

# Notizen Kapitel 7: Collisions and Health: Detecting Collisions Accurately

## *Physikalische Eigenschaften konfigurieren*

Wir erinnern uns: Objekte haben zwei Arten von Formen, einerseits die sichtbare Form (**visual shape**), unter Umständen mit vielen optischen Details, andererseits die physikalische Form (**collider**) mit einer in aller Regel möglichst einfachen Gestalt, um die Performance des Games zu steigern. Unity bietet verschiedene Arten von einfachen Collidern an: **Box**, **Sphere**, **Capsule**. Collider von komplizierten Objekten sollten nach Möglichkeit aus derart einfachen Collidern zusammengesetzt sein – Beispiel Flugzeug: ein Capsule-Collider für den Rumpf und vielleicht ein Box-Collider für die Flügel.

Sollte dieses Verfahren aber nicht gehen, würden wir einen sogenannten **Mesh Collider** (Gitter) benötigen. Darauf werden wir allerdings verzichten.

Wir möchten die Mauern unserer Festungsplattform mit einem Collider versehen:

- Im Prinzip wäre es sinnvoll, zuerst vom Grund-Mauerelement, das momentan noch über keine Collider-Komponente verfügt, einen Prefab zu erstellen und danach aus diesen Prefabs die Plattform zusammenzustellen. Nun haben wir aber die Plattform bereits erstellt, sodass diese Vorgehensweise zu mühsam wäre...
- Stattdessen können wir alle Mauern in der Hierarchy View anwählen und so auf einen Schlag mit Collidern versehen.
- Ebenso versehen wir den Player mit einem Collider. Am einfachsten eine Capsule, was zu unproblematischem Verhalten in Ecken oder auf Rampen führt. Beim Einfügen muss man die Collider-Hülle allenfalls noch anpassen (Grösse und Position).
- Bezüglich Collider-Eigenschaften sollen sich die Objekte verschieden verhalten:
  - Objekte mit **«statischem Collider»**: Keine weiteren physikalischen Eigenschaften → unbeweglich, nicht unter dem Einfluss der Schwerkraft → keine RigidBody-Komponente → z.B. Plattformboden und -wände.
  - Objekte mit **«physikalischem Collider»**: Objekte mit vollen physikalischen Eigenschaften (Schwerkraft, Einfluss anderer Kräfte), z.B. für Player, Geschosse, sonstige Gegenstände. Benötigen eine **RigidBody**-Komponente!
  - Objekte mit **«kinematischem Collider»**: Objekte mit physikalischen Eigenschaften, die sich bewegen können, allerdings selber durch Kollisionen nicht beeinflusst werden, sondern lediglich Objekte, mit denen sie kollidieren, beeinflussen, z.B. bewegliche Plattformen oder Wände → benötigt eine **RigidBody**-Komponente, aber zusätzlich den Check bei **Is Kinematic**.
  - Objekte mit **«statischem Trigger-Collider»**: Dies sind im Prinzip durchlässige Objekte, die aber bei einer Kollision (Durchquerung durch andere Objekte) einen Trigger auslösen, der in der Spiel-Logik Konsequenzen auslöst, z.B. wenn der Spieler einen Zielort erreicht, oder wenn bei der Durchquerung einer Türe neue Gegner gespawnt werden sollen, etc. → benötigt keinen RigidBody, aber dafür beim Collider den Check bei **Is Trigger**.
  - Objekte mit **«kinematischem Trigger-Collider»**: Dies sind Objekte, die sich bewegen, durch andere Objekte hindurch gehen und dabei in der Game-Logik Ereignisse auslösen, z.B. für Geschoss-Kugeln → wir benötigen einen **RigidBody** mit **Is Kinematic** und gesetztem **Is Trigger**-Check im Collider.
- Sich bewegende Objekte können also voll dynamisch sein (phys. Collider), oder rein kinematisch oder kinematisch triggernd.

- Unsere Wände etc. haben einen statischen Collider.
- Der Player ist physikalisch (= dynamisch)
- Die Enemies sind ki