

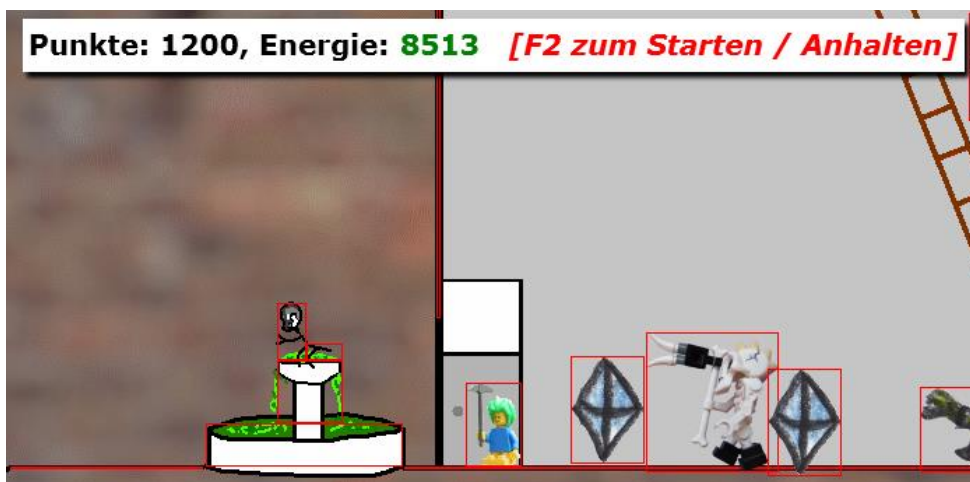
1 Wie tickt das Spiel denn so?

Schaut man sich den Spielverlauf an so scheint sich die Spielfigur einfach durch die Gegend zu bewegen und sammelt dabei Schätze ein, muss sich gegen Gegner wehren und einen bestimmten Zielpunkt erreichen. Um zu verstehen, wie man eine solche Spielrunde gestaltet ist es wichtig, ein wenig die Hintergründe und die Abläufe des Spiels zu verstehen. Nehmen wir einmal eine ganz normale Spielsituation.

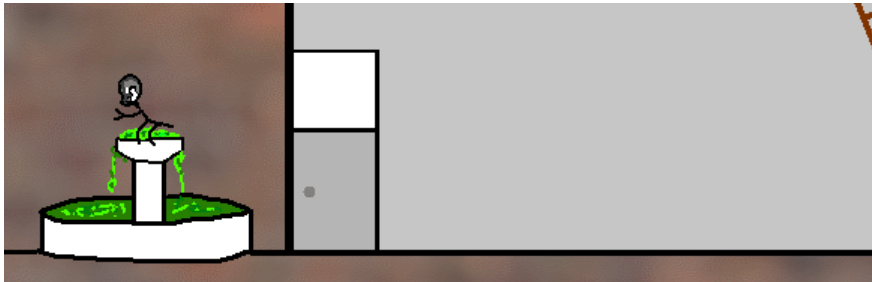


Die Spielfigur hat schon ein erstes Hindernis (den Springbrunnen) übersprungen und ist ins Innere eines Gebäudes gelangt. Hier gilt es nun durch das Einsammeln der Kristalle Punkte zu machen, zu verhindern, dass die Gegner Energie entziehen und dann über die Leiter ins nächste Stockwerk zu gelangen.

Für das Spiel selbst ist es nicht von Interesse, wie der Hintergrund des Spielfelds, die Spielfigur, die Schätze oder Gegner aussehen: es kennt nur rechteckige Hindernisse, die durch bei der Berührung durch die Spielfigur bestimmte Aktionen auslösen können – oder einfach im Weg stehen wie unser Springbrunnen. Auch die Spielfigur selbst ist für das Spiel nur ein Rechteck. In einem besonderen Testmodus ist es möglich, sich diese Sichtweise des Spiels durch rote Rahmen um die Hindernisse darstellen zu lassen.



Im Vergleich mit dem Hintergrundbild kann man auch sehr schön sehen, welche der Hindernisse fest auf diesem angebracht sind und welche auch im Verlaufe des Spiels verschwinden könnten, wie zum Beispiel Schätze und Gegner.



Diese kleine Anleitung zeigt in einfachen Schritten, wie die einzelnen Spielfelder eines Spiels aufgebaut sind und wie bestimmte Reaktionen der Hindernisse auf die Spielfigur erreicht werden können.

2 Die wichtigsten Spielelemente Schritt für Schritt

Anfangen von einem leeren Hintergrund lernen wir hier nun die wichtigsten Bausteine zum Erstellen eines Spiels und den darin enthaltenen Spielfeldern kennen. Im Anschluss gibt es dann weitere Detailinformationen für Fortgeschrittene, die vielleicht auch einmal ihr eigenes Spiel gestalten wollen.

Für ein Spiel werden neben dem ausführbaren Programmcode *TheGame.xap* in Unterverzeichnis *ClientBin* und einer HTML Startdatei drei Arten von Dateien benötigt:

- Die Beschreibung des Spielablaufs erfolgt in Textdateien mit einem besonderen XML Format und der Dateiendung XAML. Die besondere Form ergibt sich aus der verwendeten Programmierungsumgebung *Microsoft Silverlight (Version 5)*.
- Bilder für die Spielfigur, Schätze, Gegner, Spielfelder und so weiter. Es hat sich bisher als praktisch erwiesen, dass allgemeine Bildformat PNG zu verwenden.
- Optional Tondateien im MP3 Format für kurze Melodien, die etwa beim Einsammeln von Schätzen abgespielt werden können.

Auf die Erstellung von Bild- und Tondateien wird hier nicht weiter eingegangen, diese Einführung konzentriert sich auf die Beschreibung des Spielablaufs in den XAML Dateien.

2.1 Wie sehe ich denn aus?

Am Anfang steht immer die Auswahl einer Spielfigur. Eine Spielfigur ist immer mit mindestens einer Bilddatei verbunden – als Größe wird eine Breite von 40 Punkten und eine Höhe von 60 Punkten empfohlen.

```
<Figur xmlns="clr-namespace:JMS.JnRV2.Ablage.V1;assembly=GamePersistence"
    SprungStaerke="100"
    SpruengeNacheinander="2"
    BilderProSekunde="10"
    MaximaleGeschwindigkeit="5"
    Name="Lemming"
>
<Figur.BilderRuhend>
    <Bild Quelle="bilder/ruhend.png" />
```

```

</Figur.BilderRuhend>

<Figur.BilderNachLinks>
  <Bild Quelle="bilder/ruhend.png" />
</Figur.BilderNachLinks>

<Figur.BilderNachRechts>
  <Bild Quelle="bilder/ruhend.png" />
</Figur.BilderNachRechts>

<Figur.BilderImSprung>
  <Bild Quelle="bilder/ruhend.png" />
</Figur.BilderImSprung>
</Figur>

```

Jede Spielfigur hat immer einen eindeutigen Namen, in diesem Beispiel ist das unser *Lemming*. Im Spiel wird unterschieden, ob die Spielfigur gerade steht, springt oder sich horizontal bewegt. Für jede dieser vier Bewegungsarten kann ein unterschiedliches Bild verwendet werden. Unser kleiner Lemming ist aber etwas steif geraten und sieht immer gleich aus – verwendet wird die Bilddatei *ruhend.png* im Unterverzeichnis *bilder* des *ClientBin* Unterverzeichnisses.



Die Beschreibungen der Spielfiguren werden unter dem Namen *SpielerXX.xaml* im Unterverzeichnis *Spielfiguren* abgelegt. Berücksichtigt werden immer alle Dateien beginnend bei *Spieler01.xaml* mit fortlaufender Nummerierung – bei einer Lücke werden folgende Dateien ignoriert. Es kann daher im Moment maximal 99 unterschiedliche Spielfiguren geben.

2.2 Ein Fenster ins Spiel

Im Unterverzeichnis *Spiele* sind die Spieldateien untergebracht. Eine solche Datei enthält einige wenige Eckdaten für die Umgebung, in der das Spiel ausgeführt werden soll. Für diese kleine Anleitung wird etwa die Datei *DemoUmgebung.xaml* verwendet:

```

<Spiel xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

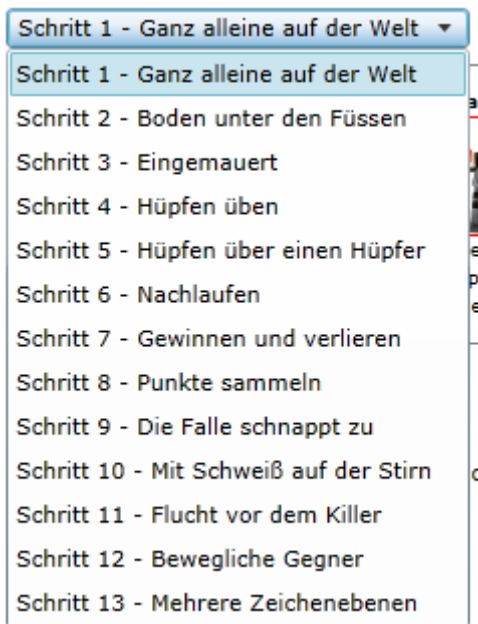
  SpielfeldKategorien="Demo"
  SichtfensterBreite="800"
  SichtfensterHoehe="600"

/>

```

Die Einstellungen für die Breite und Höhe legen den sichtbaren Bereich des gesamten Spielfelds fest. Die Spielfigur befindet sich immer genau in der Mitte des Sichtfensters und jede Bewegung der Spielfigur führt letztlich dazu, dass das Fenster über das Spielfeld bewegt wird. Wir verwenden hier ein Sichtfenster mit einer Breite von 800 Punkten und einer Höhe von 600 Punkten.

Die Kategorien sind eine durch Komma getrennte Liste von Bezeichnungen von Spielfeldern. Wird das Spiel der Anleitung gestartet, so werden nur Spielfelder zur Auswahl angeboten, die zu der Kategorie *Demo* gehören.



Wenn man diese Einführung Live mitverfolgen mit, einfach das Spiel Online aufrufen:

<http://jumpnrun.nicolas-manns.de/demo.html>

2.3 Runde für Runde

Im Unterverzeichnis Spielfelder befinden sich die Beschreibungen der verfügbaren Spielfelder. Ein Spielfeld muss immer den Namen *LevelXX.xml* haben, wobei *XX* eine Zahl zwischen 01 und 99 ist. Im Augenblick kann es daher auch nicht mehr als 99 unterschiedliche Spielfelder geben. Berücksichtigt werden beginnend mit der Zahl 01 alle Spielfelder mit fortlaufenden Nummern: eine Lücke in der Nummerierung führt dazu, dass alle folgenden Spielfelder nicht mehr berücksichtigt werden.

```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"
  Kategorie="Demo"
  Kennung="a051452f-20b7-4bc2-b54d-e69d16801b77"
  Beschreibung="Schritt 1 - Ganz alleine auf der Welt"
  Bild="bilder/demo/back01.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 1000"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"
>
</Spielfeld>
```

Auch ein leeres Spielfeld wie die erste Runde zu dieser Anleitung hat schon eine ganze Menge unterschiedlicher Einstellungen, die zum Teil erst in den folgenden Schritten beschrieben werden.

Für das Spiel am wichtigsten ist das Hintergrundbild. Im Beispiel wird eine Datei mit dem Namen *back01.png* im Unterverzeichnis *demo* des Unterverzeichnisses *bilder* verwendet. Das hier verwendete Bild ist allerdings besonders langweilig:



Dieser Hintergrund ist 4.000 Punkte breit und 4.000 Punkte hoch und besitzt im inneren ein einfaches Rautenmuster. Die Randbreite wurde der Größe des Sichtfensters zum Spiel angepasst. So sind etwa die seitlichen Ränder genau 400 Punkte breit. Damit lässt sich sehr gut die Bewegung der Spielfigur erklären.

In der Beschreibung des Spielfelds ist festgelegt, wo die Spielfigur am Anfang des Spiels stehen soll. Im Beispiel ist das ein Abstand von 1.000 Punkte vom linken und 1.000 Punkte vom unteren Rand.



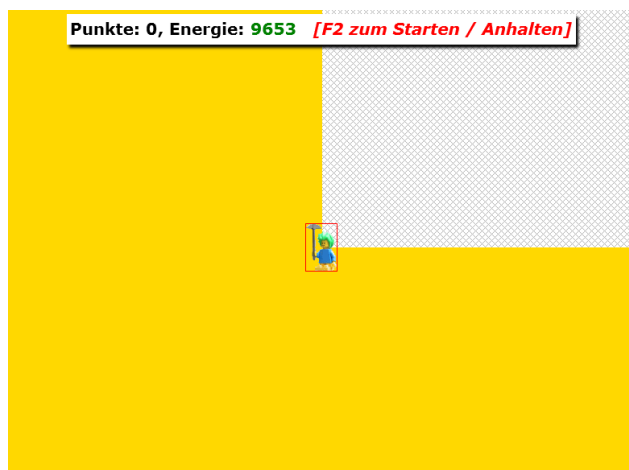
Wird das Spiel gestartet, so fällt die Spielfigur so lange nach unten, bis sie auf ein Hindernis stößt. Im ersten Spielfeld gibt es keinerlei Hindernisse, daher endet der Fall, wenn das Sichtfenster mit dem unteren Rand auf den unteren Rand des Hintergrundbildes trifft. Der Mittelpunkt der Spielfigur befindet sich dabei immer im Zentrum des Sichtfensters, daher sieht es bei dem gewählten Hintergrund so aus, als wären wir halb im Boden versunken – der untere Rand ist genau halb so hoch wie das Sichtfenster.



Genauso sieht es dann aus, wenn wir mal eben schnell ganz an den linken Rand des Spielfelds laufen: auch hier stecken wir nun zur Hälfte in der Wand.

Und viel mehr gibt es in diesem ersten Spielfeld auch schon nicht zu sehen. Vielleicht aber schon einige Informationen zu den Einstellungen eines Spielfelds. Ein Spielfeld hat immer eine kurze Beschreibung, die auch in der Auswahlliste beim Starten des Spiels angezeigt wird – für das erste Spielfeld wäre das *Schritt 1 - Ganz alleine auf der Welt*. Als Art des Spielfelds wurde *Demo* angegeben, da unser Spiel nur Spielfelder dieser Art zur Auswahl anbietet.

Zusätzlich muss jedes Spielfeld einen eindeutigen Namen erhalten, über den sich das Spiel merkt, welches Spielfeld beim letzten Versuch ausgewählt wurde. Für diesen eindeutigen Namen wird eine besondere Art von Kennzahl verwendet (http://de.wikipedia.org/wiki/Globally_Unique_Identifier, <http://guid.us>) – unser erstes Spielfeld hat den wenig anschaulichen eindeutigen Namen *a051452f-20b7-4bc2-b54d-e69d16801b77*.



2.4 Auf sicherem Boden

Das zweite Spielfeld wurde gegenüber dem ersten ein erstes Hindernis eingefügt, nämlich ein Boden.

```

<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

    Kategorie="Demo"
    Kennung="09a52384-6d9a-4d75-ab99-e22a8739d32c"
    Beschreibung="Schritt 2 - Boden unter den Füßen"
    Bild="bilder/demo/back02.png"
    BildVerloren="bilder/gameover.png"
    BildGewonnen="bilder/gamewon.png"
    InitialePosition="1000, 400"
    MaximaleSprungStaerke="100"
    MaximaleGeschwindigkeit="5"

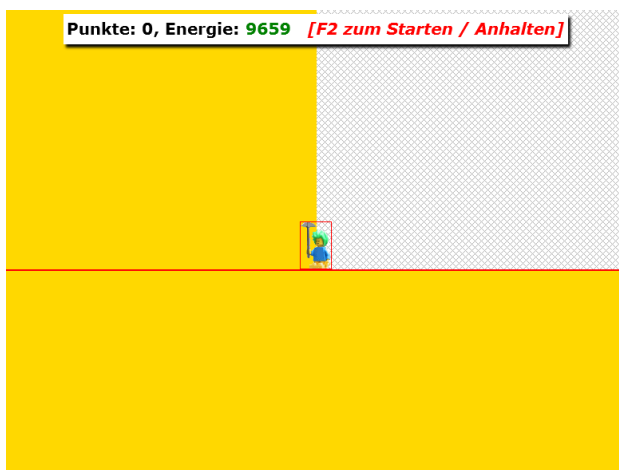
>

    <Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />

</Spielfeld>

```

Hindernisse sind grundsätzlich rechteckig und werden durch ihren Abstand vom linken und unteren Rand des Hintergrundbildes positioniert. Das Hindernis auf diesem Spielfeld hat kein eigenes Bild, sondern ist einfach nur ein Kasten mit einer Höhe von einem Punkt, der sich über das gesamte Spielfeld erstreckt. Der Abstand vom unteren Rand ist so gewählt, dass das Hindernis genau auf der Grenze des gezeichneten Rahmens liegt. Lassen wir die Spielfigur nun nach unten fallen und laufen dann ganz nach Links, so stehen wir scheinbar auf festem Boden.



2.5 Mit dem Kopf gegen die Wand

Im nächsten Spielfeld beschränken wir den Auslauf unserer Spielfigur auch nach links und rechts, in dem wir einige Hindernisse quasi als Mauern ergänzen.

```

<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

    Kategorie="Demo"
    Kennung="90d6a3a4-7417-40a5-8de2-8520cf9a2f9a"
    Beschreibung="Schritt 3 - Eingemauert"
    Bild="bilder/demo/back03.png"
    BildVerloren="bilder/gameover.png"
    BildGewonnen="bilder/gamewon.png"
    InitialePosition="1000, 400"
    MaximaleSprungStaerke="100"
    MaximaleGeschwindigkeit="5"

>

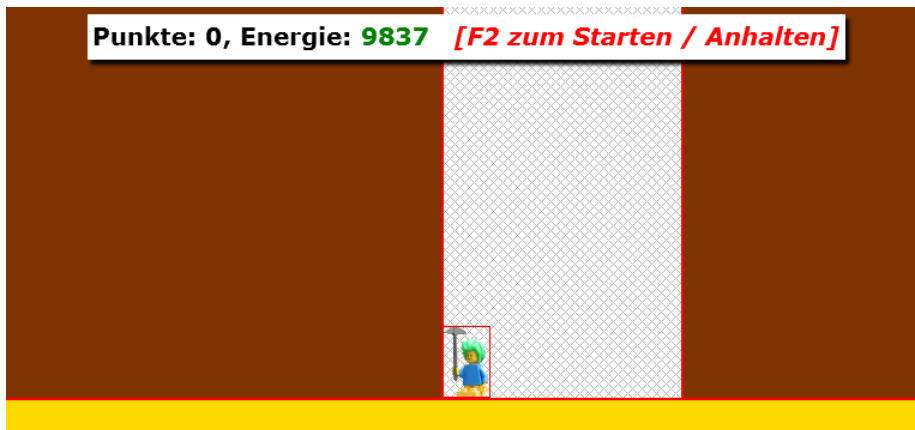
    <Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />

    <Element HorizontalePosition="900" VertikalePosition="300" Breite="1" Hoehe="500" />
    <Element HorizontalePosition="1100" VertikalePosition="300" Breite="1" Hoehe="500" />

</Spielfeld>

```

Damit der Spieler seine Grenzen auch sehen kann verwenden wir nun ein verändertes Hintergrundbild. Wollen wir mal hoffen, dass unsere Spielfigur keine Angst vor engen Räumen hat!

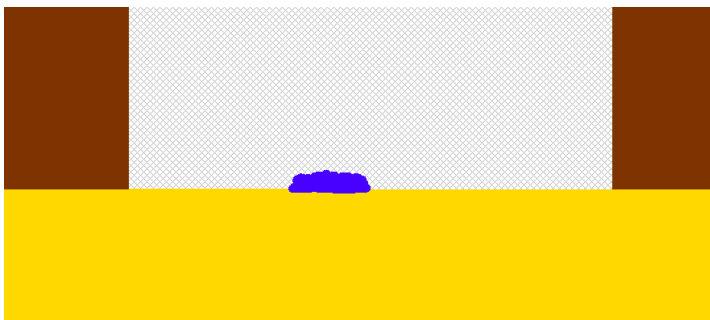


2.6 Hoch hinaus

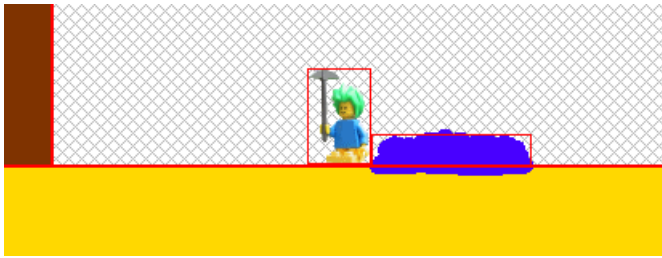
Bereits mit dieser einfachen Art Hindernisse zu beschreiben kann man schon erste Spielelemente im Spielfeld unterbringen. Die Definition des Spielfelds scheint sich kaum verändert zu haben.

```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"
  Kategorie="Demo"
  Kennung="a57bb4ae-5a21-4d11-80b0-73bbf80c37a1"
  Beschreibung="Schritt 4 - Hüpfen üben"
  Bild="bilder/demo/back04.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 400"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"
>
  <Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />
  <Element HorizontalePosition="900" VertikalePosition="300" Breite="1" Hoehe="500" />
  <Element HorizontalePosition="1100" VertikalePosition="300" Breite="100" Hoehe="20" />
  <Element HorizontalePosition="1500" VertikalePosition="300" Breite="1" Hoehe="500" />
</Spielfeld>
```

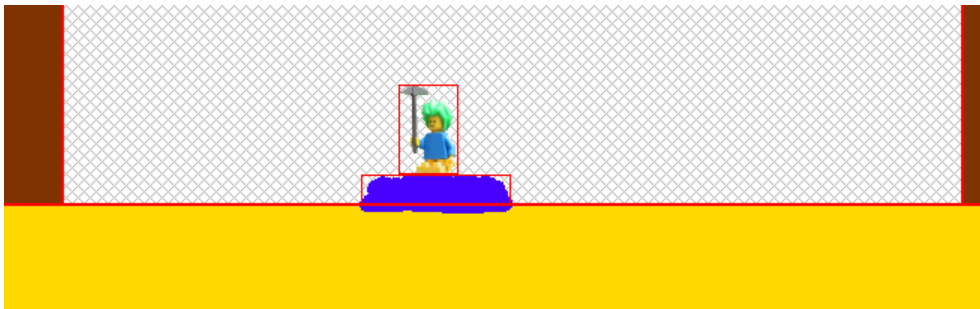
Tatsächlich wird hier aber ein veränderter Hintergrund eingesetzt, auf dem ein Bild für das Hindernis eingezeichnet ist.



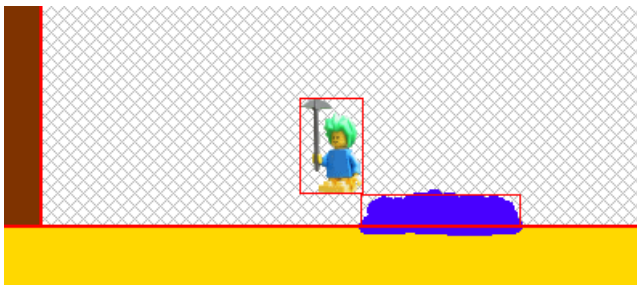
Lassen wir die Spielfigur fallen und dann nach rechts laufen, so bleibt sie erst einmal am Hindernis hängen.



Es ist natürlich ein Leichtes mit einem kleinen Hüpfen diese Hürde zu überwinden.



Man sieht hier im Testmodus durch die eingezeichneten roten Rahmen um die rechteckigen Hindernisse sehr schön, dass sich bei beliebig geformten möglicherweise eigenartige Effekte im Spiel ergeben können. Bewegt man die Spielfigur an eine der Ecken des Hindernisses, so scheint sie in der Luft zu schweben.



Sollten sich solche Effekte als zu störend erweisen, so kann man versuchen, die Konturen des Hindernisses durch viele kleinere Rechtecke nachzubilden. In den meisten Fällen kann man sich aber vermutlich eher durch eine etwas andere bildliche Gestaltung der Hindernisse helfen.

2.7 Da hüpf doch was

In den bisherigen Spielfeldern waren Hindernisse fest mit dem Hintergrund verbundene Rechtecke. Es ist aber auch möglich, beliebige Bilder zu Hindernissen zu machen und diese dann auf dem Spielfeld zu platzieren. Mit Hilfe einfacher Regeln ist es dann möglich, diesen Hindernissen auch etwas Leben einzuhauchen, wie es das nächste Spielfeld zeigt.

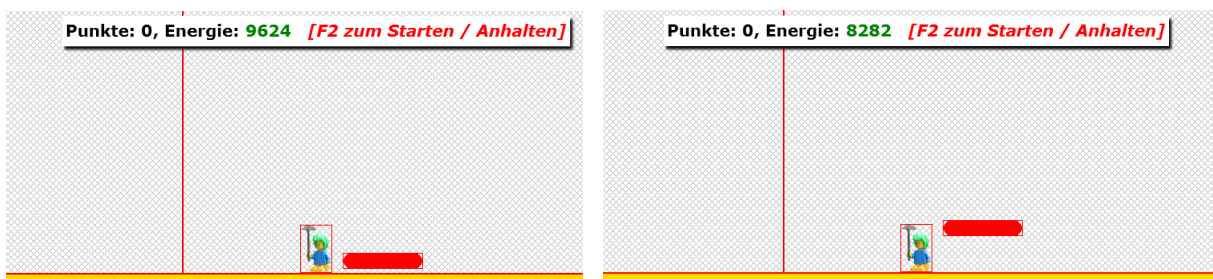
```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"
  Kategorie="Demo"
  Kennung="9a598dae-b9fc-4f6a-a984-30a68a061dd0"
  Beschreibung="Schritt 5 - Hüpfen über einen Hüpfen"
  Bild="bilder/demo/back05.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 400"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"
```

```

>
<Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />
<Element HorizontalePosition="900" VertikalePosition="300" Breite="1" Hoehe="500" />
<ElementMitBildSequenz HorizontalePosition="1100" VertikalePosition="301">
  <ElementMitBildSequenz.Bilder>
    <EinzelBild Quelle="bilder/demo/huepfer1.png" />
  </ElementMitBildSequenz.Bilder>
  <ElementMitBildSequenz.Animation>
    <VertikaleBewegung Sprunghoehe="50" Sprungzeit="0.5" />
  </ElementMitBildSequenz.Animation>
</ElementMitBildSequenz>
<Element HorizontalePosition="1500" VertikalePosition="300" Breite="1" Hoehe="500" />
</Spielfeld>

```

Das bewegliche Hindernis ist mit dem zugegeben wenig beeindruckenden Bildchen huepfer1.png verbunden – hier ist der Fantasie natürlich freier Lauf gelassen. Ist ein Hindernis mit einem Bild verbunden, so muss die Größe des Hindernisses nicht mehr angegeben werden – diese ist dann identisch zur Größe des Bildes. Zusätzlich kann dem Hindernis eine Animation zugeordnet werden. Das bewegliche Hindernis auf unserem Spielfeld bewegt sich um 50 Punkte nach oben, fällt dann wie auch die Spielfigur wieder nach unten und wiederholt diese Bewegung dann unbegrenzt.



Ansonsten verhält sich ein bewegliches Hindernis genau wie ein festes Hindernis, nur dass es je nach verwendeter Animation seine Position auf dem Spielfeld verändert. Aber auch Hindernisse mit Bild aber ohne Animation können sehr nützlich sein, selbst wenn sie sich nicht bewegen: das Bild kann so nicht nur mehrfach an verschiedenen Positionen eines Spielfelds, sondern auch in verschiedenen Spielfeldern genutzt werden, ohne dass es auf den Hintergrund gezeichnet werden muss. So ist es auch sehr einfach, Hindernisse zurecht zu rücken, ohne das Hintergrundbild mühsam anpassen zu müssen.

2.8 Hilfe, ich werde verfolgt

Ein Hindernis mit Bild kann nicht einfach nur eine Bereicherung des Hintergrunds eines Spielfelds sein, sondern auch eine Schatz oder einen Gegner repräsentieren. Eine der möglichen Animationen eines beweglichen Hindernisses ist die Fähigkeit, sich auf die Spielfigur zu oder von der Spielfigur weg zu bewegen. Wie das geht zeigt das nächste Spielfeld.

```

<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"
  Kategorie="Demo"
  Kennung="2c27d8cc-eb22-4118-9809-97e73c54bb05"
  Beschreibung="Schritt 6 - Nachlaufen"
  Bild="bilder/demo/back06.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 400"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"

```

```

>
<Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />

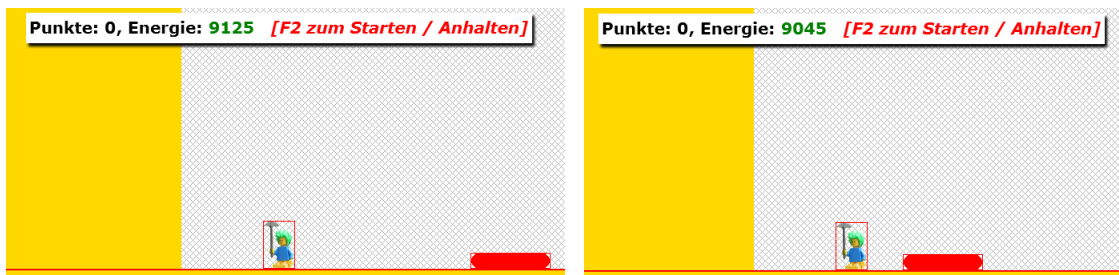
<ElementMitBildSequenz HorizontalePosition="1100" VertikalePosition="301">
  <ElementMitBildSequenz.Bilder>
    <EinzelBild Quelle="bilder/demo/huepfer1.png" />
  </ElementMitBildSequenz.Bilder>

  <ElementMitBildSequenz.Animation>
    <BewegungRelativZurSpielfigur Angriff="True" HorizontaleGeschwindigkeit="2" />
  </ElementMitBildSequenz.Animation>
</ElementMitBildSequenz>

</Spielfeld>

```

Aus Faulheit wird hier wieder das einfache Bild aus dem vorherigen Spielfeld erneut verwendet. Nur soll sich dieses nicht einfach nur auf und ab bewegen, sondern vielmehr die Spielfigur attackieren – auf diesem Spielfeld noch ohne böse Folgen.



Einen ganz lustigen Effekt kann man erzielen, wenn man mit der Spielfigur auf das Hindernis hüpft und sich langsam in eine beliebige Richtung weiter bewegt: das Hindernis bleibt unter der Spielfigur und bewegt sich mit ihr mit.



2.9 Schade, schon zu Ende

Alle bisherigen Spielfelder haben einen Fehler gemeinsam: man kann die damit verbundene Spielrunde nicht beenden. Man kann zwar nicht verlieren, aber auch nicht gewinnen. Um dies zu ermöglichen können allen Hindernissen Regeln zugeordnet werden, die bei einem Zusammentreffen mit der Spielfigur ausgeführt werden. Eine dieser Regeln erlaubt das Beenden der Spielrunde, wie das folgende Spielfeld zeigt.

```

<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

  Kategorie="Demo"
  Kennung="6565d4c2-24a4-4cc6-a0f0-3b857de7620d"
  Beschreibung="Schritt 7 - Gewinnen und verlieren"
  Bild="bilder/demo/back07.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 400"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"

```

```

>

<Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />

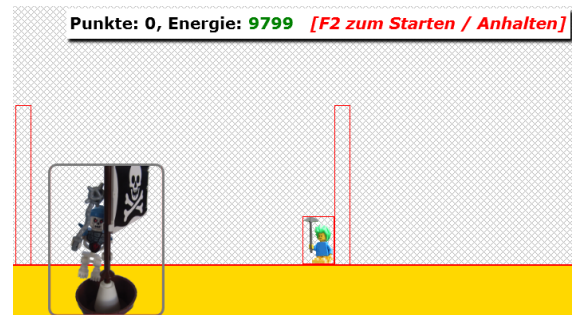
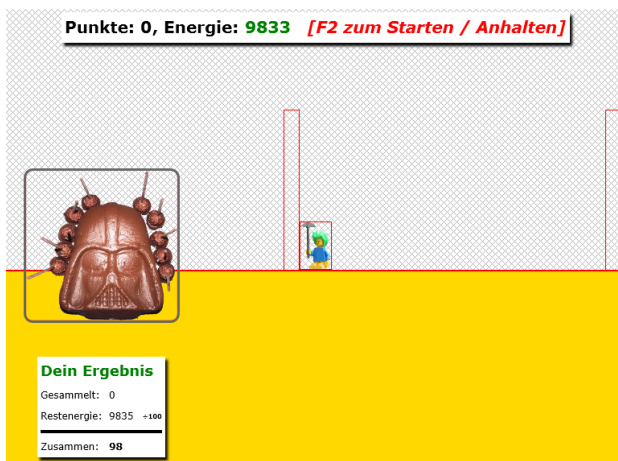
<Element HorizontalePosition="800" VertikalePosition="300" Breite="20" Hoehe="200">
  <Enderegel ArtDerKollision="VomSpielerGetroffen" Gewonnen="True" />
</Element>

<Element HorizontalePosition="1200" VertikalePosition="300" Breite="20" Hoehe="200">
  <Enderegel ArtDerKollision="VomSpielerGetroffen" Gewonnen="False" />
</Element>

</Spielfeld>

```

Wird die linke Mauer von der Spielfigur getroffen, so gilt die Runde als gewonnen. Die rechte Seite hingegen führt zur Niederlage. Natürlich sollte jedes Spielfeld mindestens die Möglichkeit bieten, die zugehörige Spielrunde zu gewinnen. Spätestens dann ist es auch notwendig, zugehörige Abschlussbilder für das Spielfeld festzulegen. Im Beispiel wird *gameover.png* bei einer Niederlage angezeigt und *gamewon.png*, wenn das Spiel gewonnen wurde – beide Bilder werden im Unterverzeichnis *bilder* gesucht.



2.10 Jäger und Sammler

Natürlich wäre es ein wenig zu langweilig, einfach bis zum Sieg oder einer Niederlage durch die Gegen zu laufen. Bewegliche Hindernisse können als Schätze definiert werden, die durch die Spielfigur eingesammelt werden können. Das nächste Spielfeld enthält einige solcher Schätze.

```

<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

  Kategorie="Demo"
  Kennung="f3b7b7ff-beda-47f6-90db-04aefd00d4e1"
  Beschreibung="Schritt 8 - Punkte sammeln"
  Bild="bilder/demo/back08.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 400"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"

>

<Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />

<ElementMitBildSequenz HorizontalePosition="1200" VertikalePosition="350"
  Melodie="melodien/sammeln.mp3">

  <ElementMitBildSequenz.Bilder>
    <EinzelBild Quelle="bilder/demo/sammel.png" />
  </ElementMitBildSequenz.Bilder>

```

```

        <Punkteregel ArtDerKollision="VomSpielerGetroffen" Punkte="100" />
        <Verschwinderegel ArtDerKollision="VomSpielerGetroffen" />
    </ElementMitBildSequenz>

    <ElementMitBildSequenz HorizontalePosition="1600" VertikalePosition="350">
        <ElementMitBildSequenz.Bilder>
            <EinzelBild Quelle="bilder/demo/sammel.png" />
        </ElementMitBildSequenz.Bilder>

        <Punkteregel ArtDerKollision="VomSpielerGetroffen" Punkte="200" />
        <Verschwinderegel ArtDerKollision="VomSpielerGetroffen" />
    </ElementMitBildSequenz>

    <ElementMitBildSequenz HorizontalePosition="2000" VertikalePosition="350"
        Melodie="melodien/sammeln.mp3">

        <ElementMitBildSequenz.Bilder>
            <EinzelBild Quelle="bilder/demo/sammel.png" />
        </ElementMitBildSequenz.Bilder>

        <Punkteregel ArtDerKollision="VomSpielerSeitlichGetroffen" Punkte="-50" />
        <Verschwinderegel ArtDerKollision="VomSpielerSeitlichGetroffen" />
        <Punkteregel ArtDerKollision="VomSpielerGetroffen" Punkte="5000" />
        <Verschwinderegel ArtDerKollision="VomSpielerGetroffen" />
    </ElementMitBildSequenz>

    <ElementMitBildSequenz HorizontalePosition="2400" VertikalePosition="310" >
        <ElementMitBildSequenz.Bilder>
            <EinzelBild Quelle="bilder/demo/sammel.png" />
        </ElementMitBildSequenz.Bilder>

        <Punkteregel ArtDerKollision="VomSpielerGetroffen" Punkte="5000" />
        <Verschwinderegel ArtDerKollision="VomSpielerGetroffen" />
    </ElementMitBildSequenz>

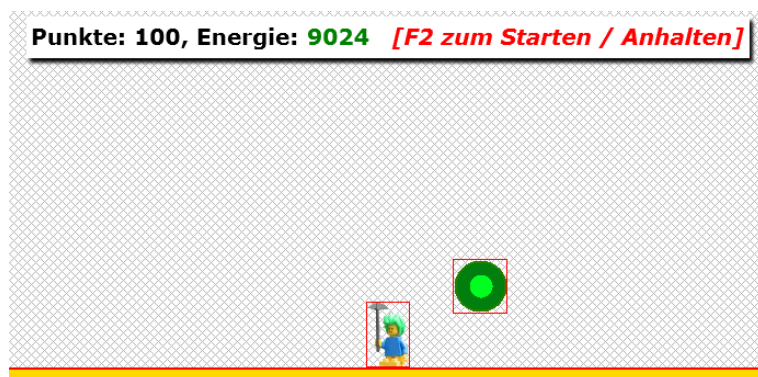
</Spielfeld>

```

Alle Schätze verwenden hier dasselbe Bild *sammel.png*, was wieder recht einfach gehalten ist. Bei einigen Schätzen wird eine kurze Melodie *sammeln.mp3* abgespielt, sobald sie von der Spielfigur berührt werden – die Tondatei befindet sich im Unterverzeichnis *melodien*.

In den einfachen Fällen wird ein bestimmter Punktestand gutgeschrieben, sobald ein Schatz von der Spielfigur getroffen wird. Der Schatz wird dann auch von dem Spielfeld entfernt und kann nicht noch einmal zur Bereicherung verwendet werden.

Die Regeln erlauben es aber auch zu unterscheiden, wie ein bewegliches Hindernis von der Spielfigur getroffen wird. Der dritte Schatz auf dem Spielfeld entzieht 50 Punkte, wenn die Spielfigur einfach nur plump von einer Seite kommend dagegen läuft. Bei einer anderen Art des Kontaktes allerdings wird eine Gutschrift von 5.000 Punkten erstellt – etwa beim Hüpfen auf den Schatz. Mit diesen Optionen lässt sich das Sammeln von Punkten etwas abwechslungsreicher und zum Teil auch anspruchsvoller gestalten.



2.11 Erscheinen und Verschwinden

Mit dem Regelwerk über das Hindernisse auf Kontakte vor allem mit der Spielfigur reagieren können, lassen sich eine Reihe von interessanten Effekten erzeugen. Mit dem folgenden Spielfeld versuchen wir eine allzu leichtsinnige Spielfigur einzufangen.

```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

  Kategorie="Demo"
  Kennung="4b812d25-6cc1-4da7-b20a-ebdd0ade823e"
  Beschreibung="Schritt 9 - Die Falle schnappt zu"
  Bild="bilder/demo/back09.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 400"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"

>

<Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />

<Element HorizontalePosition="1200" VertikalePosition="330" Breite="10" Hoehe="10">
  <Erscheineregel ArtDerKollision="VomSpielerGetroffen" Name="links" />
  <Erscheineregel ArtDerKollision="VomSpielerGetroffen" Name="rechts" />
  <Erscheineregel ArtDerKollision="VomSpielerGetroffen" Name="oben" />
  <Verschwinderegel ArtDerKollision="VomSpielerGetroffen" />
</Element>

<Element HorizontalePosition="1100" VertikalePosition="300" Breite="1" Hoehe="500"
  Name="links" AnfänglichUnsichtbar="True" Melodie="melodien/sammeln.mp3">
  <Energierregel ArtDerKollision="VomSpielerGetroffen" Lebensenergie="-9500"/>
</Element>

<Element HorizontalePosition="1300" VertikalePosition="300" Breite="1" Hoehe="500"
  Name="rechts" AnfänglichUnsichtbar="True" />

<Element HorizontalePosition="1100" VertikalePosition="400" Breite="200" Hoehe="1"
  Name="oben" AnfänglichUnsichtbar="True">
  <Enderegel ArtDerKollision="VomSpielerGetroffen" Gewonnen="True" />
</Element>

</Spielfeld>
```

Scheinbar fängt alles ganz harmlos an. Drei der Hindernisse sind so definiert, dass sie anfänglich nicht sichtbar sind und am Spielverlauf nicht teilnehmen – die Spielfigur kann problemlos die Bereiche durchqueren, die von diesen Hindernissen abgedeckt werden. Ein spezielles Hindernis reagiert auf eine Berührung durch die Spielfigur und macht nicht nur sich selbst unsichtbar sondern auch noch die anderen Hindernisse sichtbar und damit aktiv. Letztlich haben wir die Spielfigur nun eingefangen.



Ganz so gemein ist diese Runde aber nicht: die Spielfigur muss zum Gewinnen nur gegen die Decke hüpfen. Für das Hindernis links wurde eine weitere Regel eingesetzt, die dafür sorgt, dass der Spielfigur bei Kontakt 9.500 Energiepunkte entzogen werden – was auf diesem Spielfeld zu einem ziemlich schnellen Tod führen wird.

2.12 Weg damit

Eine weitere Regel erlaubt es der Spielfigur bewegliche Hindernisse zu verschieben. Da dies mit Aufwand verbunden ist wird die gesamte Bewegung deutlich verlangsamt. Das folgende Spielfeld enthält ein solches verschiebbares Hindernis.

```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

    Kategorie="Demo"
    Kennung="16686183-4199-4a58-9c95-af672c723b3e"
    Beschreibung="Schritt 10 - Mit Schweiß auf der Stirn"
    Bild="bilder/demo/back10.png"
    BildVerloren="bilder/gameover.png"
    BildGewonnen="bilder/gamewon.png"
    InitialePosition="1000, 400"
    MaximaleSprungStaerke="100"
    MaximaleGeschwindigkeit="5"

>

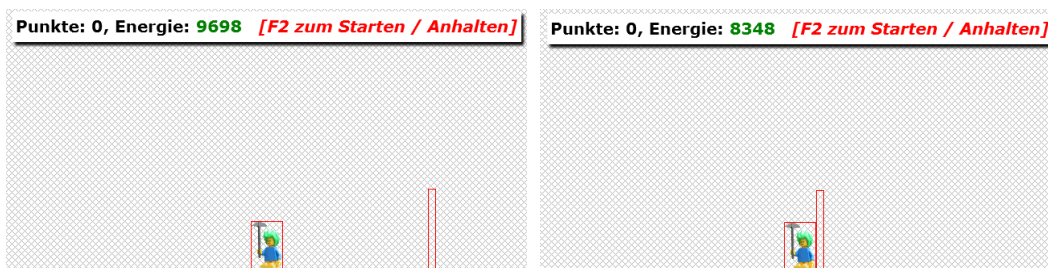
<Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />

<Element HorizontalePosition="1200" VertikalePosition="301" Breite="10" Hoehe="100">
    <Verschieberegul ArtDerKollision="VomSpielerGetroffen" />
</Element>

<Element HorizontalePosition="1300" VertikalePosition="301" Breite="1" Hoehe="1" />

</Spielfeld>
```

Als kleine Gemeinheit ist ein anderes winziges Hindernis so platziert, dass beim Schieben nach rechts sehr schnell nichts mehr geht. Die Spielfigur könnte aber das Hindernis überspringen und dann bis an den linken Bildrand schieben.



2.13 Da fliegt doch was

Im Gegensatz zu einer vertikalen Bewegung kann sich ein bewegliches Hindernis in horizontaler Richtung nur in eine Richtung bewegen. Mit einem kleinen Trick kann man es aber zumindest so aussehen lassen, als würde ein Gegner, der von links heranrast, am rechten Rand umkehren und dann auf der gleichen Bahn zurückkehren.

```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

    Kategorie="Demo"
    Kennung="94fd3402-c064-430e-b899-65b824b0275f"
    Beschreibung="Schritt 11 - Flucht vor dem Killer"
    Bild="bilder/demo/back11.png"
    BildVerloren="bilder/gameover.png"
    BildGewonnen="bilder/gamewon.png"
    InitialePosition="1000, 400"
    MaximaleSprungStaerke="100"
    MaximaleGeschwindigkeit="5"

>

<Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />
```

```

<ElementMitBildSequenz HorizontalePosition="410" VertikalePosition="330"
  Name="linksNachRechts">

  <ElementMitBildSequenz.Bilder>
    <EinzelBild Quelle="bilder/demo/killer.png" />
  </ElementMitBildSequenz.Bilder>

  <ElementMitBildSequenz.Animation>
    <HorizontaleBewegung Geschwindigkeit="5" VonLinksNachRechts="True" />
  </ElementMitBildSequenz.Animation>

  <Enderegel ArtDerKollision="SpielerIstGetroffen" Gewonnen="False" />
</ElementMitBildSequenz>

<ElementMitBildSequenz HorizontalePosition="1390" VertikalePosition="330"
  Name="rechtsNachLinks" AnfänglichUnsichtbar="True">

  <ElementMitBildSequenz.Bilder>
    <EinzelBild Quelle="bilder/demo/killer.png" />
  </ElementMitBildSequenz.Bilder>

  <ElementMitBildSequenz.Animation>
    <HorizontaleBewegung Geschwindigkeit="5" VonLinksNachRechts="False" />
  </ElementMitBildSequenz.Animation>

  <Enderegel ArtDerKollision="SpielerIstGetroffen" Gewonnen="False" />
</ElementMitBildSequenz>

<Element HorizontalePosition="400" VertikalePosition="300" Breite="1" Hoehe="1000">
  <Verschwinderegel ArtDerKollision="GetroffenAberNichtVomSpieler"
    WasSollVerschwinden="Beweglich" />
</Element>

<Element HorizontalePosition="1400" VertikalePosition="300" Breite="1" Hoehe="1000">
  <Erscheinieregel ArtDerKollision="GetroffenAberNichtVomSpieler" Name="rechtsNachLinks" />
  <Verschwinderegel ArtDerKollision="GetroffenAberNichtVomSpieler"
    WasSollVerschwinden="Beweglich" />
</Element>
</Spielfeld>

```

Für die beiden beweglichen Hindernisse wird jeweils das Bild *killer.png* verwendet. Anfänglich sichtbar und damit aktiv ist nur der Gegner, der sich von links nach rechts bewegt. Trifft dieser auf das Hindernis an der rechten Seite, so wird dieser Gegner tatsächlich vom Spielfeld entfernt. Gleichzeitig wird aber ein anderes bewegliches Hindernis an genau der Stelle sichtbar und damit aktiv gemacht, an der der erste Gegner verschwunden ist. Dieser neue Gegner sieht genau so aus wie der erste und bewegt sich nun auf gleicher Höhe von rechts nach links. Im Endeffekt sieht es so aus, als wäre der erste Gegner einfach umgekehrt. Trifft der zweite Gegner auf die linke Seite des Spielfeldes, so wird er vom Spielfeld entfernt und es ist Ruhe.



Die Spielfigur hat immer die Möglichkeit, den heranrasenden Gegner zu überspringen. Aber Vorsicht ist geboten: zumindest einmal kommt er zurück.

2.14 Gar nicht mehr so steif

Ein Hindernis mit Bild ist nicht auf ein einzelnes Bild beschränkt: es ist möglich, ein bewegliches Hindernis mit einer Bildsequenz zu versehen und damit in einer Art Daumenkino eine Bewegung zu simulieren. Ein Beispiel dafür zeigt das folgende Spielfeld.

```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"

  Kategorie="Demo"
  Kennung="8a523ec7-fc9b-4ec3-9d82-00a7530d7fd6"
  Beschreibung="Schritt 12 - Bewegliche Gegner"
  Bild="bilder/demo/back12.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 400"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"

>

<Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />

<ElementMitBildSequenz HorizontalePosition="1200" VertikalePosition="400"
  BilderProSekunde="50" Faellt="True" >

  <ElementMitBildSequenz.Bilder>
    <EinzelBild Quelle="bilder/demo/spin01.png" />
    <EinzelBild Quelle="bilder/demo/spin02.png" />
    <EinzelBild Quelle="bilder/demo/spin03.png" />
    <EinzelBild Quelle="bilder/demo/spin04.png" />
    <EinzelBild Quelle="bilder/demo/spin05.png" />
    <EinzelBild Quelle="bilder/demo/spin06.png" />
    <EinzelBild Quelle="bilder/demo/spin07.png" />
    <EinzelBild Quelle="bilder/demo/spin08.png" />
    <EinzelBild Quelle="bilder/demo/spin09.png" />
    <EinzelBild Quelle="bilder/demo/spin10.png" />
    <EinzelBild Quelle="bilder/demo/spin11.png" />
    <EinzelBild Quelle="bilder/demo/spin12.png" />
  </ElementMitBildSequenz.Bilder>

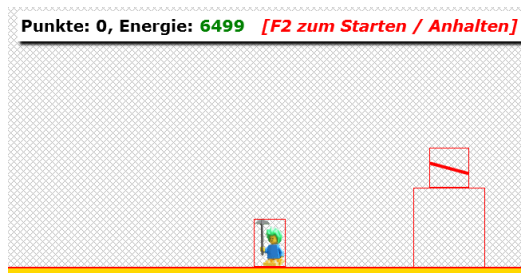
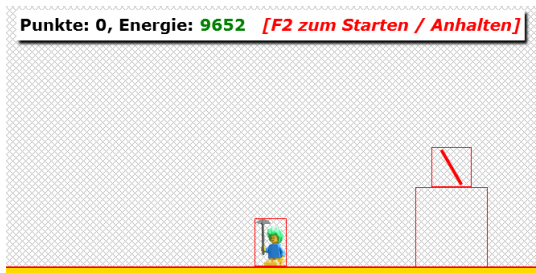
  <Verschieberegel ArtDerKollision="VomSpielerSeitlichGetroffen" />
</ElementMitBildSequenz>

<Element HorizontalePosition="1180" VertikalePosition="300" Breite="90" Hoehe="100">
  <Verschwinderegel ArtDerKollision="VomSpielerGetroffen" />
</Element>

<Element HorizontalePosition="1500" VertikalePosition="300" Breite="10" Hoehe="10">
  <Verschwinderegel ArtDerKollision="GetroffenAberNichtVomSpieler"
    WasSollVerschwinden="Beweglich" />
  <Verschwinderegel ArtDerKollision="VomSpielerGetroffen"/>
</Element>

<Element HorizontalePosition="1600" VertikalePosition="300" Breite="10" Hoehe="100">
  <Enderegel ArtDerKollision="GetroffenAberNichtVomSpieler" Gewonnen="True" />
</Element>
</Spielfeld>
```

Das sich verändernde Hindernis zeigt nacheinander die Bilder *spin01.png* bis *spin12.png* an – danach beginnt die Sequenz wieder von vorne. Die Geschwindigkeit mit der die Einzelbilder aufeinander folgen kann ebenfalls festgelegt werden. Darüber lassen sich dann unterschiedlich schnelle Veränderungen erstellen.



In diesem Spielfeld wird auch eine weitere Eigenschaft von beweglichen Hindernissen verwendet: unser veränderliches Hindernis wird nach unten fallen, sobald es die Gelegenheit dazu erhält. Am Anfang ruht es auf einem anderen Hindernis, das aber bei einer Berührung durch die Spielfigur verschwindet.



Ansonsten bietet das Spielfeld auch eine wirkliche kleine Aufgabe: um zu gewinnen muss die Spielfigur das veränderliche Hindernis bis ran an das rechte, große Hindernis schieben. Berührt es aber vorher das kleine Hindernis davor, so wird das veränderliche Hindernis einfach verschwinden. Das kleine Hindernis entfernt sich selbst, wenn die Spielfigur es berührt.

2.15 Verstecken spielen

Manchmal ist es ganz praktisch wenn die Spielfigur hinter Teilen des Spielfelds herlaufen kann und dadurch eventuell teilweise verdeckt wird. Normalerweise befindet sich die Spielfigur zusammen mit allen Hindernissen auf einer Ebene des Spiels – diese Ebene trägt die Nummer 0. Das Hintergrundbild des Spielfelds liegt immer darunter und wird dementsprechend durch die Spielfigur und die Hindernisse verdeckt – das ist die Ebene mit der Nummer -1. Hindernisse können aber auch in beliebigen Ebenen vor der Spielfigur angebracht werden – das sind dann die Ebenen 1, 2, 3 und so weiter. Die Spielfigur kann die Hindernisse der Ebenen ab 1 allerdings nicht berühren.

```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"
  Kategorie="Demo"
  Kennung="faf4c96d-c0b8-48b2-a694-e28d2f3eb3fb"
  Beschreibung="Schritt 13 - Mehrere Zeichenebenen"
  Bild="bilder/demo/back13.png"
  BildVerloren="bilder/gameover.png"
  BildGewonnen="bilder/gamewon.png"
  InitialePosition="1000, 400"
  MaximaleSprungStaerke="100"
  MaximaleGeschwindigkeit="5"
>
  <Element HorizontalePosition="0" VertikalePosition="300" Breite="4000" Hoehe="1" />
  <ElementMitBildSequenz HorizontalePosition="1200" VertikalePosition="320" Ebene="3">
    <ElementMitBildSequenz.Bilder>
      <EinzelBild Quelle="bilder/demo/vorne.png" />
    </ElementMitBildSequenz.Bilder>
  </ElementMitBildSequenz>
</Spielfeld>
```

```

        </ElementMitBildSequenz.Bilder>
    </ElementMitBildSequenz>

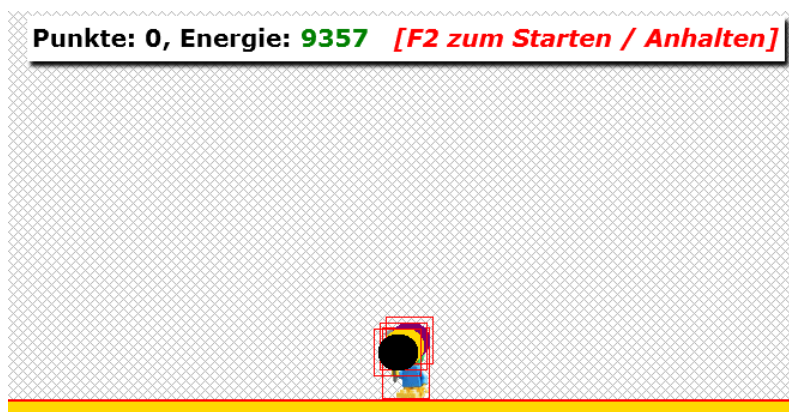
    <ElementMitBildSequenz HorizontalePosition="1210" VertikalePosition="330" Ebene="1">
        <ElementMitBildSequenz.Bilder>
            <EinzelBild Quelle="bilder/demo/hinten.png" />
        </ElementMitBildSequenz.Bilder>
    </ElementMitBildSequenz>

    <ElementMitBildSequenz HorizontalePosition="1205" VertikalePosition="325" Ebene="2">
        <ElementMitBildSequenz.Bilder>
            <EinzelBild Quelle="bilder/demo/mitte.png" />
        </ElementMitBildSequenz.Bilder>
    </ElementMitBildSequenz>

</Spielfeld>

```

Dieses Spielfeld enthält Hindernisse in drei Ebenen mit drei unterschiedlichen Bildern: *vorne.png* ist schwarz, *hinten.png* violett und *mitte.png* gelb. Die Spielfigur kann sich nun hinter diese Hindernisse stellen und wird von diesen teilweise verdeckt – genau wie sich die Hindernisse in den einzelnen Ebenen gegenseitig verdecken.



3 Einige Informationen zur Vertiefung

Wie versprochen hier nun einige Details zur Beschreibung eines Spiels – zum Teil wirklich nur für Profis geeignet, die ein eigenes Spiel erstellen wollen.

3.1 Jetzt kann man sehen, wenn ich laufe

Genau wie veränderliche Hindernisse kann man auch der Spielfigur etwas Bewegung verpassen. Nicht nur ist es möglich, die verschiedenen Bewegungsarten (ruhen, springen, rechts, links) mit unterschiedlichen Bildern zu versehen, sondern man kann auch jeder Bewegungsart mehrere Bilder zuordnen. Bei mehreren Bildern werden diese genau wie bei den veränderlichen Hindernissen mit einer für die jeweilige Spielfigur festgelegten Rate sich immer wiederholend abgespielt.

```

<Figur xmlns="clr-namespace:JMS.JnRV2.Ablage.V1;assembly=GamePersistence"

    SprungStaerke="100"
    SpruengeNacheinander="2"
    BilderProSekunde="8"
    MaximaleGeschwindigkeit="5"
    Name="Jack"
>

    <Figur.BilderRuhend>
        <Bild Quelle="bilder/spieler3ruhend.png" />
    </Figur.BilderRuhend>

    <Figur.BilderNachLinks>
        <Bild Quelle="bilder/spieler3links1.png" />
    </Figur.BilderNachLinks>

```

```

        <Bild Quelle="bilder/spieler3links2.png" />
    </Figur.BilderNachLinks>

    <Figur.BilderNachRechts>
        <Bild Quelle="bilder/spieler3rechts1.png" />
        <Bild Quelle="bilder/spieler3rechts2.png" />
    </Figur.BilderNachRechts>

    <Figur.BilderImSprung>
        <Bild Quelle="bilder/spieler3sprung1.png" />
        <Bild Quelle="bilder/spieler3sprung2.png" />
    </Figur.BilderImSprung>

</Figur>

```

Unser *Jack* ist zwar recht einfach und hat für jede Bewegungsart nur zwei verschiedene Bilder, zeigt aber im laufenden Spiel schon sehr schön den gewünschten Effekt – im Ruhezustand gibt es nur ein Bild.



Die Bilder entsprechen den entsprechenden Bezügen aus der Beschreibung der Spielfigur beginnend oben und links mit *spieler3ruhend.png* bis nach unten und rechts *spieler3sprung3.png*. Eine flüssige Bewegung kommt so natürlich noch nicht zu Stande, da wird man wohl mindestens fünf Bilder pro Bewegungsart brauchen.

3.2 Was muss ich denn nun wirklich tun?

Das Spiel ist ohne Tricks erst einmal nur über einen Web Server direkt nutzbar. Als Grundlage müssen folgende Dateien und Verzeichnisse vorhanden sein:

- Über die Datei *index.html* im Hauptverzeichnis wird das Spiel selbst gestartet. In dieser Datei muss der Name des Spiels gesondert eingetragen werden, ansonsten sollte die Vorlage verwendet werden.

```
<param name="initParams" value="TestRahmen=false,Spiel=DasSpiel" />
```
- Diese erwartet ebenfalls im Hauptverzeichnis die Datei *silverlight.js*.
- Der Programmcode des Spiels selbst wird in der Datei *TheGame.xap* im Unterverzeichnis *ClientBin* erwartet.
- Über den Namen des Spiels wird eine zugehörige Beschreibung im Unterverzeichnis *Spiele* gesucht – im Beispiel wäre das *DasSpiel.xaml*.
- Die Beschreibungen der Spielfiguren befinden sich im Unterverzeichnis *Spielfiguren*, es werden immer alle Dateien beginnend mit *Spieler01.xaml*, dann *Spieler02.xaml* bis hin zu *Spieler99.xaml* berücksichtigt - fehlt eine Nummer, so werden nur die Spielfiguren mit kleineren Nummern erkannt.
- Die Spielfelder werden im Unterverzeichnis *Spielfelder* genau wie die Spielfiguren ermittelt, nur dass hier die Namen *Level01.xaml*, *Level02.xaml* und so weiter verwendet werden.

Für das Aussehen und den Ablauf des Spiels werden dann noch diverse Bild- und Tondateien benötigt. In XAML Beschreibung eines Spielfelds oder einer Spielfigur werden hier immer relative Dateinamen der Art *bilder/hintergrund.png* verwendet. Die zugehörige Datei wird grundsätzlich unterhalb des *ClientBin* Unterverzeichnisses gesucht – im Beispiel also die Datei *hintergrund.png* im Unterverzeichnis *bilder* von *ClientBin*.

4 Details zum Nachschlagen

Hier nun eine vollständige Referenz aller Möglichkeiten, ein Spiel zu beschreiben.

4.1 Spielfigur

```
<Figur xmlns="clr-namespace:JMS.JnRV2.Ablage.V1;assembly=GamePersistence"

    SprungStaerke="100"
    SpruengeNacheinander="2"
    BilderProSekunde="8"
    MaximaleGeschwindigkeit="5"
    Name="Jack"
>

    <Figur.BilderRuhend>
        <Bild Quelle="bilder/spieler3ruhend.png" />
    </Figur.BilderRuhend>

    <Figur.BilderNachLinks>
        <Bild Quelle="bilder/spieler3links1.png" />
        <Bild Quelle="bilder/spieler3links2.png" />
    </Figur.BilderNachLinks>

    <Figur.BilderNachRechts>
        <Bild Quelle="bilder/spieler3rechts1.png" />
        <Bild Quelle="bilder/spieler3rechts2.png" />
    </Figur.BilderNachRechts>

    <Figur.BilderImSprung>
        <Bild Quelle="bilder/spieler3sprung1.png" />
        <Bild Quelle="bilder/spieler3sprung2.png" />
    </Figur.BilderImSprung>

</Figur>
```

Um einem Spieler für alle Spielfelder die freie Wahl der Spielfigur zu ermöglichen wird empfohlen, alle Spielfiguren gleich groß zu gestalten.

4.1.1 Name

Der eindeutige Name der Spielfigur.

4.1.2 SprungStaerke

Legt fest, wie hoch die Spielfigur maximal springen kann – die Angabe erfolgt in Punkten. Je nach Spielfeld kann diese Höhe noch weiter beschränkt werden. Als Standardwert wird 100 empfohlen.

4.1.3 SpruengeNacheinander

Die Anzahl der Sprünge, die eine Spielfigur unmittelbar hintereinander ausführen kann – als Standardwert wird 2 empfohlen. Spring eine Figur in die Höhe, so beginnt sie sofort wieder nach unten zu fallen. Die hier festgelegte Zahl bestimmt, ob während des Fallens ein erneuter Sprung ausgelöst werden kann. Die Anzahl der unmittelbar aufeinander folgenden Sprünge sollte nicht allzu hoch gewählt werden, da es ansonsten so aussehen kann, als könnte die Spielfigur fliegen.

4.1.4 MaximaleGeschwindigkeit

Eine relative Angabe für die maximale Geschwindigkeit der Spielfigur beim Laufen nach rechts oder links – empfohlen wird als Standardwert 5. In einem Spielfeld kann eine Beschränkung auf einen kleineren Wert vorgenommen werden.

4.1.5 BilderProSekunde

Die Geschwindigkeit, mit der mehrere Bilder einer Bewegungsart hintereinander abgespielt werden. Ist allen Bewegungsarten nur ein einziges Bild zugeordnet, so wird diese Einstellung ignoriert. An-

sonsten hängt sie von den jeweiligen Bildsequenzen ab und sollte so gewählt werden, dass flüssige Bewegungen entstehen.

Man beachte, dass diese Einstellung nur für alle Bewegungsarten gemeinsam eingestellt werden kann.

4.1.6 BilderRuhend / BilderNachLinks / BilderNachRechts / BilderImSprung

Pro Bewegungsart lässt sich eine Sequenz von Einzelbildern festlegen, die in der angegebenen Reihenfolge abgespielt werden – nach dem letzten Bild der Sequenz wieder beginnend bei dem ersten.

```
<Bild Quelle="bilder/spieler3ruhend.png" />
```

Die angegebene Bilddatei wird immer im Unterverzeichnis *ClientBin* erwartet – im Beispiel die Datei *spieler3ruhend.png* in einem weiteren Unterverzeichnis *bilder* von *ClientBin*.

4.2 Spiel

```
<Spiel xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"
```

```
    SpielfeldKategorien="Demo"  
    SichtfensterBreite="800"  
    SichtfensterHoehe="600"
```

```
/>
```

4.2.1 SichtfensterBreite / SichtfensterHoehe

Die Ausdehnung des Sichtfensters in Punkten. Da verschiedene Spieler mit verschiedenen großen Browserfenstern arbeiten sollte man es hier nicht übertreiben. Der vorgeschlagene Standard ist mit 800 Punkten Breite und 600 Punkten Höhe allerdings inzwischen doch eher etwas zu mager.

4.2.2 SpielfeldKategorien

Hier kann eine durch Kommas getrennte Liste von Spielfeldarten stehen - im Beispiel wird als einzige Art *Demo* verwendet. Ist eine Liste angegeben, so werden nur solche Spielfelder als zum Spiel gehörend betrachtet, die eine entsprechende Art haben. Ist hingegen kein Wert angegeben, so werden nur die Spielfelder berücksichtigt, bei denen die Art nicht definiert ist.

4.3 Spielfeld

```
<Spielfeld xmlns="clr-namespace:JMS.JnRV2.Ablage;assembly=GamePersistence"
```

```
    Kategorie="Demo"  
    Kennung="faf4c96d-c0b8-48b2-a694-e28d2f3eb3fb"  
    Beschreibung="Schritt 13 - Mehrere Zeichenebenen"  
    Bild="bilder/demo/back13.png"  
    BildVerloren="bilder/gameover.png"  
    BildGewonnen="bilder/gamewon.png"  
    InitialePosition="1000, 400"  
    InitialeLebensenergie="10000"  
    MaximaleSprungStaerke="100"  
    MaximaleGeschwindigkeit="5"
```

```
>
```

```
<Element ... />
```

```
<ElementMitBildSequenz ... />
```

```
</Spielfeld>
```

4.3.1 Kategorie

Optional die Art des Spielfeldes, über die eine Zuordnung zu Spielen stattfindet. Darüber kann ein Spielfeld durchaus in verschiedenen Spielen genutzt werden.

4.3.2 Beschreibung

Der Name des Spielfelds, so wie ihn der Spieler zur Auswahl angeboten bekommt.

4.3.3 Bild

Der Name der Datei mit dem Hintergrundbild. Bei den Standardgrößen für das Sichtfenster (800 Punkte breit und 600 Punkte hoch) und die Spielfigur (40 Punkte breit und 60 Punkte hoch) sollte man sich hier etwas zurückhalten. Eine Breite von 4.000 Punkten kann für normale Spielfelder schon ausreichend sein.

4.3.4 BildVerloren / BildGewonnen

Diese Bilddateien werden angezeigt, wenn die mit dem Spielfeld verbundene Spielrunde gewonnen oder verloren wurde.

4.3.5 InitialePosition

Die Position der Spielfigur beim Starten der zu dem Spielfeld gehörenden Spielrunde. Angegeben wird durch ein Komma getrennt der Abstand vom linken und unteren Rand des Spielfelds, jeweils in Punkten. Die Angabe selbst bezieht sich auf den Mittelpunkt der Spielfigur und nicht etwa auf einen der Eckpunkte des zugehörigen Rechtecks.

4.3.6 InitialeLebensenergie

Der anfängliche Energiepegel der Spielfigur, der Standardwert ist 10.000. Der Energiepegel verringert sich ständig, sobald das Spiel einmal läuft und nicht angehalten wurde. Auch die Energieregeln kann dazu führen, dass sich dieser Wert verändert. Hat der Energiepegel einmal den Wert 0 erreicht oder unterschritten, so gilt das Spiel als verloren.

4.3.7 MaximaleSprungStaerke / MaximaleGeschwindigkeit

Mit diesen Werten ist es möglich, die Fähigkeiten einer Spielfigur für dieses Spielfeld auf die angegebenen Werte zu beschränken. Die Standardwerte sind wie für die Spielfiguren vorgeschlagen 100 und 5.

4.3.8 Kennung

Jedes Spielfeld muss einen eindeutigen Namen erhalten, über den zum Beispiel die Spielergebnisse verwaltet werden. Hier werden so genannte eindeutige Kennungen (http://de.wikipedia.org/wiki/Globally_Unique_Identifier) verwendet, die man sich Online leicht erstellen lassen kann (<http://guid.us>).

4.4 Hindernis auf dem Hintergrund

```
<Element
  AnfänglichUnsichtbar="False"
  Breite="100"
  Hoehe="100"
  HorizontalePosition="500"
  Melodie="music/boing.mp3"
  Name="mauer"
  VertikalePosition="300" >

  <Enderegel ... />
  <Energierregel ... />
  <Erscheineregel ... />
  <Punkteregel ... />
  <Verschieberegel ... />
  <Verschwinderegel ... />
  ...

</Element>
```

Auf die einzelnen Regeln wird gleich noch gesondert eingegangen.

4.4.1 HorizontalePosition / VertikalePosition

Legt fest, über welchen Teil des Hintergrunds das Hindernis gelegt werden soll. Die Angabe in Punkten erfolgt für den Abstand vom linken und unteren Rand des Spielfelds. Anders als bei der Anfangsposition der Spielfigur bezieht sich die Angabe aber auf die linke untere Ecke des Hindernisses.

4.4.2 Breite / Hoehe

Die Breite und Höhe des Hindernisses jeweils in Punkten angegeben.

4.4.3 AnfänglichUnsichtbar

Kann auf Wunsch auf den Wert *True* gesetzt werden um das Hindernis erst einmal nicht zu berücksichtigen. Es kann dann im Verlaufe des Spiels durch eine Regel hinzugenommen werden.

4.4.4 Name

Der eindeutige Name des Hindernisses. Eine Angabe ist nur erforderlich, wenn das Hindernis im Spielverlauf ein- oder ausgeschaltet werden sollen. Hindernisse haben im Allgemeinen keinen Namen.

4.4.5 Melodie

Ist hier eine MP3 Datei angegeben so wird diese immer abgespielt, wenn das Hindernis von der Spielfigur getroffen wird.

4.5 Bewegliches Hindernis

Bewegliche Hindernisse und Hindernisse auf dem Hintergrund haben eine ganze Reihe Einstellungen gemeinsam, die hier nicht noch einmal erläutert werden sollen.

```
<ElementMitBildSequenz
  AnfänglichUnsichtbar="False"
  HorizontalePosition="500"
  Melodie="music/boing.mp3"
  Name="mauer"
  VertikalePosition="300"
  BilderProSekunde="20"
  Ebene="0"
  Faellt="False"
>

  <Enderegel ... />
  <Energierregel ... />
  <Erscheineregel ... />
  <Punkteregel ... />
  <Verschieberegel ... />
  <Verschwinderegel ... />
  ...

  <ElementMitBildSequenz.Bilder>
    <EinzelBild Quelle="bilder/eins.png" />
    ...
  </ElementMitBildSequenz.Bilder>

  <ElementMitBildSequenz.Animation>
    ...
  </ElementMitBildSequenz.Animation>
</ElementMitBildSequenz>
```

Auf die einzelnen Regeln und Animationen wird gleich noch gesondert eingegangen. Die Angabe einer Animationsregel ist nicht zwingend und beschreibt bei Vorhandensein die selbstständige Eigenbewegung des Hindernisses.

4.5.1 Bilder / Breite / Hoehe

Es muss immer mindestens eine Bilddatei angegeben werden – soll das Hindernis unsichtbar bleiben ist es ein leichtes, ein vollständig transparentes Bild zu erstellen. Es wird empfohlen sich auf die Ausdehnungen des Bildes zu verlassen und auf eine zusätzliche Angabe von Breite und Höhe zu verzichten.

Wenn mehr als ein Bild angegeben wird so müssen alle Bilddateien exakt die gleichen Ausdehnungen haben. Die Bilder werden dann nacheinander abgespielt und diese Sequenz nach dem letzten Bild wieder von vorne begonnen.

4.5.2 BilderProSekunde

Wenn mehr als eine Bilddatei verwendet wird so legt diese Zahl fest, wie schnell die Bilder wechseln sollen. Zu kleine Werte führen zu einer ruckelnden Veränderung des Hindernisses, zu große Werte können die Bedienbarkeit des Spiels als Ganzes negativ beeinflussen.

4.5.3 Faellt

Diese Einstellung kann auf *True* gesetzt werden um das Hindernis dazu zu bringen wie die Spielfigur nach unten zu fallen. Ohne eine ausdrückliche Änderung bleiben Hindernisse in der Luft an der Position hängen, an der sie ursprünglich erschienen sind – wenn die Spielfigur sie nicht verschieben kann.

4.5.4 Ebene

Die Standardeinstellung 0 bewirkt, dass das Hindernis mit der Spielfigur in Berührung kommen kann. Größere Werte führen dazu, dass das Hindernis vor die Spielfigur gezeichnet wird und von dieser nicht berührt werden kann. Die Spielfigur wird dann zwischen Hintergrund und Hindernis gezeichnet und erscheint eventuell teilweise oder vollständig verdeckt. Negative Werte sind hier nicht erlaubt.

4.6 Regel

Jedes Hindernis kann eine beliebige Anzahl von Regeln haben die ausgelöst werden, wenn das Hindernis von irgendetwas berührt wird.

```
<Enderegel ArtDerKollision="Getroffen" Gewonnen="True" />
<Energierregel ArtDerKollision="Getroffen" Lebensenergie="1000" />
<Erscheineregel ArtDerKollision="Getroffen" Name="decke" />
<Punkteregel ArtDerKollision="Getroffen" Punkte="1000" />
<Verschieberegel ArtDerKollision="Getroffen" />
<Verschwinderegel ArtDerKollision="Getroffen" WasSollVerschwinden="Beweglich" />
```

4.6.1 ArtDerKollision

Diese Eigenschaft ist allen Regeln gemeinsam und legt fest, auf welche Art der Berührung die Regel ausgelöst werden soll:

- *Getroffen*: das Hindernis wurde von irgendetwas berührt.
- *VomSpielerGetroffen*: das Hindernis wurde von der Spielfigur berührt
- *GetroffenAberNichtVomSpieler*: das Hindernis wurde von einem anderen Hindernis berührt
- *VomSpielerSeitlichGetroffen*: die Spielfigur ist in das Hindernis hinein gelaufen
- *SpielerIstGetroffen*: die Spielfigur wurde von einem Hindernis getroffen

4.6.2 Enderegel

Die Spielrunde wird beendet und die Einstellung *Gewonnen* legt fest, ob das Spiel als gewonnen (*True*) oder verloren (*False*) gewertet werden soll.

4.6.3 Energieregeln

Der Energiepegel der Spielfigur wird um den Wert der Einstellung *Lebensenergie* verändert – erhöht oder verringert, je nach Vorzeichen. Sobald der Energiepegel den Wert 0 erreicht oder sogar unterschreitet gilt das Spiel als verloren.

4.6.4 Erscheinerregeln

Macht ein anderes Hindernis sichtbar und lässt es damit am Spiel teilnehmen. Die Eigenschaft *Name* wird zur Referenz auf das gewünschte Hindernis verwendet.

4.6.5 Punkteregeln

Verändert den Punktestand um den in der Einstellung *Punkte* angegebenen Wert. Dieser kann positiv oder negativ sein. Der Punktestand wird nur ausgewertet, wenn das Spiel gewonnen wird.

4.6.6 Verschieberegeln

Ein Hindernis mit dieser Regel kann seine Position durch Verschieben verändern.

4.6.7 Verschwinderegeln

Wenn diese Regel aktiv wird, so wird das mit der Eigenschaft *WasSollVerschwinden* bezeichnete Hindernis aus dem Spiel entfernt. Folgende Alternativen sind dabei vorgesehen:

- *Selbst*: das Hindernis, zu dem die Regel gehört, verschwindet
- *Beweglich*: das Hindernis, durch dessen Bewegung die Berührung ausgelöst wurde, verschwindet
- *Fest*: das Hindernis, dass durch eine Bewegung eines anderen Hindernisses berührt wurde, verschwindet

Die Spielfigur kann über diese Regel nicht aus dem Spiel entfernt werden.

4.7 Animation

Hindernisse können nicht nur durch Regeln oder Fallen ihre Position auf dem Spielfeld verändern, sondern auch selbstständig Eigenbewegungen ausführen.

```
<BewegungRelativZurSpielfigur Angriff="False" HorizontaleGeschwindigkeit="4" />  
<HorizontaleBewegung Geschwindigkeit="4" VonLinksNachRechts="True" />  
<VertikaleBewegung Sprunghöhe="100" Sprungzeit="2" />
```

4.7.1 VertikaleBewegung

Mit dieser Eigenbewegung hüpft das Hindernis auf und ab. Die Einstellung *Sprunghöhe* legt fest, wie weit die Bewegung nach oben erfolgen soll – die Angabe erfolgt in Punkten. Die *Sprungzeit* bestimmt zusätzlich, wie lange das Hindernis braucht, um die Sprunghöhe zu erreichen. Ist der Sprung einmal begonnen, so fängt das Hindernis auch sofort an zu fallen. Ist die Sprungzeit erreicht, fällt es nur noch nach unten. Unten angekommen beginnt die Bewegung von neuem.

Manchmal muss man etwas mit den beiden Einstellungen herumexperimentieren um den gewünschten Effekt zu erreichen.

4.7.2 HorizontaleBewegung

Die Einstellung *Geschwindigkeit* legt die relative *Geschwindigkeit* fest, mit der sich das Hindernis horizontal bewegen soll. Diese Angabe verwendet die gleichen Werte wie die maximale Geschwindigkeit der Spielfigur auf dem Spielfeld. Interessante Effekte können sich übrigens bei der Kombination einer horizontalen Bewegung und einem fallenden Hindernis ergeben.

Die zusätzliche Einstellung *VonLinksNachRechts* legt fest, ob sich das Hindernis auf dem Spielfeld von links nach rechts (*True*) oder von rechts nach links (*False*) bewegt.

4.7.3 BewegungRelativZurSpielfigur

Mit der Einstellung *Angriff* wird festgelegt, ob sich das Hindernis in Richtung (*True*) der Spielfigur bewegt und diese quasi angreift oder sich von der Spielfigur entfernt (*False*) also scheinbar flieht. Die *HorizontaleGeschwindigkeit* bestimmt dabei, wie schnell sich das Hindernis bewegt. Es wird die gleiche relative Angabe wie für die maximale Geschwindigkeit der Spielfigur auf dem Spielfeld verwendet. Zu große Werte können daher leicht dazu führen, dass die Spielfigur keine Chance hat, einem Angriff zu entgehen oder ein flüchtendes Hindernis einzuholen.

4.8 HTML Startseite

Es wird empfohlen die Vorlage *index.html* als Startseite für ein eigenes Spiel zu verwenden. Für diesen Fall muss lediglich eine Zeile angepasst werden.

```
<param name="initParams" value="TestRahmen=false,Spiel=DasSpiel" />
```

Als Wert für *Spiel* muss der Name der Beschreibungsdatei des Spiels angegeben werden – die Voreinstellung würde nach der Datei *DasSpiel.xaml* im Unterverzeichnis *Spiele* suchen. Zur Aktivierung des Testmodus mit den roten Rahmen um die Spielfigur und die Hindernisse muss einfach nur der Wert von *TestRahmen* auf *true* verändert werden. Oft macht es auch Sinn einfach zwei Startseiten bereit zu stellen: eine mit und eine ohne die Rahmen.