#11.2 Gummiband

Als Übung wollen wir ein Gummiband simulieren.

Um das Gummiband zu simulieren, gehen wir von einem Haken und einem Ball aus, der durch ein Gummiband mit dem Haken verbunden ist.

Wir erstellen also ein CircleShape und RectangleShape im Mainscreen.  
// Deklariere hier Objekte oder Klassenvariablen!

CircleShape Ball;

RectangleShape Holder;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen

// Hier Startwerte setzen!

public override void setup()

{

Ball = new CircleShape();

Ball.Radius = 20;

Ball.Origin = new Vector2f(20, 20);

Ball.FillColor = Color.Red;

Ball.Position = new Vector2f(400, 400);

Holder = new RectangleShape();

Holder.Size = new Vector2f(10,10);

Holder.Origin = new Vector2f(5, 5);

Holder.FillColor = Color.Green;

Holder.Position = new Vector2f(400, 300);

}

// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen

public override void loop()

{

draw(Ball);

draw(Holder);

}

Da der Ball sich bewegen soll, erstellen wir einen Geschwindigkeitsvektor für ihn.

// Deklariere hier Objekte oder Klassenvariablen!

CircleShape Ball;

RectangleShape Holder;

Vector2f speed;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen

// Hier Startwerte setzen!

public override void setup()

{

speed = new Vector2f(0,0);

[...]  
 }

Da speed ein Geschwindigkeitsvektor ist, müssen wir jeden Frame die Position von Ball um diesen Vektor ändern.

// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen

public override void loop()

{

Ball.Position += speed;

draw(Ball);

draw(Holder);

}

Noch bewegt sich der Ball nicht, weil keine Kräfte auf ihn wirken. Das ändern wir indem wir einen Vektor gravity erstellen, der die Gravitation darstellt. Diesen stellen wir auf (0|0,2).

Vector2f gravity;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen

// Hier Startwerte setzen!

public override void setup()

{

speed = new Vector2f(0,0);

gravity = new Vector2f(0,0.2f);  
 [...]  
 }

Diese Kraft müssen wir jetzt noch auf unseren Geschwindigkeitsvektor addieren.

// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen

public override void loop()

{

speed += gravity;

Ball.Position += speed;

draw(Ball);

draw(Holder);

}

Jetzt sollte der Ball runterfallen. Als nächstes wollen wir das Gummiband einbauen. Dazu brauchen wir physikalische Formeln, die das Verhalten eines Gummibands beschreiben.  
Als vereinfachung gehen wir davon aus, dass ein Gummiband eine gedämpfte harmonische Schwingung erzeugt.

Für Harmonische Schwingungen gilt, dass die Rückstellkraft proportional zur Auslenkung ist. Dass heißt, ums länger wir unser Gummiband ziehen, desto Stärker zieht es sich zusammen. Wir berechnen also die Länge des Gummibands und können daraus eine Kraft herleiten, die auf den Ball wirkt.

Hierfür nutzen wir die Uility.Function Library im Template.

Die Länge des Gummibands kann man errechnen, indem man die Positionen des Haken und des Balls voneinander abzieht und die Länge des resultierenden Vektors berechnet.

// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen

public override void loop()

{

speed += gravity;

Ball.Position += speed;

float strength = (float)utility.Functions.lengthOfVector(Ball.Position - Holder.Position);

draw(Ball);

draw(Holder);

}

Jetzt haben wir unsere Kraft, diese wollen wir jetzt auf unseren Ball anwenden. Aber wir brauchen dafür einen Vektor. Deswegen bilden wir den Einheitsvektor des Differenzvektors, dieser zeigt vom Ball direkt zum Holder und hat die Länge 1. Wenn wir diesen mit der Stärke multiplizieren, ist das unsere Rückstellkraft. Diese versehen wir noch mit einem Faktor, damit die Kraft nicht zu stark ist.

// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen

public override void loop()

{

float strength = (float)utility.Functions.lengthOfVector(Ball.Position - Holder.Position);

Vector2f force = strength \* utility.Functions.getUnitVector(Holder.Position - Ball.Position) \* 0.01f;

speed += force;

speed += gravity;

Ball.Position += speed;

draw(Ball);

draw(Holder);

}

Jetzt schwingt unser Ball, aber noch ist es kein Gummiband. Gummibänder üben nur Kraft aus, wenn man sie länger zieht als ihre gewöhnliche Länge. Deswegen ziehen wir 100 von der strength variable ab, so dass das Gummiband eine Länge von 100 Pixeln hat. Außerdem müssen wir jetzt negative Werte nullen.

public override void loop()

{

float strength = (float)utility.Functions.lengthOfVector(Ball.Position - Holder.Position) - 100;

if (strength < 0) strength = 0;

Vector2f force = strength \* utility.Functions.getUnitVector(Holder.Position - Ball.Position) \* 0.01f;

speed += force;

speed += gravity;

Ball.Position += speed;

draw(Ball);

draw(Holder);

}

Als letztes Simulieren wir noch Reibung, indem wir den Speed Vektor mit einem Faktor multiplizieren, der etwas kleiner als 1.0 ist.

public override void loop()

{

float strength = (float)utility.Functions.lengthOfVector(Ball.Position - Holder.Position) - 100;

if (strength < 0) strength = 0;

Vector2f force = strength \* utility.Functions.getUnitVector(Holder.Position - Ball.Position) \* 0.01f;

speed += force;

speed += gravity;

speed \*= 0.95f;

Ball.Position += speed;

draw(Ball);

draw(Holder);

}

Das Gummiband ist jetzt fertig, aber um es wirklich in Aktion zu sehen, kann man mithilfe von Mausklicks den Haken bewegen. Dazu brauchen wir nur folgende If-Klammer:

if (Mouse.IsButtonPressed(Mouse.Button.Left))

{

Holder.Position = new Vector2f(Mouse.GetPosition(game.gameWindow).X, Mouse.GetPosition(game.gameWindow).Y);

}