



**LVA: 186.831 UE Computergraphik  
Abgabe 2 - Dokumentation**

**Dario Sommer (11777720)  
Maximilian Johannes Zenz (01633058)**

### ENTWICKLUNGSSTATUS:

Das Spiel ist fertig implementiert.

### FEATURES:

- 2D-Map: Details siehe „Implementierung“
- Punktestands-Panel, welches sich mit Pacman bewegt

### STEUERUNG:

W, A, S, D	Bewegen von Pacman in horizontaler Ebene
Maus	Ansicht drehen/Pacman steuern
linke Maustaste	Zurück schauen
I/O	Ändern der Helligkeit
ESC	Schließen des Spiels
F3	Aktivieren/Deaktivieren des Vollbildmodus
H	HUD anzeigen/nicht anzeigen
J/K	Subdivisions von Pacman ändern

### IMPLEMENTIERUNG:

- *Labyrinth*: Größe kann durch Konstante SIZE in der Klasse Maze gesetzt werden. Wird zufällig generiert. Algorithmus eine Art Tiefensuche.
- *PhysX*: Nvidia PhysX wurde eingebunden nach Anleitung auf TUWEL um eine physikalische Abhängigkeit der Spielfigur zu gewährleisten. Durch einsetzen der PhysX-Engine wurde gewährleistet, dass die Spielfigur nicht durch Wände fahren kann sowie dass die Herzen physikalisch korrekt auf den Boden fallen und dort liegen bleiben.
- *OBJ-Loader*: Einfacher OBJ-Loader wurde implementiert damit Obj-Dateien die durch Blender generiert wurden einfach integriert werden können.
- *Spielsteuerung*: Die Spielsteuerung wurde einfach über eine PlayerController Klasse realisiert der man das Spielerobjekt und die Tasten übergibt und anschließend das PhysX-Objekt und das OpenGL-Objekt des Spielers bewegt wird.
- *Münzen*: Münzen können über eine CoinController Klasse entweder einzeln oder in Reihen erstellt werden.
- *Geistersteuerung*: Es wird jedem Geist einen Controller zugewiesen dem Koordinaten übergeben werden, wonach der Geist die zugewiesenen Koordinaten nachfährt. Hat der Geist weniger als einen Punkt zum Nachfahren, so berechnet sich dieser einen neuen. Es wird geschaut, in welcher Richtung sich der Geist bewegt (um zu verhindern, dass dieser direkt umdreht) und auf welchen Feldern sich Wände befinden. Ist der Spieler nicht in der Nähe, so ist die Auswahl rein zufällig. Ab einem bestimmten Abstand wählen die Geister immer das Feld, welches sich am nächsten zum Spieler befindet.



## IMPLEMENTIERUNG:

- *Subdivision Surfaces*: Es wurde in die OBJ-Loader-Klasse ein Loop Subdivision Algorithmus implementiert nach den Folien aus der VO Computergrafik und ein wenig Recherche im Internet. Standardmäßig wird dieser in Laufzeit nur auf alle Geister und Pacman angewendet da es bei den Münzen und den geraden Wänden wenig Sinn gemacht hätte, einen solchen Algorithmus anzuwenden. Weiter kann man während des Spielens die Subdivisions der Spielfigur mittels Tasteneingabe höher oder niedriger einstellen was allerdings mit geringen Ladezeiten verbunden ist.
- *Video textures*: Anfangs werden alle Frames der Video Textur eingelesen und danach werden sie nach einer bestimmten Zeit umgeschaltet. Da wir nur sehr langsam zwischen den Frames umschalten wollen, haben wir absichtlich keine 25 fps gewählt da wir sonst unnötig viele gleiche Bilder eingelesen hätten.
- *Cel Shading*: Beim Cellshading oder Comic Shading handelt es sich um eine Shading Methode, bei der der Helligkeitsverlauf nicht dynamisch, sondern in Abstufungen erfolgt. Hierzu haben wir zuerst den diffusen Anteil der Farbe in den HSV Farbraum umgerechnet danach die berechnete Helligkeit am Punkt mittels der Floor Funktion gestuft und wieder in RGB umgerechnet.
- *Contours via edge detection*: Die Kanten Erkennung findet rein im Shader statt und ist dafür zuständig, dass die Außenlinien der Spielobjekte besser erkennbar sind. Hierfür wird mittels der Objektnormalen berechnet, wann ein Objekt eine Kante besitzt die vom Spieler wegschaut und diese schwarz gezeichnet. Leider passiert hier der Effekt das auch Objekte, die sehr flach betrachtet werden als Kante gezählt werden.

## FEATURES:

- **Verpflichtend:**
  - Playable
  - 3D Geometry
  - Win/Loose Condition
  - Intuitive Controls
  - Intuitive Camera
  - Moving Objects
  - Documentation
  - Adjustable Parameters (Auflösung und Framerate durch Configfile assets/settings, Fullscreen und Brightness durch F3 bzw. I/O änderbar)
- **Optional:**
  - Physics Engine
  - Heads-Up Display (Mini Map)
- **Effekte:**
  - Subdivision Surfaces
  - Hierarchical Animation (Münze vom Punktestand-Panel)
  - Video textures (Jumppads)
  - Cel Shading
  - Contours via edge detection

## BELEUCHTUNG:

Ein Point-Light welches sich hinter Pacman befindet beleuchtet alles was sich im Lichtkegel befindet. Sonst gibt es keine Lichtquelle. Die Mini Map wird durch das Ambient Light beleuchtet und benötigt daher keine eigene Lichtquelle.

## TEXTUREN:

DDS-Texturen durch UV-Koordinaten auf den Objekten angebracht

## ZUSÄTZLICHE BIBLIOTHEKEN:

- Nvidia PhysX 4.1

