Einsendeaufgabe

|  |  |  |  |
| --- | --- | --- | --- |
| Rigid Bodies – Grundlagen | |  | Code: |
| UTY14C-XX1-N01 |
|  | | | |
| Name: | Vorname: |  | **Fernlehrer/in:** |
|  |  |  |
| Postleitzahl und Ort: | Straße: | **Datum:** |
|  |  |  |
| Studien- bzw. Vertrags-Nr.: | Lehrgangs-Nr.: | **Note:** |
|  |  |  |
| Bitte reichen Sie Ihre Lösungen über die Online-Lernplattform ein oder schicken Sie uns  diese per Post. Geben Sie bitte immer den Code zum Studienheft an (siehe oben rechts). | | | **Unterschrift Fernlehrer/in:** |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Datum: | 27.11.2023 |  | UTY14C |

Bei den folgenden zwei Multiple-Choice-Aufgaben ist jeweils nur eine Antwort richtig.

1. Aufgabe:

Was ist ein Wheel Collider?

a) [!!br0ken!!]  Ein kugelförmiger Collider.

b) [X]  Ein Collider speziell für Fahrzeugräder.

c) [!!br0ken!!]  Eine Rigidbody-Komponente, die automatisch Fahrzeugräder erzeugen kann.

d) [!!br0ken!!]  Ein Game-Objekt speziell für Fahrzeugräder.

e) [!!br0ken!!]  Ein Skript speziell für Fahrzeugräder.

2. Aufgabe:

Sie wollen, dass eine Rigidbody-Komponente nicht durch die Physiksimulation verändert werden kann. Welche Eigenschaft müssen Sie aktivieren?

a) [!!br0ken!!]  No Physics

b) [!!br0ken!!]  No Simulation

c) [!!br0ken!!]  No Move

d) [X]  Is Kinematic

e) [!!br0ken!!]  Not Kinematic

3. Aufgabe:

Erweitern Sie die Szene **szene06** aus dem Projekt **uty14c\_demos** zu einem einfachen Billardspiel. Ändern Sie dazu das Spielfeld so, dass es ungefähr einem Billardtisch entspricht. Wie groß Sie den Tisch dabei genau machen, spielt keine Rolle. Er sollte nur länger als breit sein. Ergänzen Sie an den Rändern Barrieren mit einem Material, die die Kugeln zurückprallen lassen. Damit die Kugeln nicht endlos weiterrollen, verwenden Sie für die Spielfläche und die Kugeln ein Material mit einer hohen Friktion (zum Beispiel das Material **MaxFriction** aus den Standard Assets). Setzen Sie außerdem die Masse der Kugeln hoch. Für die Kugel des Spielers können Sie zum Beispiel den Wert 1000 benutzen und für die anderen Kugeln den Wert 100.

Sorgen Sie dafür, dass die Kugel des Spielers eindeutig von den anderen Kugeln zu unterscheiden ist – zum Beispiel durch die Farbe.

Setzen Sie in die Ecken des Spielfeldes sowie am linken und rechten Rand in der Mitte Löcher, in denen die Kugeln verschwinden. Dazu können Sie flache Zylinder benutzen. Wenn die Kugel des Spielers in solch ein Loch fällt, soll sie wieder auf den Tisch gelegt werden. Die anderen Kugeln sollen aus dem Spiel genommen werden.

Für die Steuerung dürfen Sie in dem Spiel nicht mehr den Ball nach rechts und links bewegen. Sie können hier zum Beispiel ein Game-Objekt als Zielpunkt verwenden und den Stoß in Richtung dieses Objekts wirken lassen. Damit das Zielobjekt ständig zu sehen ist, sollten Sie eine Kamera in der Vogelperspektive auf das Spielfeld blicken lassen.

Die Kraft des Stoßes soll über das Drücken der Leertaste bestimmt werden. Je länger der Spieler, die Taste drückt und festhält, desto stärker soll der Stoß sein. Begrenzen Sie die Stärke aber nach oben.

Der Spieler soll einen neuen Stoß durchführen können, wenn seine Kugel nahezu still steht.

**Hinweis:**

Beschreiben Sie für die Aufgabe die Schritte, die Sie durchführen und reichen Sie das komplette Projekt mit der Lösung ein.

Packen Sie das Projekt mit einem entsprechenden Programm und beachten Sie die maximale Uploadgröße Ihrer Online-Lernplattform. Prüfen Sie, ob Sie Dinge, die Sie nicht benötigen, entfernen können – zum Beispiel Controller, die Sie nicht verwenden. Die Ordner dieser Controller können Sie löschen.

Ordner, von denen Sie nicht genau wissen, was sie enthalten, sollten Sie nicht löschen. Legen Sie vor jedem Löschen immer eine Sicherungskopie des Projektordners an.  
  
Ich habe einen einfachen Billiard Tisch in Wings3D gebaut und in Unity importiert. Ich nahm das Billiard Spiel der Demos als Vorlage und erweiterte es um die geforderten Features. Ich habe eine Hilfe Funktion auf die Taste „F1“ gelegt die die Steuerung noch einmal kurz erklärt. Hier die Tastatur Steuerung:  
· „A“ und „D“ um zu drehen  
· „Q“ und „E“ zum hinein und herauszoomen  
· „W“ und „S“ steuern die Höhe der Kamera  
· „Leertaste“ gedrückt halten um Energie für den Stoß aufzubauen  
· „R“ setzt die weiße Kugel an den Kopfpunkt zurück  
· „V“ zeigt eine 90° Aufsicht also Vogelperspektive des Tisches   
· „Escape“ Taste beendet das Spiel  
  
Ausserdem habe ich eine Maussteuerung implementiert. Die Maussteuerung ist grob, die Tastatursteuerung fein. Beides läßt sich gleichzeitig benutzen. Energie sammeln mit der Primären (meistens linken) Maustaste. Zoomen mit Mausrad und drehen sowie Höhe über XY-Maus Achsen. Die Eingabe wird mit Lerp geglättet. Man kann das Spiel ausschließlich mit Maus oder Tastatur oder beidem zusammen (empfohlen) spielen.  
  
Ich generiere den Stoß über die Sichtlinie Kamera → weiße Kugel. Die Kamera ist auf die weiße Kugel fixiert. Ich habe zusätzlich einen Energiebalken und eine numerische Darstellung der Stoßenergie hinzugefügt damit man die Energie besser einschätzen kann. Im oberen Bildschirmbereich gibt es einen Zähler für die verbleibenden Kugeln. Es ist möglich über Bande zu spielen, Eintrittswinkel ist gleich Austrittswinkel. Es wird angezeigt unter dem Energie Display, ob ein Stoß möglich ist. Falls die X und /oder Z Geschwindigkeit der weißen Kugel über .1f bzw unter -.1f liegt wird der Stoß blockiert.  
  
Es befinden sich weitere Erläuterungen im Quelltext.   
  
Ich habe diesmal die Szene mit allen enthaltenen Abhängigkeiten exportiert um Inkompatibilitäten zu vermeiden. Bitte legen Sie ein neues Projekt in Unity 2018 an und importieren sie das Paket. Es ist auch möglich das Paket in Unity 2022 zu importieren, allerdings funktioniert das Input System dort etwas hackeliger. Wenn man das Projekt rendert, sollte auch alles in Ordnung sein aber ich empfehle Unity 2018.

4. Aufgabe:

Erstellen Sie über Wheel Collider eine einfache Panzersteuerung, die die Steuerung über Ketten nachbildet. Bei der Vorwärtsbewegung drehen sich beide Ketten nach vorne und bei der Rückwärtsbewegung nach hinten. Beim Lenken soll sich jeweils abhängig von der Richtung eine Kette nach vorne drehen und die andere nach hinten. Das Modell muss sich auf der Stelle drehen können.

Die Ketten für den Panzer müssen Sie nicht nachbilden. Es reichen jeweils zwei Wheel Collider vorne und hinten am Game-Objekt, die Sie entsprechend steuern. Als Vorlage können Sie das Projekt **uty14c\_simplecar** mit dem einfachen Automodell verwenden.

Beschreiben Sie, wie Sie für die Lösung der Aufgabe vorgehen. Reichen Sie neben der Beschreibung auch alle Skripte ein, die Sie erstellen.

Ic hhabe mit Wings3D ein Fahrzeug gebaut und die WheelCollider vom SimpleCar angebracht. Ich habe das Fahrzeug „Wanze“ genannt. Die Wanze hat vier LenkungsTypen zur Auswahl über die Zahlen Tasten „1“, „2“, „3“ und „4“.   
  
1. Standard Vorderrad Lenkung  
2. Gleichlauf Lenkung wie in einem Monstertruck  
3. Kettentrieb Lenkung (ziemlich ineffektiv ohne Ketten) ← diese Aufgabe  
4. Hinterrad Lenkung  
  
Die Bremse liegt auf der „Leertaste“. Mit „L“ läßt sich das große Licht einschalten. Beenden über „Escape“. Hilfe bei drücken von „F1“  
  
Ich benutze einfach den activeTorque mit für die Hinterräder und negiere sie auf einer Seite für den Kettentrieb.  
  
Die Antenne ist von Peter Leary und aus dem Unity Forum. Links befinden sich in der Lizenzen.txt Datei.  
  
Die Schatten werden nicht korrekt gerendert Wenn mehrere Spots leuchten und das Wasser Standard Asset produziert Fehlermeldungen die ich nicht verstehe. Weitere Anmerkungen befinden sich im Quelltext der Fahrzeuge.  
  
Die kompletten Projektdaten habe ich auch nochmal in die Cloud hochgeladen (siehe Readme.txt).

Kommentar:

(nur vom/von Fernlehrer/in auszufüllen)

!!br0ken!!