Maturarbeit

Entwicklung eines Computerspiels mit Unity

# Einleitung

# Die Spiele-Engine Unity

## Die Unity Physik-Engine

### Das Dynamic Tree Modul

Ein Spiel besteht in der Regel aus sehr vielen Objekten. Für den Spielverlauf muss häufig die Beziehung zwischen Objekten geprüft werden, zum Beispiel zur Prüfung:

* ob zwei Objekte miteinander kollidieren,
* ob ein Objekt aus Perspektive der Spielfigur sichtbar ist,
* oder zum Finden des Bodens.

Für die Berechnung werden sogenannte Casts verwendet. Die Casts kann man sich als Linien (Raycast) oder Flächen (Boxcast) vorstellen. Für 3D-Spiele gibt es auch räumliche Casts (Spherecast), was aber nicht Teil diese Arbeit ist. Mit den Casts wird geprüft, ob sich diese mit den Objekten schneiden.

<TODO> Graphik mit Objekten und einem Raycast

Die Berechnung von Schnittpunkten komplizierter Objekt-Formen wäre allerdings langsam, daher werden die Objekte durch rechteckige Rahmen eingegrenzt. Für jedes Objekt wird mit den oberen und unteren Ecken sogenannte Axis Aligned Bounding Boxes (AABB) berechnet.

<TODO> Graphik mit Objekten, den AABBs und einem Raycast

Je mehr Objekte vorhanden sind desto länger würde die Prüfung dauern, wenn jedes Objekt einzeln berechnet wird (Brute Force Vorgehen). Um die Effizienz der Berechnung zu verbessern, werden die AABBs der Objekte daher in Gruppen zusammengefasst. Wenn ein Raycast eine AABB-Gruppe trifft, wird die Berechnung innerhalb dieser Gruppe fortgeführt, wodurch insgesamt weniger Berechnungen benötigt werden.

<TODO> Graphik für AABB-Gruppen

Für die Erstellung der AABB-Gruppen werden die AABBs von jeweils zwei Objekten in ein neues AABB zusammengefasst. Dieser Vorgang wird wiederholt, bis eine AABB-Gruppe übrig ist, die alle Objekte beinhaltet. Die AABBs der Objekte und der Objektgruppen werden in einem Binärbaum, einer sogenannten Bounding Volume Hierarchy (BVH) gespeichert.

<TODO> Graphik eines Bounding Volume Hierarchy (Folie 18)

#### Such-Algorithmus mit Bounding Volume Hierarchy

Durch die Bounding Volume Hierarchy können zuerst die grössten AABBs kontrolliert werden, wodurch bereits viele Objekte ausgeschlossen werden können, falls eines der AABBs an der Spitze der Hierarchie nicht vom Raycast getroffen wird. Dieser Vorgang wird wiederholt bis die AABBs nur noch einzelne Objekte enthalten. Anschliessend wird die Berechnung für die tatsächliche Objektform durchgeführt.

#### Erstellung der Bounding Volume Hierarchy

Es gibt drei Varianten für die Erstellung der Bounding Volume Hierarchy:

* Bottom Up
* Top Down
* Incremental

Bei dem Bottom Up Vorgehen werden die Objekte mit dem jeweiligen Nachbar-Objekt in eine Gruppe zusammengefasst. Danach wird dieser Vorgang mit den AABBs der neu erstellten Gruppen wiederholt, bis nur noch eine AABB übrig ist. Mit dem Bottom Up Vorgehen werden alle Objekte gleichmässig in der Bounding Volume Hierarchy verteilt.

Bei dem Top Down Verfahren wird zuerst die übergreifende AABB erstellt, die alle Objekte beinhaltet. Danach wird die übergreifende Gruppe wiederum in zwei Gruppen geteilt. Für die Teilung gibt es zwei Möglichkeiten. Die AABB-Gruppe wird entweder in der Mitte aufgeteilt oder es werden die zwei AABBs mit der geringsten Oberfläche gesucht. Bei der ersten Variante kann die Unterteilung durch die einfachere Berechnung effizienter durchgeführt werden, die dadurch entstehenden Gruppen sind jedoch sehr unterschiedlich. Bei der zweiten Variante dauert das Erstellen der Bounding Volume Hierarchy länger, die entstehenden Gruppen sind jedoch bei der späteren Berechnung effizienter. Je kleiner die Fläche der AABBs ist, desto geringer ist die Wahrscheinlichkeit, dass ein Raycast das AABB trifft.

Das inkrementale Verfahren fängt mit einem einzelnen Objekt im BVH an und es werden die weiteren Objekte eingefügt und mit einem anderen Objekt in eine AABB zusammengefasst. Der dadurch entstehende Baum ist meistens ineffizient und kann durch einen Algorithmus verbessert werden.

<TODO> Der Algorithmus zum Einfügen weiterer Objekte

Wenn ein neues Objekt in die BVH eingefügt werden soll, wird mit einem der anderen Objekte eine neue Node mit AABB gebildet welche die beiden Objekte enthält. Die darüberliegenden AABBS werden so angepasst das sie da das neue Objekt einschliessen. Welche der Objekte die beste Möglichkeit ist, wird durch die Surface Area Heuristic mithilfe der Einfüge Kosten berechnet.

<TODO> Die Surface Area Heuristic (SAH) zur Berechnung der Einfüge Kosten

Die Einfüge Kosten für jede node wird mit der Oberfläche der AABBs berechnet. Weil es sehr ineffizient ist, jede einzelne einfüge Möglichkeit zu überprüfen wird die Surface Area Heuristic (SAH) verwendet. Dazu wird die Fläche der neu hinzugefügten Node und die Anpassungen der darüberliegenden Nodes addiert.  
<TODO> Bsp Formel mit Bild   
Es werden die Kosten aller Nodes auf einer ebene Ausgerechnet und wenn die neu angepasste Node besser als die davor Beste ist wird sie als neue beste Node eingetragen. Dieses vorgehen wird bei den ???? der besten Node der Ebene wiederholt. Wenn die beste Node bestimmt ist wird kontrolliert, ob es überhaupt möglich ist das die darunterliegenden Nodes effizienter sind. Dafür wird überprüft, ob die Flächenänderung der darüberliegenden AABBs plus das AABB des neuen Objekts weniger kosten als die aktuellen besten Kosten.  
<TODO>grafische Erklärung und Formel

# Die Entwicklung von «Das Spiel»

# «Das Spiel»

# Nachwort

# Quellenverzeichnis