

Mathelernspiel

Daniel Knorn

Hochschule Fulda

Matrikel-Nr.: 132104

Lärchenstr. 3, 36124 Eichenzell

daniel.knorn@informatik.hs-fulda.de

Oliver Krah

Hochschule Fulda

Matrikel-Nr.: 531797

Winterbergstr. 8, 36119 Neuhof

oliver.krah@informatik.hs-fulda.de

ABSTRAKTE BESCHREIBUNG

In dieser Ausarbeitung beschreiben wir die Herangehensweise und Entwicklung eines Interaktiven Systems, im Rahmen eines Projektes, des Moduls *Entwicklung interaktiver Systeme*, der Hochschule Fulda. Zu den verwendeten Technologien zählen in diesem Projekt X3DOM und jQuery. Das System basiert auf einer Webanwendung, mit dem Hintergrund die mathematischen Grundlagen der Geometrie auf spielerische Art und Weise zu vermitteln.

SCHLÜSSELWÖRTER

X3DOM; jQuery; Geometrie; Lernspiel;

EINFÜHRUNG

Bei der Anwendung handelt es sich um eine spielerische Applikation mit dem Ziel, die Grundlagen der Geometrie nahezubringen. Dabei stehen einige Geometrische Figuren zur Verfügung, bei denen jeweils das Volumen oder die Oberfläche zu berechnen ist. Im Folgenden wird erläutert wie das Spiel genau funktioniert, welche Geometrien benutzt wurden, wie diese erstellt wurden und welche Funktionalitäten implementiert wurden.

DIE IDEE

Der Grundgedanke war auf einfache Art die Geometrie näher zu bringen und diese Erfahrung mit einer Webanwendung zu verknüpfen. Als Lernziel gilt es sich Formeln und Vorgehensweisen einzuprägen und diese bei Bedarf anwenden zu können.

DAS SPIELPRINZIP

Das Prinzip ist einfach. Es geht darum, so viele Fragen wie möglich richtig zu beantworten. Dabei werden dem Spieler zu jeder Frage auch eine entsprechende Geometrische Figur gezeigt. Zu jeder Figur gibt es verfügbare Daten, mit Hilfe derer das Volumen oder die Oberfläche zu berechnen ist. Hat der Anwender die Aufgabe erfolgreich gelöst, geht es weiter mit einem komplexeren Gebilde, bis schlussendlich die letzte Frage beantwortet wurde.

DAS INTERFACE

Das Interface gliedert sich in zwei Kategorien, zum einen das Startinterface und zum Anderen das Spielinterface. Zum Spielstart sieht man oben links einen ausgegrauten Button. Dieser wird erst während des Spiels relevant. Neben dem Titel der Anwendung, gibt es oben rechts einen Button der zum eigentlichen Spiel führt. (siehe Abb. 1)

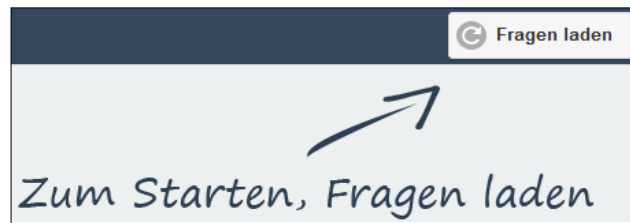


Abbildung 1: Interface, bevor Spiel gestartet wurde

Nachdem dieser geklickt wurde erscheint das Spielinterface, indem es immer noch die gleiche Kopfzeile gibt. Lediglich mit dem Unterschied, das nun der rechte Button ausgegraut ist und nicht der linke. Wird nun der Button „Fragen“ geklickt, erscheint in der linken Hälfte des Bildschirms ein Fenster, welches einen Überblick über alle vorhandenen Fragen gestattet.(siehe Abb.2) Im Zentrum des Bildes sieht man die Geometrische Figur, welche sich nach Belieben drehen lässt. Besteht die Figur aus Teilobjekten, lassen sich diese selektieren. Ist ein Objekt selektiert, so erscheinen am rechten Bildschirmrand die Informationen über dieses Objekt. Am unteren Bildschirmrand kann man die eigentliche Frage zur gezeigten Figur, die entsprechend benötigte Formel und das Eingabefeld zur Antwort finden. Links neben der Frage befindet sich ein kleiner Tipp. Rechts neben dem Eingabefeld befindet sich der Button "Antwort validieren", mit Hilfe dessen der Benutzer seine Antwort auf Richtigkeit testen kann.

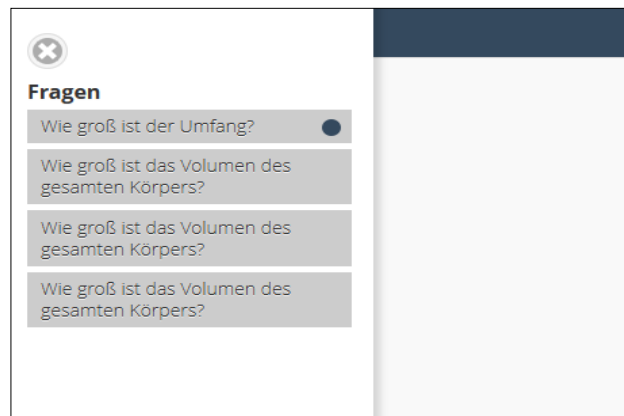


Abbildung 2: Übersicht der Fragen

ALLGEMEINER SPIELABLAUF

Zu Beginn des Spiels müssen zuerst die Fragen geladen werden, was äquivalent zu „Spiel starten“ ist. Wurde das Spiel gestartet, ist das Spielinterface sichtbar und das Spiel kann beginnen. Nun gilt es, die gestellte Frage zu beantworten. Zu dieser gibt es noch zusätzlich die Formel. Damit der Benutzer auch die Werte erhält, mit denen er das gesuchte Ergebnis berechnen soll, muss dieser das Objekt selektieren. Hierbei kann es sich um ein einzelnes Objekt oder um ein zusammengesetztes Objekt handeln. Sind mehrere Geometrien zu sehen, so müssen diese einzeln selektiert und berechnet werden. Die dazu benötigten Werte erscheinen beim Selektieren am rechten Bildschirmrand. Ist ein Objekt selektiert, wird dieses bläulich eingefärbt. (siehe Abb. 3) So kann der Benutzer genau erkennen, welches Objekt er ausgewählt hat. Zuletzt müssen die Zwischenergebnisse dann aufsummiert werden.

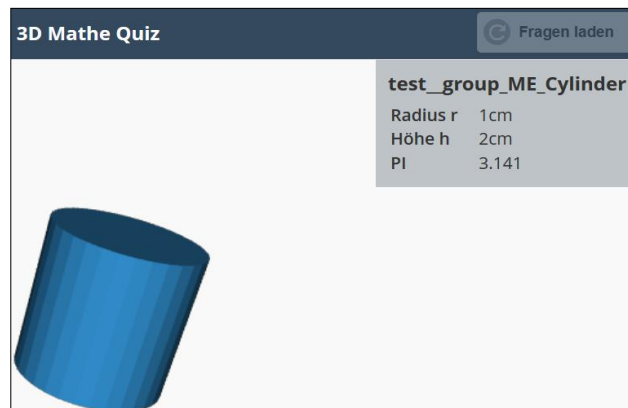


Abbildung 3: Selektiertes Objekt

Nachdem der Anwender das Ergebnis berechnet hat, kann er es nun als Antwort in das Eingabefeld eintragen und auf „Antwort validieren“ klicken. Ist die Antwort falsch, färbt sich der untere Bereich rot (siehe Abb. 4), was signalisiert, dass die Antwort falsch war. Ist die Antwort richtig, färbt sich der Bereich kurz grün, bevor dann die nächste Frage geladen wird und der Bereich sich wieder grau färbt.



Abbildung 4: Falsche Antwort

Wurden alle Fragen korrekt beantwortet, wird ein neues Interface als neue Ebene geladen und über das eigentliche Interface gelegt. Auf diesem findet man lediglich den Schriftzug „GEWONNEN“. (siehe Abb. 5)



Abbildung 5: Das Spiel wurden gewonnen

DIE GEOMETRIEN

Was wäre ein Lernspiel, zur Abfrage geometrischer Objekte, ohne solche im Spiel implementiert zu haben. Es ist eigentlich offensichtlich, dass es sich hierbei um eine der Kernfunktionen der Anwendung handelt, da es ohne Geometrien schlichtweg unmöglich ist ein solches Lernspiel zu entwickeln. Deshalb wird an dieser Stelle beschrieben, welche Objekte im Spiel verfügbar sind und wie diese erstellt bzw. implementiert wurden. Innerhalb der Anwendung gibt es folgende Konstrukte:

Kugel, Quader, Würfel, Zylinder, Kegel, Hochhaus, Gebäude, Kreuz, Turm

Blender

Alle geometrischen Figuren, die für das Lernspiel zur Verfügung stehen, wurden vorher mit Blender 2.69 (Open Source erhältlich auf <http://www.blender.org>) von uns modelliert. Zylinder, Kugel, Quader und Kegel können dabei relativ einfach und schnell als Standardobjekte erstellt werden. Somit mussten nur Skalierungen auf X, Y und Z-Achse erfolgen, so dass die Figuren auch den Werten, mit denen gerechnet werden soll, entsprechen. Die zusammengesetzten Objekte bestehen aus diesen Standardfiguren und wurden ebenfalls dementsprechend skaliert (siehe Abb.6).

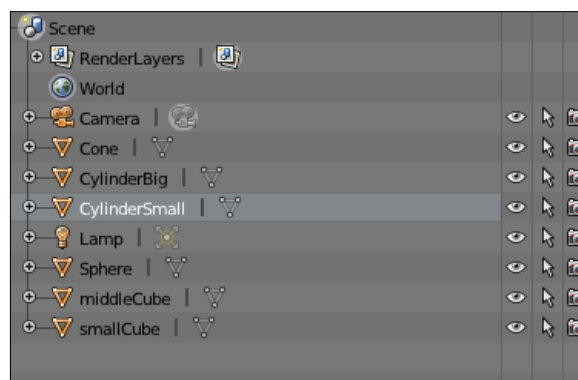


Abbildung 6: Objekt in Blender

Auch wurde jede Figur, aus dem das Gebilde besteht, in Blender als eigenes Objekt erstellt und dementsprechend benannt(siehe Abb.7), was beim Exportieren enorme Vorteile bringt. Schlussendlich müssen diese als *X3D Extensible 3D(.x3d)* exportiert werden. Das hat den Vorteil, dass man diese Dateien anschließend in X3DOM so verändern kann, dass sie beispielsweise unterschiedliche Farben haben. Auch von großem Vorteil war hier, dass man auf die einzelnen Objekte, innerhalb der .x3d Datei, zugreifen konnte. Das ermöglichte uns das bereits beschriebene Selektieren von Teilobjekten.

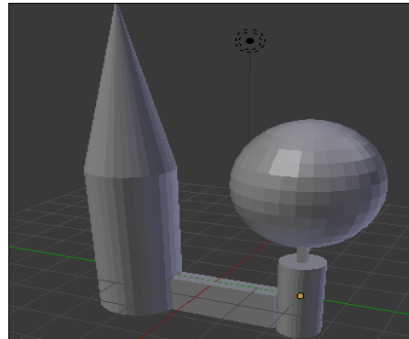


Abbildung 7: Einzelne Objekte innerhalb der Datei

Implementierung der Objekte

Bevor man die x3d-Dateien implementieren kann, müssen diese noch modifiziert werden. Hierbei wurde mit einem Editor in die Datei, unter Anderem, ein Group-Tag hinzugefügt, welches alle enthaltenen Teilobjekte umschließt. Dies ermöglicht, dass auf die einzelnen Teilobjekte innerhalb der Datei zugegriffen werden kann. Ebenfalls wurden in der Datei die Metainformationen entfernt und bei jedem Objekt der Material-Tag mit dem umschließenden Appearance-Tag hinzugefügt. Damit wurde nun die Materialfarbe beschrieben. (siehe Abb. 8) Anschließend wurden die gesamten Objekte implementiert. Hierfür wurden die Dateien in einen Unterordner des Projektes gelegt und später über diesen Pfad im Script geladen.

```
<Group DEF="group_ME_Cone">
  <Shape>
    <Appearance>
      <material diffuseColor='gray'>
      </material>
    </Appearance>
```

Abbildung 8: Gruppierung und Farbgebung

DIE FRAGEN

Für ein Lernspiel sind die Fragen natürlich ebenfalls immens wichtig und irgendwo müssen diese auch implementiert werden. Alle Fragen, die im Spiel gestellt werden, werden alle in eine *JSON-Datei* geschrieben und diese werden anschließend im Script aufgerufen. Dazu später mehr. Die Datei liefert das zu ladende Model, alle verfügbaren Fragen zu diesem Model (bestehend aus Frage, Formel, Ergebnis und Einheit) und die Tipps, die für einzelne Teilobjekte verfügbar sind. (siehe Abb. 9) Dadurch, dass die Fragen in einer separaten Datei stehen, erhalten wir den Vorteil, dass das Spiel einfach und schnell durch neue Fragen erweiterbar ist. Aber auch die Werte der Modelle, die in einer *JSON-Datei* abgelegt sind, bringen einige Vorteile mit sich. Es ist nämlich dadurch möglich die Werte nach Belieben zu ändern und somit den Schwierigkeitsgrad zu erhöhen. Ebenfalls wird dadurch eine klare Struktur erzwungen, die das gesamte Projekt übersichtlicher macht.

```
"1": {
  "model": "zylinder",
  "fragen": {
    "0": {
      "frage": "Wie grosszig; ist der Umfang?",
      "formel": "U = 2 * pi * r",
      "ergebnis": "6.282",
      "einheit": "cm^sup2;"
    }
  },
  "tips": {
    "group_ME_Cylinder": {
      "1": {
        "name": "Radius r",
        "wert": "1cm"
      },
      "2": {
        "name": "H\u00f6he h",
        "wert": "2cm"
      },
      "3": {
        "name": "PI",
        "wert": "3.141"
      }
    }
  }
},
```

Abbildung 9: die JSON-Datei

TECHNISCHER SPIELABLAUF

Der allgemeine Spielablauf wurde bereits erläutert. Allerdings stellt sich dabei die Frage, wie dieser Ablauf realisiert wurde. Im Folgenden wird Schritt für Schritt erklärt, was an welcher Stelle im Spiel, im Hintergrund geschieht.

Schritt 1: Seite laden

Zu Beginn wird die gesamte Seite geladen, dabei auch das eigentliche Interface, welches als Ebene unter dem Startbildschirm liegt.

Schritt 2: JSON-Datei auslesen

Wie bereits erwähnt, beinhaltet diese Datei sämtliche Informationen über Werte, Fragen und Antworten und diese müssen schließlich auch ausgelesen werden. Dabei werden die einzelnen Daten in einem ausgelesen und in einem Array abgespeichert. Die Fragen werden nun in der Frageliste ausgegeben, welche über den Button „Fragen“ angezeigt werden können. Dort ist auch die Anzahl der Fragen einzusehen. Sind die Fragen geladen, wird der Button „Fragen laden“ grau. Damit ist die Funktion deaktiviert, da man diese im weiteren Spielverlauf schlichtweg nicht mehr benötigt. Anschließend wird die erste Übung als aktive Frage deklariert und wird auf der Frageliste als *aktuell* gekennzeichnet. (siehe Abb.10) Dies geschieht über einen kleinen Punkt neben der Frage. Nun wird die Frage, mit Formel und der Einheit, in der das Ergebnis geliefert werden muss, in der unteren Bildschirmhälfte angezeigt. Zuletzt wird das Model in den Anzeigebereich geladen und angezeigt.

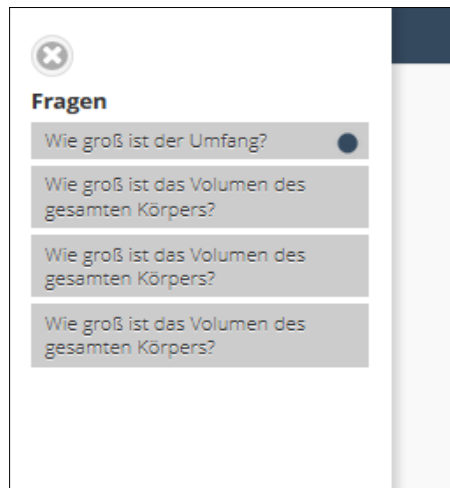


Abbildung 10: Frageliste

Schritt 3: Anzeigen der Werte

Um das Ergebnis zu berechnen werden natürlich die Werte benötigt. Wie bereits beschrieben, erfolgt das durch Selektieren des Objektes. Wird ein Objekt selektiert, wird das Fenster am rechten Bildschirmrand sichtbar. Die zugehörigen Daten, die zuvor aus der JSON-Datei gelesen wurden und sich schon im Array befinden, werden hier nun ausgegeben. Deselektiert der Anwender das Model wieder, so verschwindet der Layer wieder.

```
function showObjInfo(obj) {
    $('#questionTips').empty();
    $('#questionTips').append('<h4>' + obj + '</h4><table>
</table>');

    var groupDef = obj.replace(/test_/g, '');

    $.each(activeBlock.tips[groupDef], function(i, tip) {
        $('#questionTips table').append('<tr><td>'
+ tip.name + '</td><td>' + tip.wert + '</td></tr>');
    });
}
```

Abbildung 11: Daten anzeigen

Schritt 4: Antwort validieren

Ist das Ergebnis berechnet worden, soll dieses mit der Lösung abgeglichen werden. Hierbei klickt der Anwender auf den Button „Antwort validieren“. Beim Validieren wird der Eingabewert in dem Inputfeld mit dem Wert aus der JSON-Datei abgeglichen(siehe Abb.12). Ist dieser korrekt, färbt sich der untere Bildschirmrand grün. Bei falscher Antwort rot und der Spieler muss erneut einen Wert eingeben. Egal in welcher Farbe sich der Antwortbereich färbt, es geschieht auf dieselbe Art. Dabei wird allerdings nicht wie bisher eine neue Ebene über das Interface gelegt, sondern lediglich die Hintergrundfarbe des *Frageboards* geändert.

```
function checkAnswer() {
    var userAnswer = $('#questionContent input
[name="answer"]').val();

    if(userAnswer == activeQuestion.ergebnis)
        return true;

    return false;
}
```

Abbildung 12: Antwort validieren

Schritt 5: Nächste Frage laden

Wenn die richtige Lösung validiert wurde, muss die nächste Frage geladen werden. Dies funktioniert, indem der Fragebereich und natürlich auch die Frageliste aktualisiert werden (siehe Abb.13). In der Frageliste wird nun wie gewohnt die aktuelle Frage gekennzeichnet. Ebenfalls muss das neue Model geladen werden, damit das Objekt auch der Frage entspricht. Schlussendlich wird der Inhalt des Eingabefelds gelöscht, damit eine neue Lösung eingegeben werden kann.

```
function loadNextQuestion() {
    //active question number | next question number
    var aqn = parseInt($('#questions li.active').attr('data-number'));
    var nqn = aqn;

    //active block number
    var abn = parseInt($('#questions li.active').attr('data-block')) + 1;

    if(!$('#questions li.active').length) {
        nqn = 0;
        abn = 1;
    }

    if(typeof questions[abn] == 'undefined')
        return false

    //aktive Frage setzen
    activeBlock = questions[abn];
    activeQuestion = questions[abn].fragen[0];

    //Fragebereich aktualisieren
    $('#questionContent .question').html('
    <span>Frage ' + abn + ' :</span> ' + activeQuestion.frage);
    $('#questionContent .info span').html
    ('Antwort in ' + activeQuestion.einheit);
    $('#questionContent .formel span').html(activeQuestion.formel);

    //Sidebar aktualisieren
    $('#questions li.active').removeClass('active');
    $('#questions li[data-number="' + abn + '"]').addClass('active');

    //Model aktualisieren
    $('#the3DContent inline').attr('url',
    'models/' + activeBlock.model + '.x3d');
```

Abbildung 13: Neue Frage laden

Schritt 6: Gewonnen

Hat der Spieler die letzte Frage richtig beantwortet, so hat er gewonnen. Dabei wird wieder eine neue Ebene über das Interface gelegt, auf der „GEWONNEN“ steht.

ERWEITERBARKEIT

Inwiefern ist das Mathe Lernspiel noch zu erweitern, da die Grundfunktionen schließlich implementiert sind? Das ist korrekt! Allerdings gibt es noch einige Dinge, die beispielsweise noch integriert werden könnten. Unter Anderem so etwas wie ein Schwierigkeitsgrad. So gibt es zum Beispiel verschiedene Schulklassen, die natürlich auch verschiedenen Stoff behandeln. So könnte man das Lernspiel um mehr Themengebiete der Mathematik erweitern. Aber um bei den Geometrischen Figuren zu bleiben, könnte man auch dort unterschiedlich komplexe Gebilde verwenden. Eine Einstellungsmöglichkeit zu Beginn des Spielstarts wäre eine Möglichkeit. Dabei kann der Anwender zwischen möglichen Schwierigkeitsstufen auswählen, die seinem Können entsprechen. Dadurch, dass wir, wie bereits beschrieben, eine Datei haben, die alle Fragen und Antworten enthält, ist es natürlich relativ einfach so etwas zu realisieren. Ebenfalls ist es möglich die Werte der einzelnen Modelle noch ein wenig zu verkomplizieren. Statt einer Seitenlänge oder einem Radius von zwei, könnte man höhere Werte oder gar Fließkommazahlen verwenden. Für die Berechnung könnte dann auch ein Taschenrechner in die Anwendung implementiert werden. Dadurch würde das Lernspiel ebenfalls an Umfang gewinnen.

PROBLEMSTELLUNG

Natürlich gab es, wie bei allen Projekten, auch hier einige Probleme bei der Entwicklung. Die meisten dieser Probleme waren relativ leicht zu beheben, bei anderen war es allerdings schwerer. Beispielsweise war geplant, von allen Objekten das Volumen und die Oberfläche als Ergebnis zu verlangen, was jedoch nur bei den simpleren Objekten realisierbar war. Bei einem umfangreichen Model, das aus vielen kleinen Objekten besteht, gab es das Problem, dass wenn es galt die Oberfläche zu berechnen, dies nicht ganz möglich war. Bei diesen geometrischen Figuren gibt es das Hindernis, dass es einen kleinen Teil der Fläche gibt, der von einem anderen Objekt verdeckt wird. (siehe Abb.14) Auf der Abbildung werden zum Beispiel gleich mehrere Teile der Fläche von anderen Objekten bedeckt, was es überaus schwierig macht hier die genaue Oberfläche zu berechnen. Dieser kleine Teil zählt nicht zur gesamten Oberfläche und somit gilt es momentan nur, das Volumen der zusammengesetzten Modelle zu berechnen.

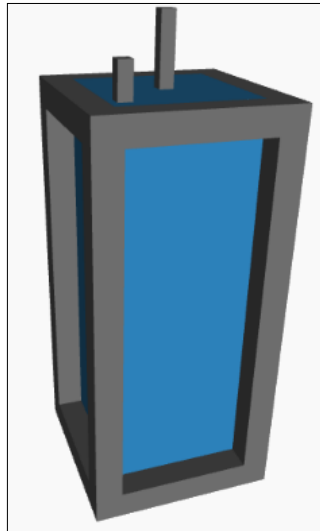


Abbildung 14: Verdeckter Teil der Oberfläche

VERGLEICHBARE PROJEKTE

Es gibt sicherlich einige andere Anwendungen im Internet, die sich auf Geometrien beziehen. Deswegen werden an dieser Stelle einige Vergleiche zwischen dieser und anderen Webanwendungen aufgeführt. Bei unserem Spiel geht es primär um die Beantwortung der Fragen und Einprägen der Formeln zu den einzelnen geometrischen Figuren. Bei anderen Projekten die wir gefunden haben, ging es hauptsächlich darum, dass der Spieler die einzelnen Geometrien kennenlernt.

Beispielprojekt 1

Das erste Projekt, welches wir betrachten wollen, behandelt primär, wie bereits beschrieben, das Kennenlernen der Geometrien. (Quelle: <http://www.math-play.com/3d-shapes.html>) Spielprinzip hierbei ist es, die einzelnen Namen der Geometrien, den Figuren zuzuweisen. (siehe Abb.15) Das geschieht, indem der Anwender den Namen mit der Maus einfach auf die Figur zieht. Dieses Projekt basiert allerdings nicht auf X3DOM und ist dadurch auch relativ unspektakulär, was die Interaktion mit der Anwendung angeht. Ebenfalls ist es dort schwerer ein Gefühl für 3-Dimensionale Objekte zu bekommen, da einzelne Figuren in *2D-Sprites* abgebildet werden. Nehmen wir nun unser Projekt als Vergleich, so wird offensichtlich, dass bei unserem mehr Interaktion mit dem Spieler gefordert wird, als ein simples *Drag&Drop*. Allerdings ist die Grundidee im Prinzip gleich, da auch diese Beispielanwendung das Wissen über Geometrien vermitteln will.

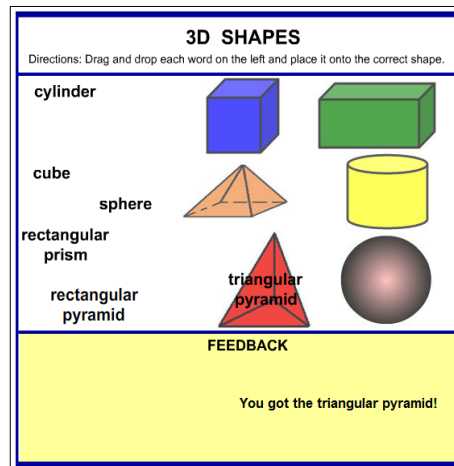


Abbildung 15: Projektbeispiel 1

Beispielprojekt 2

Das nächste Beispiel ist in dem Sinne kein Mathelernspiel. Allerdings sollte auch ein Vergleich zu einer X3DOM-Anwendung stattfinden. Aus diesem Grund wurde dieses Beispiel gewählt. (Quelle: <http://math.nist.gov/~Sressler/deckjs/introduction/x3domDeck.html#slide-6>) Dabei handelt es sich lediglich um eine kleine Präsentation in der Geometrischen Figuren im 3-Dimensionalen Raum dargestellt werden und diese mit der Maus drehbar sind. (siehe Abb.16) Wie erwähnt handelt es sich hierbei nicht um ein Spiel.

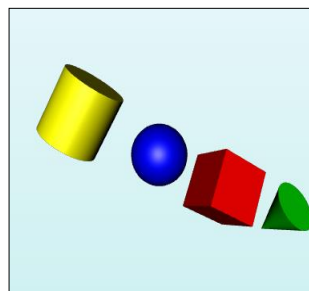


Abbildung 16: Projektbeispiel 2

FAZIT

Kommen wir nun zum Fazit. Anfänglich waren wir skeptisch, bezüglich dieser Idee. Doch im Laufe der Implementierung machte es auf gewisse Art sogar Spaß. Die Idee, räumliches Denken und geometrische Mathematik in einem Spiel zu vereinen, ist relativ gut gelungen. Zwar gibt es noch Funktionalitäten, die implementiert werden können, wie unter Erweiterbarkeit erwähnt, doch das Grundprinzip wurde umgesetzt. Während der Entwicklung gab es auch Probleme, die allerdings auf gutem Wege gelöst werden konnten. Somit haben wir während des Projekts Einiges an Erfahrung und Wissen gewinnen können.

Schlussendlich lässt sich sagen das sich mit X3DOM gut arbeiten lässt und 3-Dimensionale Objekte relativ leicht einzubinden sind. Für zukünftige *3D Browser Anwendungen*, würden wir uns wieder dafür entscheiden.