

Abschlusspräsentation

- 3D Softwareprojekt -

Underlord

Gliederung

- **Spielprinzip**
- **Projektmanagement**
 - Zielsetzung
 - Aufgabenverteilung
 - Meilensteine I bis VI
- **Technische Umsetzung**
- **Fazit**
- **Demo**

Spielprinzip

- Hexagonbasiertes 3D-Strategiespiel
- Dungeon bauen
- Heldenwellen abwehren
- Spielende
- Highscore

Projektmanagement

- **Zielsetzung:**
 - Lauffähiges Spiel
 - Referenz
 - Gute Note

Projektmanagement

- **Zielsetzung:**

- Lauffähiges Spiel
- Referenz
- Gute Note

- **Aufgabenverteilung:**

- Dominik: Daten- und Verwaltungsstrukturen sowie Steuerung
- Patrick: Logik
- Max: Animation und Darstellung
- Ursprünglich noch Mareen Johannes: Modellierung und GUI

Projektmanagement

- **Meilenstein I (22.05):**
 - Vorstellung des TDD
 - Organisatorische Aufbauarbeit:
 - Austausch von Skype und Emaildaten
 - Einrichten eines Redmine-Servers
 - Termin für wöchentliche Treffen finden
 - Erste Machbarkeitstests mit Hexagonfeldern und Mauselektion
 - Einarbeitung in XNA-Animation

Projektmanagement

- **Meilenstein II (12.06):**
 - Patrick und Dominik:
 - *Hexagon*-, *Room*- und *Map*-Klasse sowie erste *Nest*- und *Creature*-Klasse
 - Mapgenerator (Patrick)
 - *Interaction*-Klasse (Dominik)
 - Max:
 - Grundstruktur des Animationsystems
 - Überarbeitung und Einbindung der Modelle
 - Mareen:
 - Modellierung des Ameisen-Nests
 - Probleme:
 - Technische Schwierigkeiten bei Aufsetzung des Redmine-Servers

Projektmanagement

- **Meilenstein III (10.07):**

- Patrick:

- *Creature-AI* und *Imp-AI*-Klasse (noch ohne *Jobsystem*)

- Dominik:

- Erweiterung der *Interaction*-Klasse um u.a. Hotkey-Steuerung

- Max:

- Anpassung des Animationsystems

- Modellierung einiger Gebäude, Imps und Helden

- Mareen:

- Minimap

- Probleme:

- Mareens Projektbeteiligung nahm ab

Projektmanagement

- **Meilenstein IV (25.07):**

- Patrick und Dominik:

- Erstes Balancing
 - Beginn mit *GUIElement*-Klasse (Dominik)
 - Fertigstellung der *AI*-Klasse (Patrick)

- Max:

- Komplettierung und Einbindung der Modelle
 - Überarbeitung des Helden-Modells

- Probleme:

- Umorganisation und Verteilung der Aufgaben nach Mareens Ausscheiden
 - Radikale Kürzung des Spielumfangs

Projektmanagement

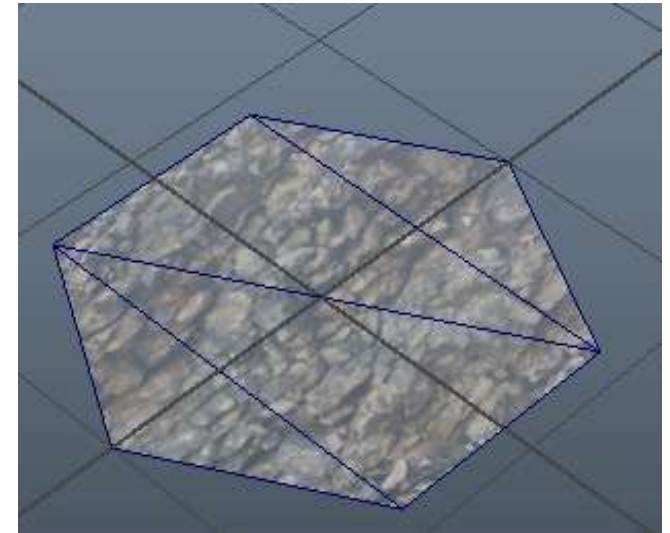
- **Meilenstein V (15.08):**
 - Patrick:
 - Überarbeitung der Minimap
 - *Spell*-Klasse
 - Dominik:
 - Erstellung der *GUI*-Klasse
 - Fertigstellung der *Upgrade*-Klasse
 - Max:
 - Vollständige Verknüpfung der Animationslogik mit dem *AI*-System
 - Einfaches Beleuchtungssystem
 - Probleme:
 - Flackern des Spiel-Screens

Projektmanagement

- **Meilenstein VI (06.09 eigentlich 02.09):**
 - Patrick und Dominik:
 - Finale Balancing-Phase
 - Implementierung der *Wavecontroller*- und der *Player*-Klasse (Patrick)
 - Fertigstellung der *GUI* und Einbettung der Hilfstexte (Dominik)
 - Max:
 - Fertigstellung sämtlicher GUI-Grafiken und Überarbeitung der *GUI*-Klassen
 - Erstellung des Feuerball-Modells
 - Probleme:
 - Maxs krankheitsbedingter Ausfall auf Grund einer schweren Lebensmittelvergiftung
 - Zeitliche Schwierigkeiten und Abstimmungsprobleme
 - Kleinere Unsauberkeiten am Ende

Technische Umsetzung

- **Grundlegende Datenstrukturen:**
 - *Hexagon-Klasse:*
 - Zentrales Datenelement
 - Kennt Nachbarn



Technische Umsetzung

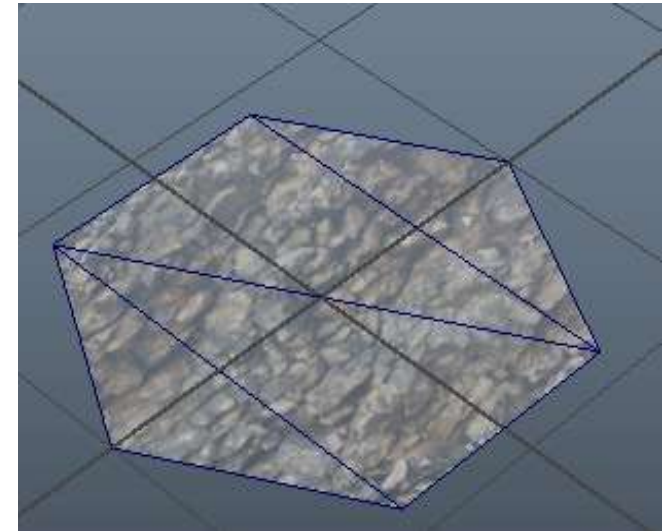
- **Grundlegende Datenstrukturen:**

- *Hexagon-Klasse:*

- Zentrales Datenelement
 - Kennt Nachbarn

- *Map-Klasse:*

- Torusartige Struktur
 - Zentrales Verwaltungselement
 - Mapgenerator



Technische Umsetzung

- **Grundlegende Datenstrukturen:**

- *Hexagon-Klasse:*

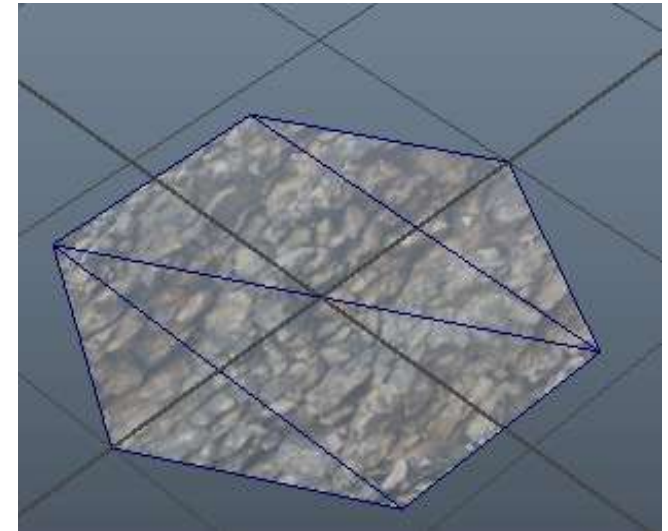
- Zentrales Datenelement
 - Kennt Nachbarn

- *Map-Klasse:*

- Torusartige Struktur
 - Zentrales Verwaltungselement
 - Mapgenerator

- *Thing-Klasse:*

- Objekt auf Hexagon
 - Verschiedene Typen



Technische Umsetzung

- **Spiellogik:**
 - *Interaction*-Klasse:
 - Behandlung von Spielereingaben
 - Gamestates

Technische Umsetzung

- **Spiellogik:**
 - *Interaction*-Klasse:
 - Behandlung von Spielereingaben
 - Gamestates
 - *AI*-Klasse:
 - Imp- und Creature-AI
 - Imp-Jobsystem
 - Freund/Feinderkennung
 - Wegeberechnung

Technische Umsetzung

- **Spiellogik:**

- *Jobsystem:*

- 6 Jobtypen
 - 3 Listen/Queues in Map
 - Automatisches Aufschieben

Technische Umsetzung

- **Darstellung:**

- *BasicModel:*

- Kapselt statische Modelle (z.B. Wände, Nester) und erweitert diese um *Update*- und *Draw*-Methode
 - *Draw* arbeitet mit *XNA-BasicEffect*

Technische Umsetzung

- **Darstellung:**

- *BasicModel:*

- Kapselt statische Modelle (z.B. Wände, Nester) und erweitert diese um *Update*- und *Draw*-Methode
 - *Draw* arbeitet mit *XNA-BasicEffect*

- *CharacterModel:*

- Ableitung aus der *BasicModel*-Klasse
 - Kapselt dynamische Modelle mit Animationsdaten (z.B Imps)
 - Implementiert die *ClipPlayer*-Klasse zur Realisierung der Animation
 - *Draw* arbeitet mit:
 - *BasicEffect* (statischer Modelteil)
 - *SkinnedEffect* (dynamischer Modelteil)

Technische Umsetzung

- **Animation:**

Animationssystem setzt sich aus folgenden Klassen zusammen:

Technische Umsetzung

- **Animation:**

Animationsystem setzt sich aus folgenden Klassen zusammen:

- *Animation-Import-Pipeline*-Klasse:
 - Kernstück des Animationsystems
 - Importiert die Skelett-Animationsdaten
 - Ableitung der *ModelProcessing*-Pipeline

Technische Umsetzung

- **Animation:**

Animationsystem setzt sich aus folgenden Klassen zusammen:

- *Animation-Import-Pipeline*-Klasse:
 - Kernstück des Animationsystems
 - Importiert die Skelett-Animationsdaten
 - Ableitung der *ModelProcessing*-Pipeline
- *Clip*-Klasse:
 - Datenstruktur zum Speichern der importierten Animationsdaten (*Keyframes* und *Bone*-Objekte im Model)

Technische Umsetzung

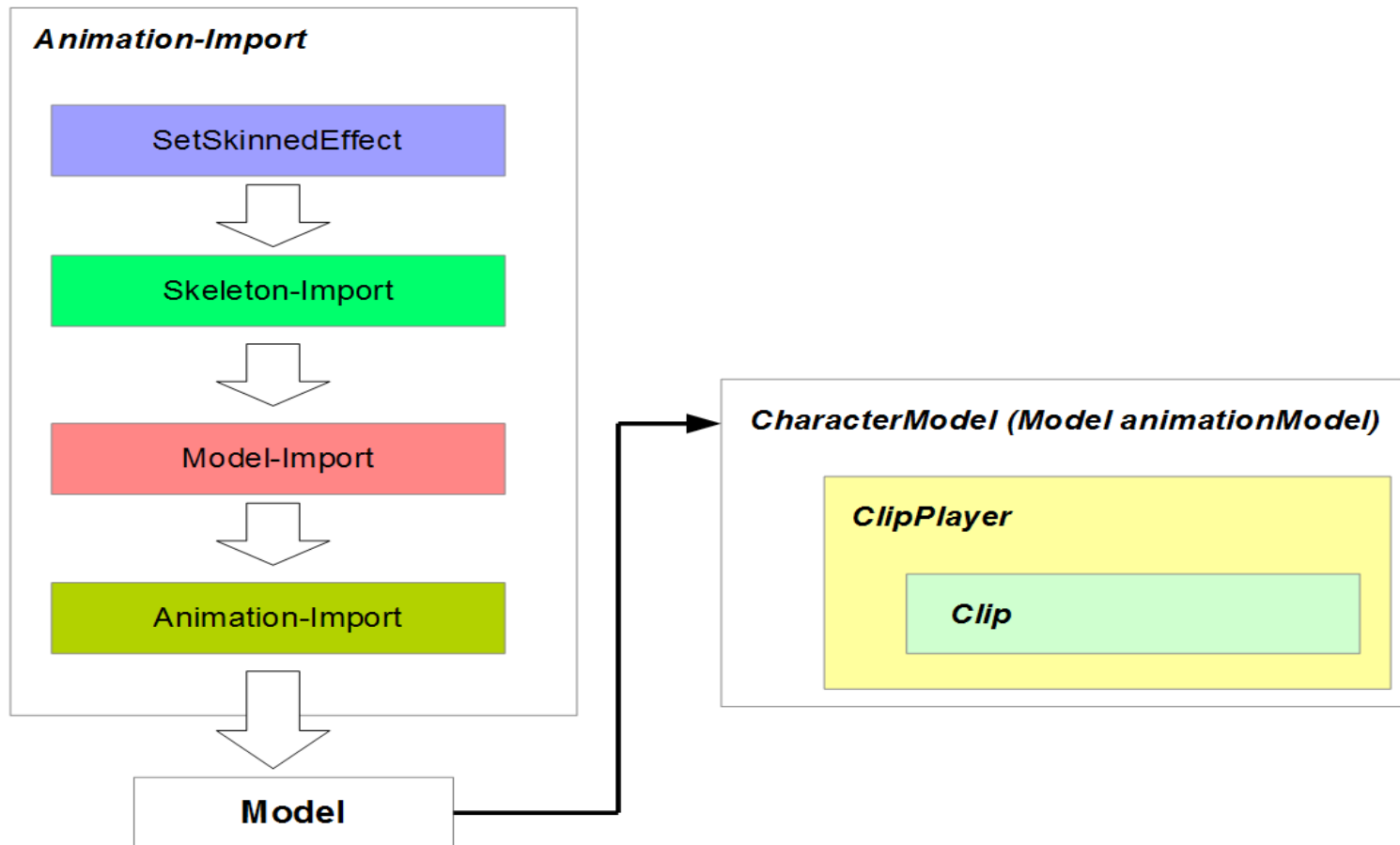
▪ Animation:

Animationsystem setzt sich aus folgenden Klassen zusammen:

- *Animation-Import-Pipeline*-Klasse:
 - Kernstück des Animationsystems
 - Importiert die Skelett-Animationsdaten
 - Ableitung der *ModelProcessing*-Pipeline
- *Clip*-Klasse:
 - Datenstruktur zum Speichern der importierten Animationsdaten (*Keyframes* und *Bone*-Objekte im Model)
- *ClipPlayer*-Klasse:
 - Spielt den *Clip* ab
 - Clip-Position bestimmt Zeitpunkt im *Clip* und damit 2 *Keyframes*
 - Lineare Interpolation dieser liefert Orientierungsmatrix für Model-*Bones*

Technische Umsetzung

- Darstellung und Animation:



Fazit

- **Stolperstein:**
 - Arbeitsaufwand unterschätzt
 - Ausscheiden eines Teammitglieds
 - Technische Probleme mit Redmine-Server
 - Krankheitsbedingter Ausfall

Fazit

- **Stolperstein:**

- Arbeitsaufwand unterschätzt
- Ausscheiden eines Teammitglieds
- Technische Probleme mit Redmine-Server
- Krankheitsbedingter Ausfall

- **Erfolge:**

- Kernelemente vollständig
- Lauffähiges Spiel
- Zufriedenstellendes Ergebnis



Spieldemo

Underlord

Abschlusspräsentation

- 3D Softwareprojekt -

Underlord