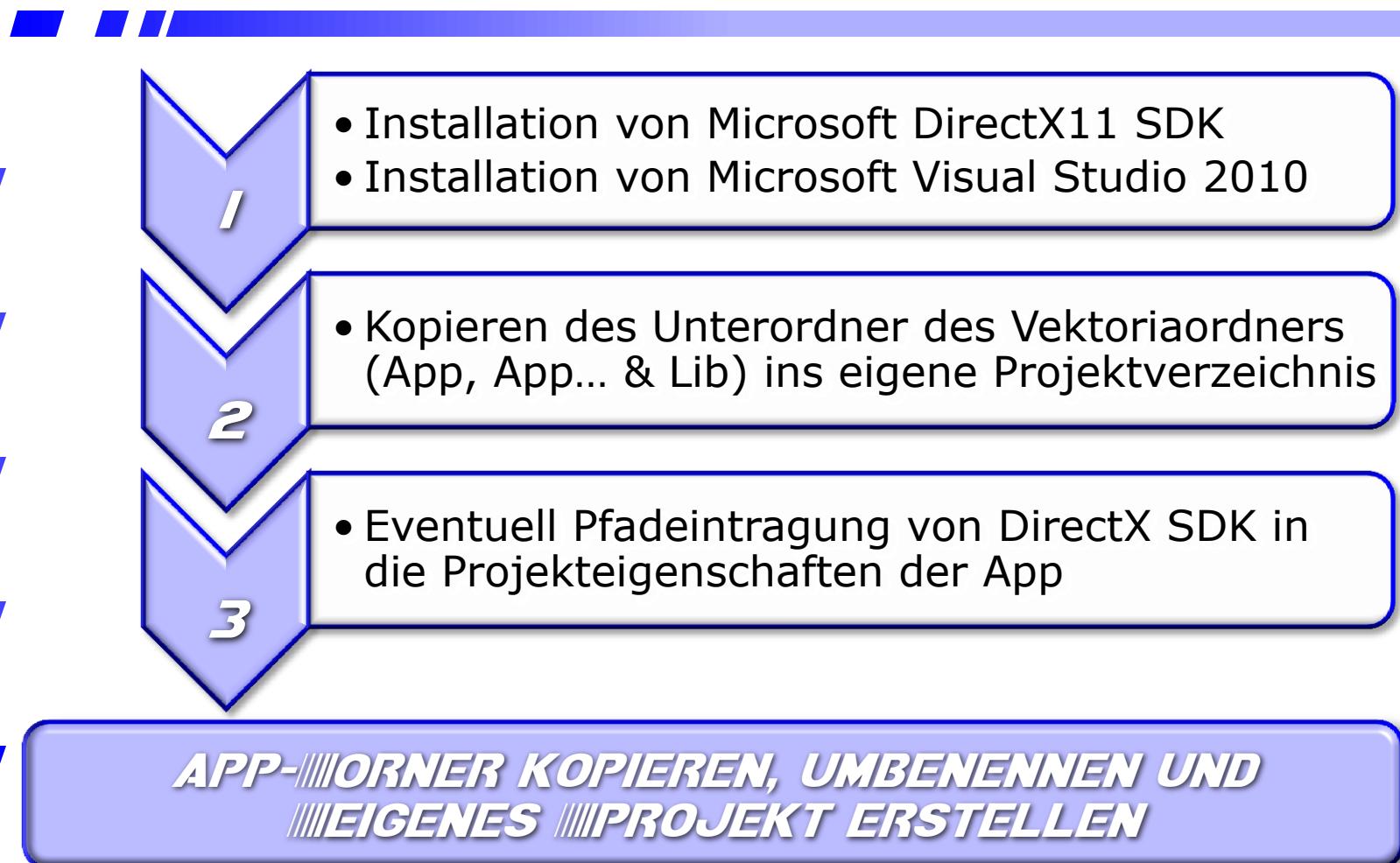


PROF. DR. TOBIAS BREINER
VEKTORIA MANUAL

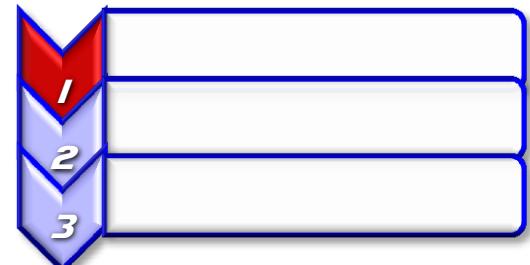
36 VON 49
VEKTORIA EINFUEHRUNG



Installationsschritte



Installationsschritt 1



VEKTORIA V4APP-ORNER KOPIEREN, UMBENENNEN
UND EIGENES PROJEKT ERSTELLEN

- Installation von Microsoft Visual Studio 2013
- Installation von Microsoft DirectX11 SDK

Achtung Nr. 1: Reihenfolge beachten!

Erst das DirectX11 SDK und dann erst das Microsoft Visual C++ installieren.



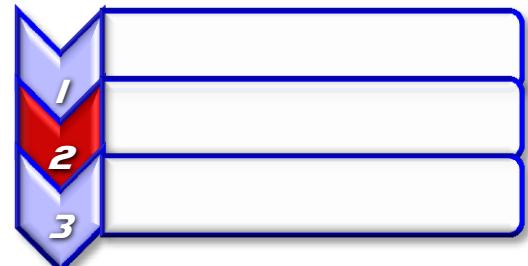
Achtung Nr. 2: Möglichst nicht die Express-Version von Microsoft Visual C++ 2013 verwenden (ungetestet), Professional, Premium oder Ultimate gehen dagegen auf jeden Fall.



Achtung Nr. 3: Auf jeden Fall das Software Development Kit (SDK) von DirectX11 wählen und nicht die „normale“ Anwenderversion ohne SDK!



Installationsschritt 2



VEKTORIA V13 APP-ORDNER KOPIEREN, UMBENENNEN
UND EIGENES PROJEKT ERSTELLEN

Kopieren des Inhalts des Vektoriaordner (VektoriaV13_Alpha)
ins eigene Projektverzeichnis:

1 // / / /

Lib

ist die eigentliche Bibliothek mit Library und Schnittstellen-Header

2 // / / /

App

ist eine leere Template-Applikation.
Diese kann man kopieren, umbenennen und seine eigene 3D Anwendung erzeugen.

3 // / / /

AppHalloWelt

ist eine Musteranwendung,
die nur „Hallo Welt“ ausgibt

4 // / / /

AppHalloKugel:

ist eine Musteranwendung, die eine 3D-Kugel ausgibt. Mit ihr kann man u.a. testen, ob die Grafikkarte ausreicht.

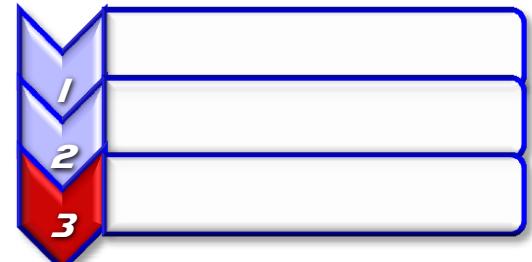
5 // / / /

Die Ordner sollten direkt in diesen Ordner abgelegt werden:
...\\EigeneDokumente\\Visual Studio2013\\Projects\\



Installation von Vektoria

Installationsschritt 3



VEKTORIA V4APP-VERNER KOPIEREN, UMBENENNEN
UND EIGENES PROJEKT ERSTELLEN

Der 3. Installationsschritt setzt die korrekten Pfade in der Template App.

Er ist nur dann notwendig, wenn man DirectX 11 SDK nicht ins Standardpfad installiert hat oder man die Lib von Vektoria in ein anderes Verzeichnis als die App gelegt hat!

Ansonsten kann man Installationsschritt 3 überspringen!

.

4 // / /

5 // / /

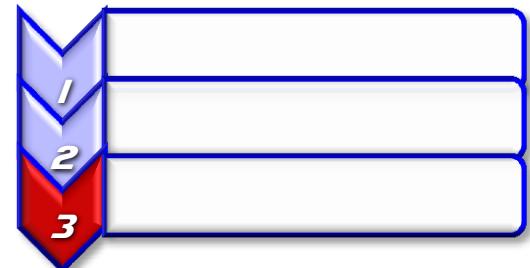


PROF. DR. TOBIAS BREINER
VEKTORIA MANUAL

40 VON 49
VEKTORIA EINFUEHRUNG

Installation von Vektoria

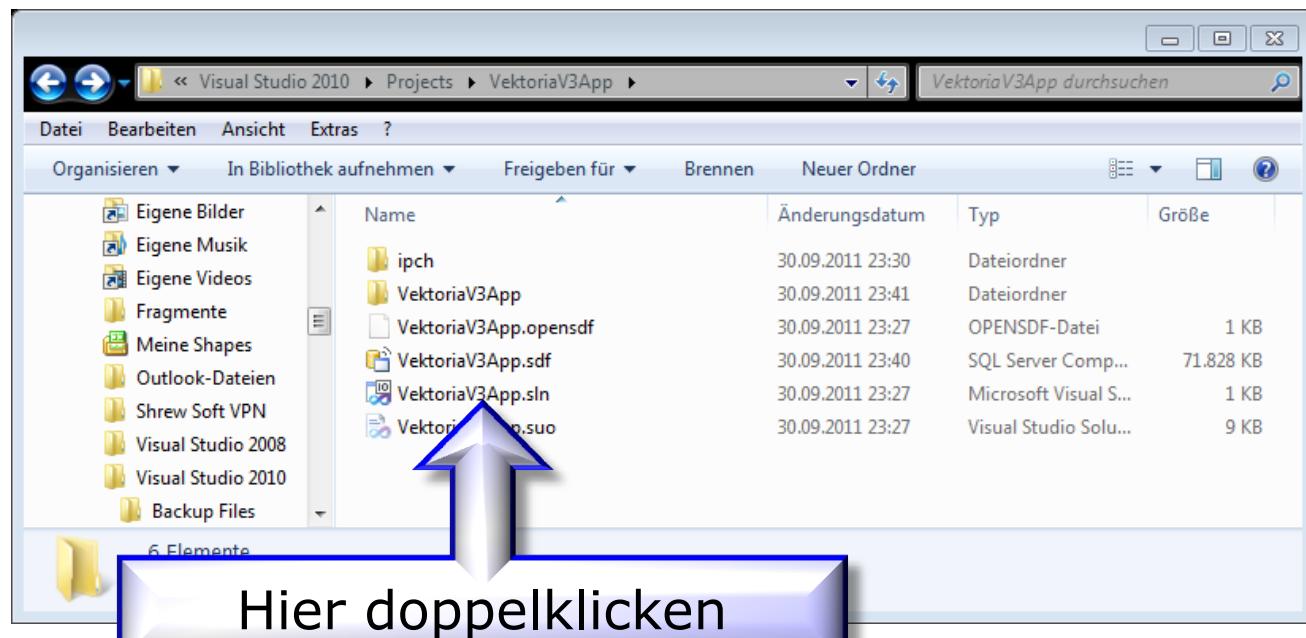
Installationsschritt 3a



VEKTORIA V4APP-ORDNER KOPIEREN, UMBENENNEN
UND EIGENES PROJEKT ERSTELLEN

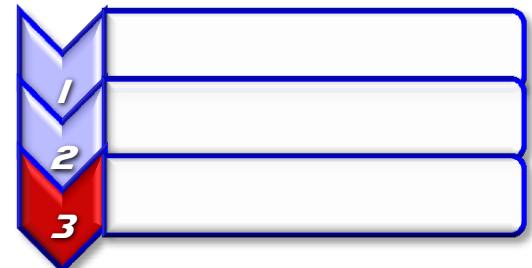
In den Ordner App gehen.

Auf **VektoriaApp.sln** doppelklicken, um Musteranwendung zu öffnen.



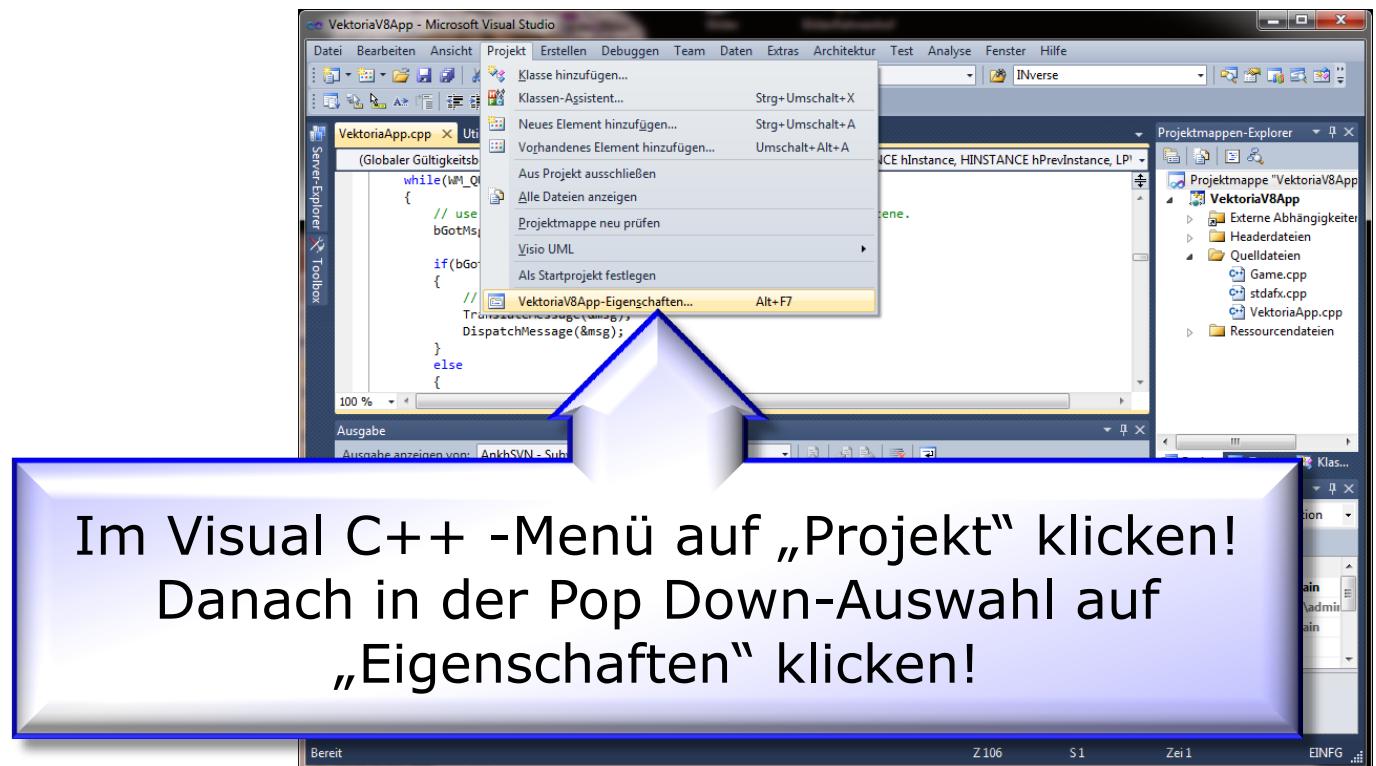
Installation von Vektoria

Installationsschritt 3b



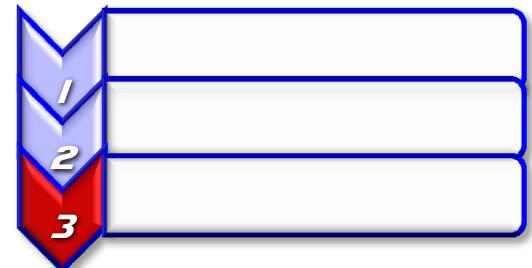
■■■VEKTORIA■■■VBAPP■■■ORNER KOPIEREN,
UMBENENNEN UND ■■■EIGENES ■■■PROJEKT ERSTELLEN

Setzen des Pfades des kopierten Vektoriaordners in den
Projekteigenschaften von Visual C++

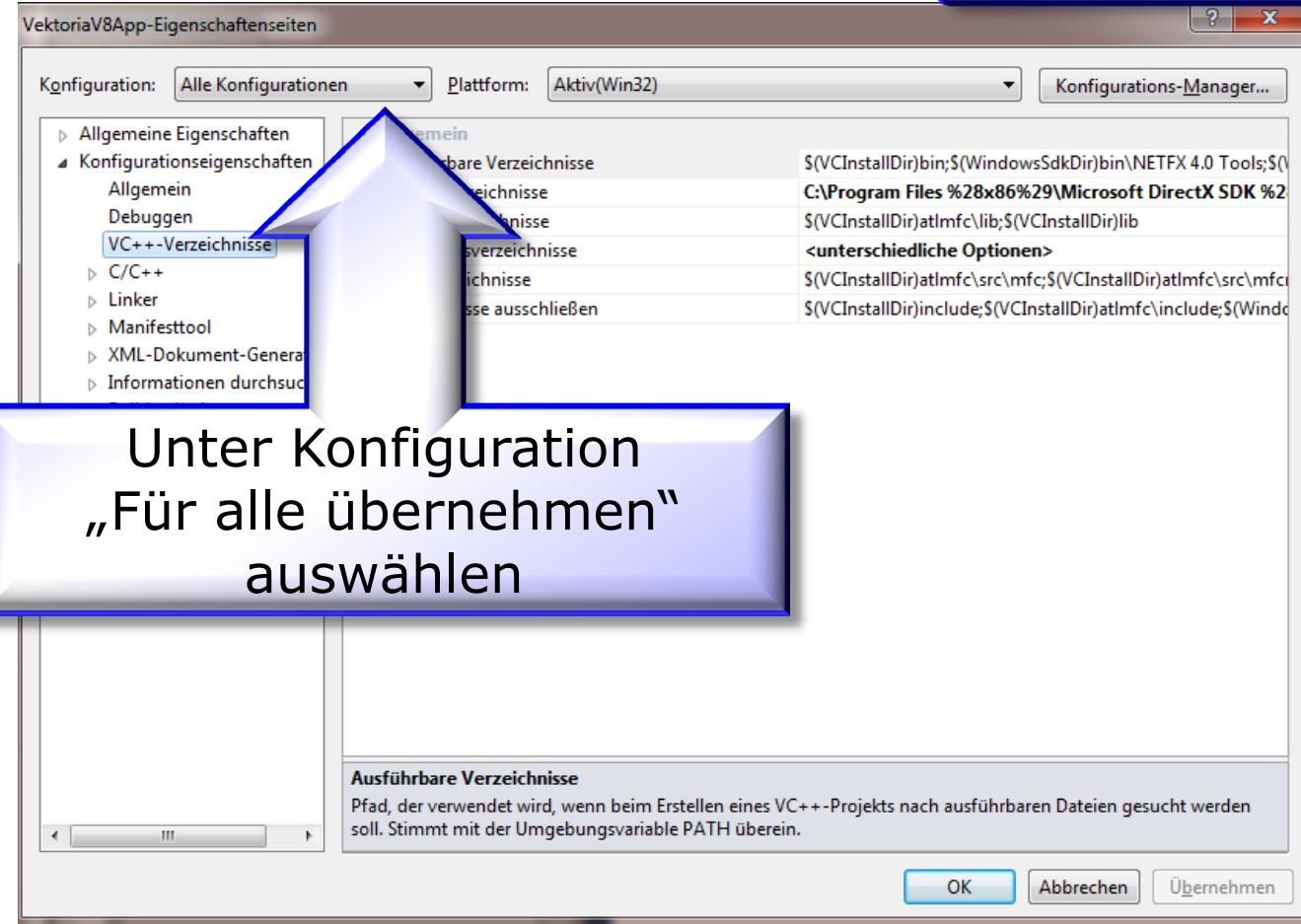


Installation von Vektoria

Installationsschritt 3c

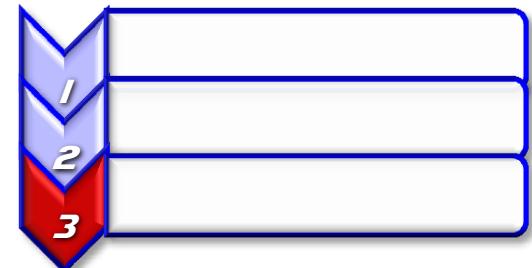


VEKTORIA // VBAPP-ORNER KOPIEREN,
UMBENENNEN UND EIGENES PROJEKT ERSTELLEN

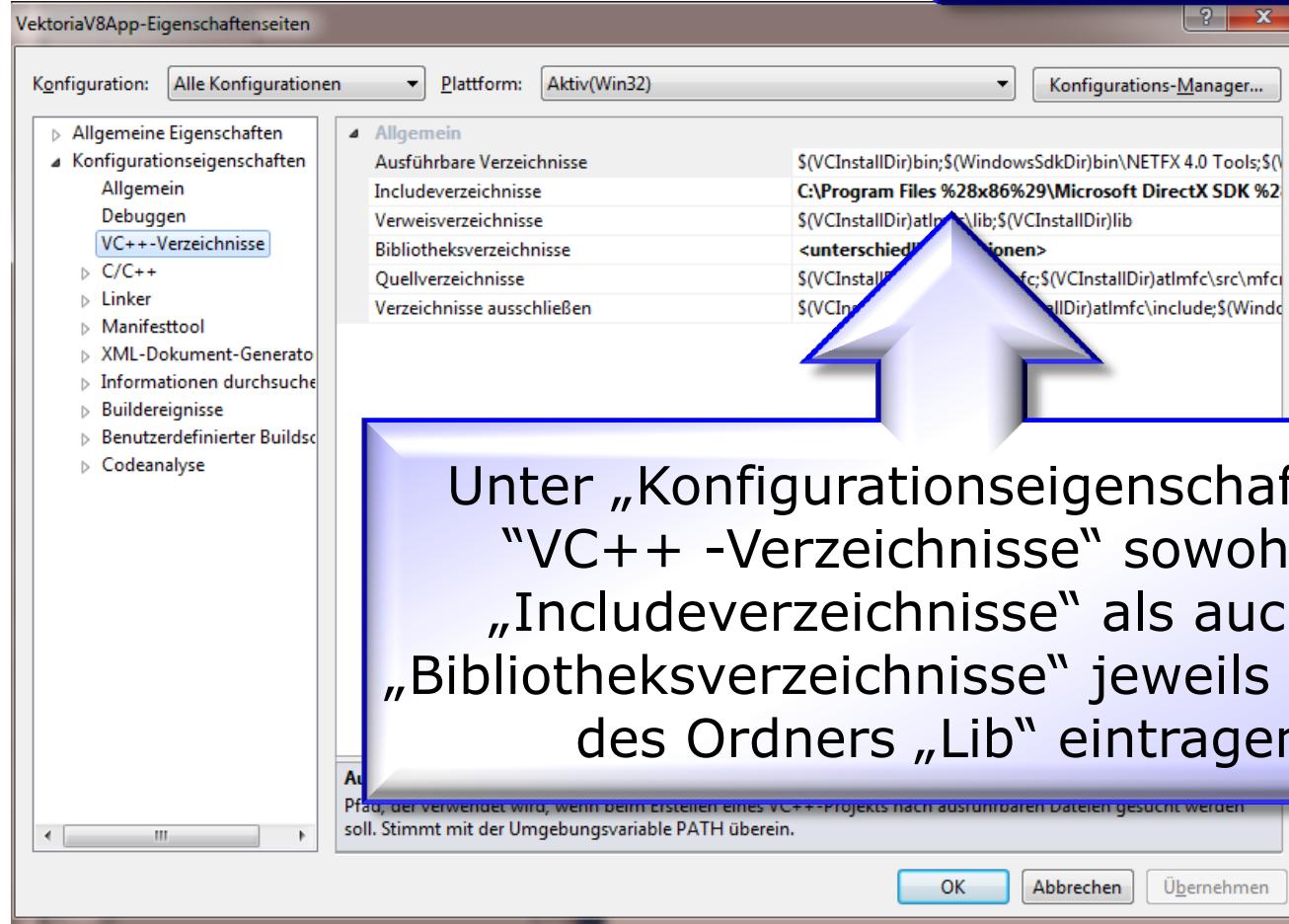


Installation von Vektoria

Installationsschritt 3d



■■■VEKTORIA■■■VBAPP■■■ORNER KOPIEREN,
UMBENENNEN UND ■■■EIGENES ■■■PROJEKT ERSTELLEN

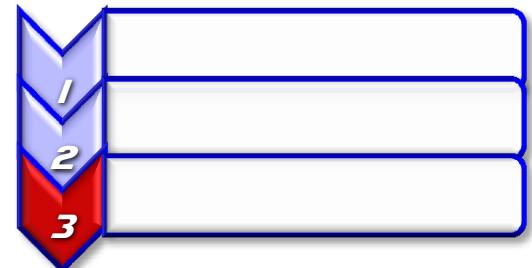


Unter „Konfigurationseigenschaften“ –
„VC++ -Verzeichnisse“ sowohl bei
„Includeverzeichnisse“ als auch bei
„Bibliotheksverzeichnisse“ jeweils den Pfad
des Ordners „Lib“ eintragen!

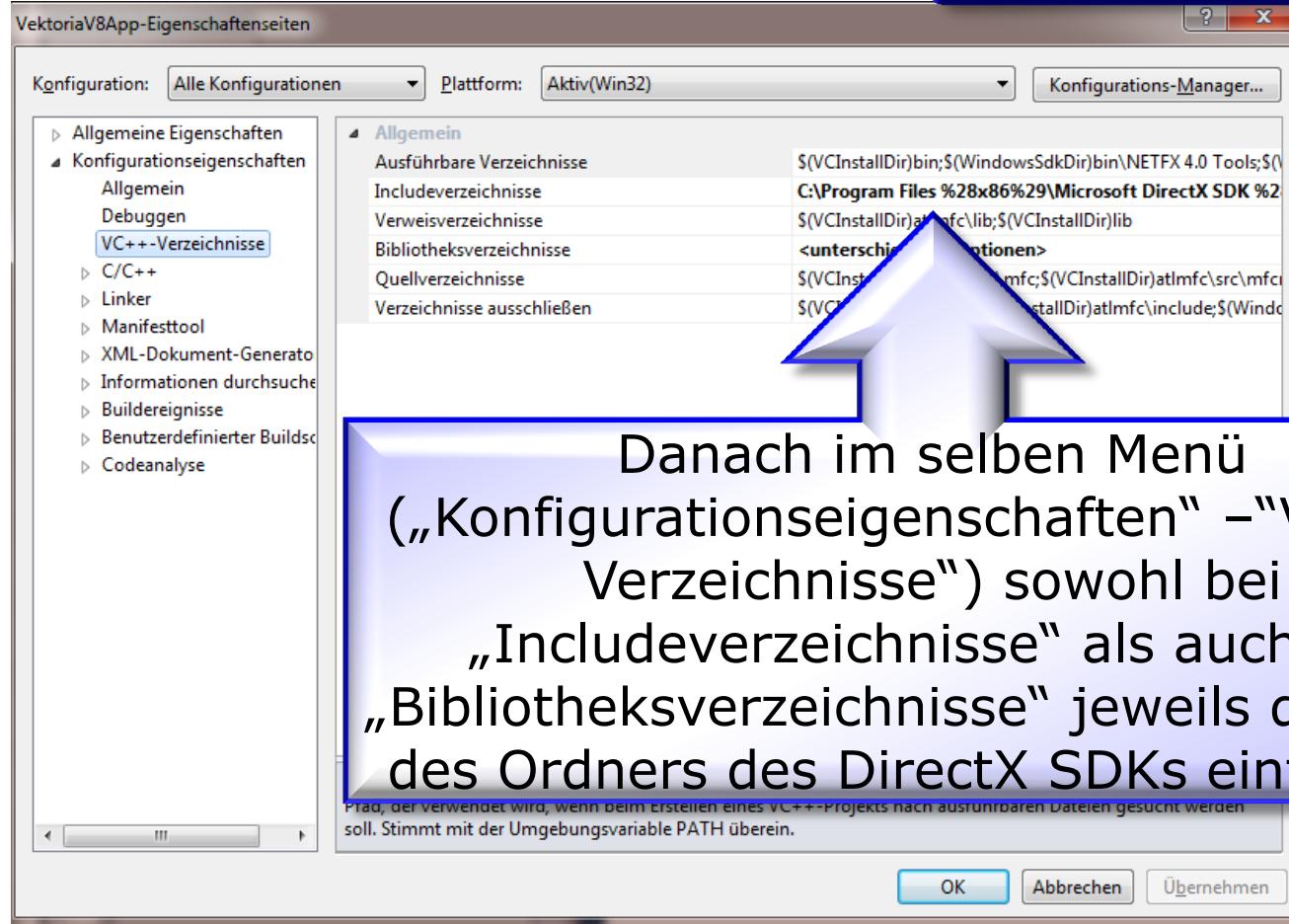


Installation von Vektoria

Installationsschritt 3d



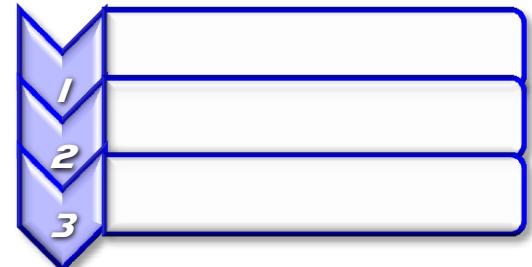
■■■VEKTORIA■■■APP■■■ORNER KOPIEREN,
UMBENENNEN UND ■■■EIGENES ■■■PROJEKT ERSTELLEN



Danach im selben Menü
„Konfigurationseigenschaften“ – „VC++ -
Verzeichnisse“ sowohl bei
„Includeverzeichnisse“ als auch bei
„Bibliotheksverzeichnisse“ jeweils den Pfad
des Ordners des DirectX SDKs eintragen!



Nach der Installation



■■■ VEKTORIA V■■■APP-■■■ORDNER KOPIEREN, UMBENENNEN
UND ■■■EIGENES ■■■PROJEKT ERSTELLEN

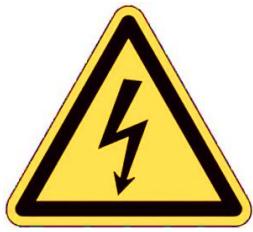
Um eine eigene Vektoria-Applikationen zu erstellen,
kann nun der Musterordner App Verzeichnis kopiert
werden.

Der neue Ordner sollte dann noch sinnfällig
umbenannt werden und Sie können
loslegen!

4 // / /

5 // / /





Portierung V10 => V11, V12, V13

Bei bestehenden Projekten, die von V10 auf V11, V12 oder V13 hochportiert werden, muss beachtet werden, dass bei V11 das Root-Objekt mit dem Parameter „psplash“ initialisiert werden muss, damit etwas angezeigt wird!
Auch die Initialisierung des Frames hat sich geändert.

V10

```
void CGame::Init(  
    HWND hwnd,  
    HWND hwndDX,  
    RECT rectWnd)  
{  
    m_zf.Init(hwnd,  
              rectWnd.right,  
              rectWnd.bottom);  
    ...
```

V13

```
void CGame::Init(  
    HWND hwnd,  
    Csplash * psplash)  
{  
    m_zr.Init(psplash);  
    m_zf.Init(hwnd);  
    ...
```





Portierung V12 => V13

Zusätzlich wurden die Image-Textures in Diffuse-Textures umbenannt, daher müssen alle diesbezüglichen Methoden umgeschrieben werden.

1 // / / /

2 // / / /

3 // / / /

4 // / / /

5 // / / /

V12 bzw. V13 Beta

MakeTextureImage
MakeTextureGlowAsImage
SetTextureImage
...

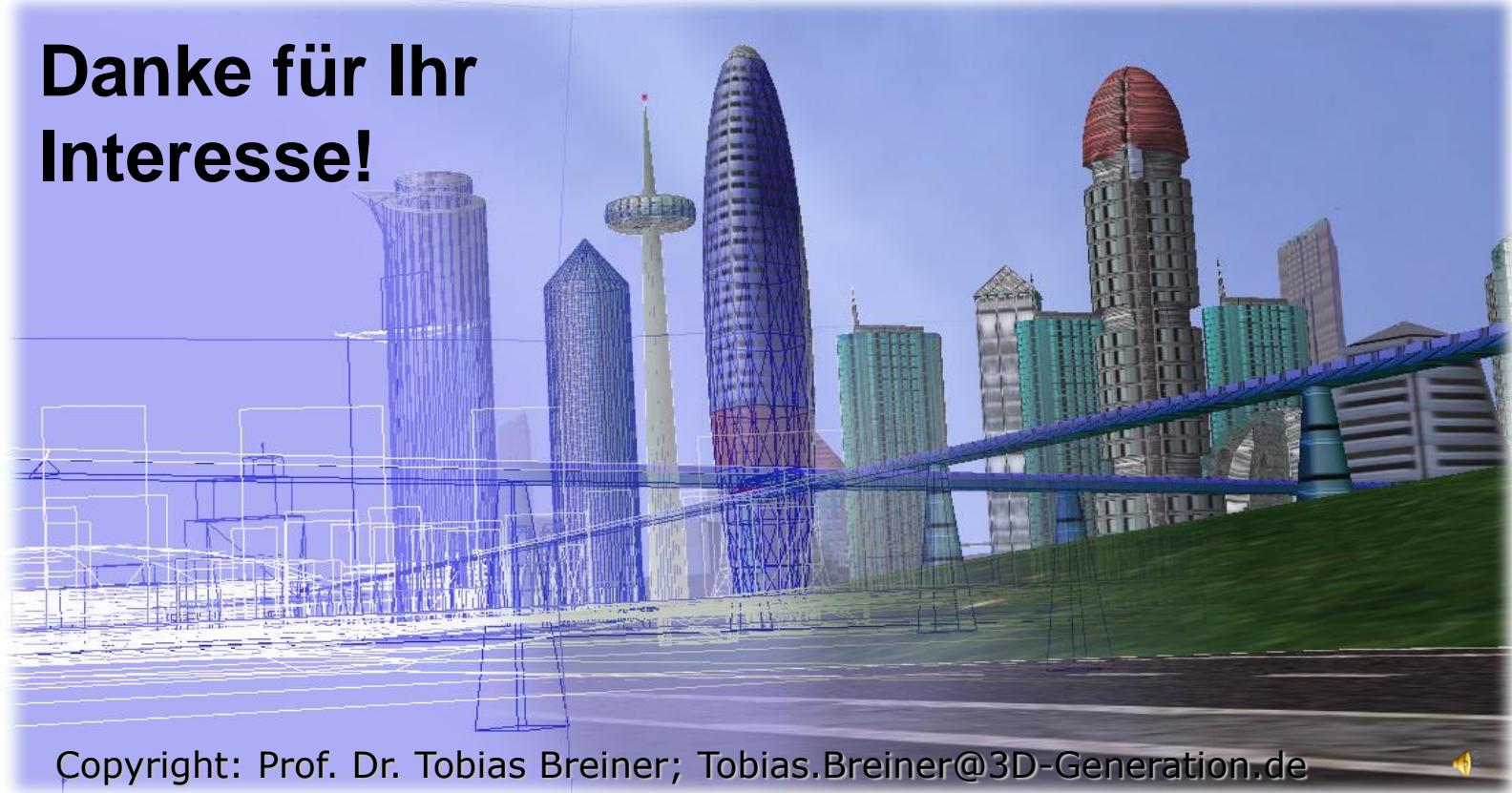
V13

MakeTexturediffuse
MakeTextureGlowAsDiffuse
SetTextureDiffuse
...



|||||GAME OVER

Danke für Ihr
Interesse!



Copyright: Prof. Dr. Tobias Breiner; Tobias.Breiner@3D-Generation.de

