



universität  
wien

# DIPLOMARBEIT

Titel der Diplomarbeit

„Spieledidaktik – Einstieg in die Programmierung  
mit Hilfe von Spielen und Game-Design“

Verfasser

Andreas Braun

angestrebter akademischer Grad

Magister der Naturwissenschaft (Mag.rer.nat.)

Wien, 2013

Studienkennzahl lt. Studienblatt:

A 190 299 884

Studienrichtung lt. Studienblatt:

Lehramtsstudium, UF Psychologie und Philosophie,  
UF Informatik und Informatikmanagement

Betreuer:

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Futschek



*Gewidmet meinen Eltern,  
Hartmann und Hildegard Braun*

# INHALTSVERZEICHNIS

|          |                                               |           |
|----------|-----------------------------------------------|-----------|
| <b>1</b> | <b>Einleitung &amp; Problemstellung.....</b>  | <b>1</b>  |
| 1.1      | Einleitung.....                               | 1         |
| 1.2      | Methodik und Fragestellung.....               | 4         |
| <b>2</b> | <b>Konstruktivistische Didaktik.....</b>      | <b>5</b>  |
| 2.1      | Warum Konstruktivismus?.....                  | 5         |
| 2.2      | Grundlagen.....                               | 7         |
| 2.3      | Postulate von Kersten Reich.....              | 13        |
| 2.3.1    | Das Zusammenspiel der drei Grundbegriffe..... | 17        |
| <b>3</b> | <b>Multiplayer Classroom.....</b>             | <b>19</b> |
| 3.1      | „World of Warcraft“.....                      | 19        |
| 3.1.1    | Spielablauf.....                              | 20        |
| 3.1.2    | Kommunikation und Sozialisation im Spiel..... | 22        |
| 3.1.3    | Nur ein Spiel.....                            | 23        |
| 3.1.4    | Erfolgsrezept?.....                           | 24        |
| 3.2      | Das Unterrichtskonzept von Lee Sheldon.....   | 26        |
| 3.2.1    | Der Lehrplan.....                             | 27        |
| 3.3      | Feedback.....                                 | 31        |
| 3.4      | Reflexion & Anmerkungen.....                  | 34        |
| 3.5      | Didaktische Analyse.....                      | 39        |
| <b>4</b> | <b>Didaktische Spiele.....</b>                | <b>43</b> |
| 4.1      | Abwägung der Alternativen.....                | 43        |
| 4.2      | Kurzvorstellung.....                          | 46        |
| 4.3      | Ausarbeitungen.....                           | 49        |
| 4.3.1    | BYOB & Heapsort.....                          | 49        |
| 4.4      | Didaktische Spiele.....                       | 52        |
| 4.4.1    | Catch me if you can.....                      | 55        |
| 4.4.2    | Circuit.....                                  | 60        |
| 4.4.3    | Donkey Kong.....                              | 63        |
| 4.4.4    | PacMan.....                                   | 72        |
| 4.4.4.1  | PacMan easy.....                              | 73        |

|                                       |            |
|---------------------------------------|------------|
| 4.4.4.2 PacMan medium.....            | 77         |
| 4.4.4.3 PacMan hard.....              | 82         |
| 4.4.4.4 PacMan master.....            | 84         |
| 4.5 Didaktische Reflexion.....        | 87         |
| <b>5 Theorie &amp; Praxis.....</b>    | <b>88</b>  |
| 5.1 Konstruktivistische Didaktik..... | 89         |
| 5.2 Multiplayer Classroom.....        | 91         |
| <b>6 Conclusio.....</b>               | <b>94</b>  |
| <b>7 Abstract (deutsch).....</b>      | <b>96</b>  |
| <b>8 Abstract (english).....</b>      | <b>97</b>  |
| <b>9 Literaturverzeichnis.....</b>    | <b>98</b>  |
| <b>10 Lebenslauf.....</b>             | <b>102</b> |



# **1 EINLEITUNG & PROBLEMSTELLUNG**

## **1.1 Einleitung**

Als angehender Lehrer ist die Kompetenz- und Wissensvermittlung für mich ein sehr zentrales Thema meiner Ausbildung, vor allem, da es in nicht allzu ferner Zukunft die täglich zentralste Aufgabe meiner Arbeit darstellen wird. Daher bin ich natürlich sehr an den verschiedensten Methoden und Konzepten interessiert, die diese Vermittlung erleichtern, die in der Lage sind Komplexität zu reduzieren und ein Umfeld schaffen können, das möglichst lernförderlich wirkt. Die Reduktion der abstrakten Komponenten von Programmiersprachen orientiert sich hierbei an Theorien zur didaktischen Reduktion, wie sie etwa Martin Lehner definierte: „Didaktische Reduktion findet immer dann statt, wenn umfangreiche und komplexe Sachverhalte aufbereitet werden, um sie für die Lernenden überschaubar und begreifbar zu machen.“ (Lehner, 2012, S. 9) Der Fokus liegt dabei vor allem in der Auffindung von essentiellen Themenbereichen, die bei einem Einstieg in die Programmierung vor allem im Bereich des Erlernens von Algorithmen und der Aneignung von algorithmischen Denken angesiedelt ist, da dieser konzeptuelle Ansatz meiner Meinung nach die Quintessenz bildet, welche dann in weiterer Folge auf andere Programmiersprachen umgelegt werden kann. Die sprachspezifischen Eigenheiten in der Schreibweise sowie daraus resultierende Rechtschreibfehler und für ungeübte Anwender nichtssagende Fehlermeldungen könnten bereits mit der Wahl einer geeigneten Lernumgebung im Wesentlichen eliminiert oder zumindest über weite Strecken auf ein Minimum reduziert werden.

Im Laufe meines Studiums sind mir zahlreiche Möglichkeiten untergekommen, wie dies bewerkstelligt werden könnte, wobei ich eine Vielzahl davon selbst erfahren und probieren durfte. Im Rahmen der schulpraktischen Ausbildung und der fachbezogenen Praktika ist mir durch die Reflexionen und Gespräche mit den Betreuungslehrerinnen und -lehrern stets bewusster geworden, dass es für einen positiven Lernerfolg sehr zweckdienlich ist, wenn die Lernenden sich aktiv mit einem Thema auseinandersetzen können, was im Unterrichtsfach Informatik durch die Arbeit am Computer ja fast schon unvermeidlich, aber auch in anderen Fächern

möglich ist. Dabei wurde mir unter anderem auch bewusst vor Augen geführt, dass es eine besondere Kunst der Didaktik zu sein scheint, die Neugier der Lernenden zu wecken, was wesentlich einfacher klingt, als es dann tatsächlich ist, denn heutzutage konkurrieren Lehrerinnen und Lehrer immer stärker mit der bunten Vielfalt von elektronischen und multimedialen Unterhaltungsmaschinerien, was aber wiederum Überlegungen in mir weckte, ob nicht gerade dieser große Verführer und eigentliche Konkurrent etwas zur Didaktik beitragen könnte.

In einem Seminar an der Universität Wien zum Thema „didaktisches Design“ im Sommersemester 2010 stieß ich dann das erste Mal auf Lee Sheldon, damals Professor an der Indiana University, der seine Studentinnen und Studenten nach einer für didaktische Zwecke adaptierte Form des Spieldesigns von World of Warcraft, einem Computerspiel (genauer: MMORPG<sup>1</sup>) der Firma Blizzard Entertainment, unterrichtete. Auf den ersten Blick wirkte dieser Unterrichtsstil auf mich etwas befremdlich, doch auch durchaus faszinierend, denn anstatt Computerspiele als eine bloße Ablenkung zu sehen, schien Sheldon einen Weg gefunden zu haben einen Kontrahenten der Lehrenden in ihren Dienst zu nehmen sowie darin enthaltene Mechaniken und Konzepte, die scheinbar einen motivierenden Einfluss ausüben können, ausfindig gemacht zu haben. Gerade bei dieser Gratwanderung zwischen Unterhaltung und Lernen im Bereich der Didaktik ist es mir wichtig, dass meine Überlegungen und Ausarbeitungen auf einer didaktischen Theorie basieren, da hierdurch bereits Maßstäbe und fundierte wissenschaftliche Konzepte existieren, auf die zum Zwecke der Reflexion und Explikation der Funktionsweisen zurück gegriffen werden kann. Dadurch soll weitestgehend sicher gestellt werden, dass die entwickelten und reflektierten Modelle sich schon grundsätzlich an wissenschaftlichen Standards und nicht am reinen Vergnügen orientieren.

Im Rahmen der Methoden sollen die Möglichkeiten und Grenzen von spielerischen und intuitiven Zugängen vor allem für ProgrammieranfängerInnen ausgearbeitet werden, wobei in diesem Abschnitt BYOB (engl.: Build your own blocks) untersucht wird, das speziell für Neueinsteiger in der Programmierung entworfen wurde und mit dessen Hilfe es möglich sein könnte, einen leichteren und einfacheren Zugang zu den

---

<sup>1</sup> MMORPG: Massively Multiplayer Online Role-Playing Game

siehe <http://eu.battle.net/wow/de/game/guide/> [zuletzt abgerufen am 08. November 2012]

eher abstrakten Strukturen von Programmiersprachen zu bieten. Die wohl wesentlichsten Hürden für Anfängerinnen und Anfänger im Programmieren sind (vgl. Futschek, 2012):

1. Syntax der Programmiersprache
2. Fehlermeldungen

Grafische Lernumgebungen könnten genau hier Abhilfe schaffen, da weniger die syntaktische Richtigkeit, die häufig unverständliche Fehlermeldungen liefert und dadurch oftmals ein nicht zu unterschätzendes Hindernis im Erlernen darstellt, sondern vielmehr der algorithmische Ablauf im Vordergrund steht. „At the beginning of learning algorithmic thinking learners often have problems with this abstract type of thinking and with the abstraction in a programming environment.“ (Futschek & Moschitz, 2011, S. 155) Zwei weitere Ursachen für Probleme beim Einstieg in die Programmierung wären hiermit einerseits das abstrakte, algorithmische Denken, das vielleicht mit klaren Aufgabenstellungen und visuellen Repräsentationen leichter erlernt werden kann, wodurch Spiele diese Hürde eventuell vermindern könnten, da diese die Aufgabe meist visuell aufbereiten können. Andererseits, und hier besteht erneut die Möglichkeit, dass sich grafische Lernprogramme als sehr nützlich erweisen können, erschweren die abstrakten Umgebungen und deren Handhabung noch zusätzlich eine reibungslose Einführung.

Die zentralere Rolle, die Algorithmen bei diesen Betrachtungen einnehmen, ist auch bei den aktuellen bildungspolitischen Diskussionen und der zunehmenden Fokussierung auf das Erlernen und Verstehen von Konzepten ein sehr interessanter Pluspunkt. In diesem Rahmen werden auch mögliche Projekte vorgestellt und entwickelt, die für eine Einführung in das Programmieren verwendet werden können, denn die Untersuchungen der Methoden sollen ein Pool von möglichen didaktischen Spielen für ProgrammieranfängerInnen ergeben, die den Spagat zwischen professionellem didaktischen Einsatz und intuitivem Zugang schaffen. Gerade durch Spiele könnte eine Möglichkeit existieren, um den Grad an Abstraktion zu reduzieren, denn die Lernenden können hierdurch die Verbindung von theoretischem, abstrahiertem Modell und dessen realer Anwendung herstellen (vgl. Shaffer, Squire, Halverson & Gee, 2004, S. 4 f.). Die Kriterien, welche die Methoden dabei erfüllen sollen ist einerseits eine möglichst signifikante Reduktion der Komplexität für ProgrammieranfängerInnen und andererseits eine Förderung der Motivation.

Letztendlich möchte ich im Rahmen meiner Ausarbeitungen und Untersuchungen einige didaktische Methoden und Konzepte präsentieren, die eine möglichst lernförderliche Symbiose aus Didaktik und Spieldesign ergeben sollen. Dadurch möchte ich auch Möglichkeiten aufzeigen, die dazu dienen können, dass die Komplexität vor allem für jene Lernenden zu reduziert wird, die noch keine Erfahrung im Umgang mit Programmiersprachen haben. Die meist spielerisch angehauchten und durch didaktisch-pädagogische Überlegungen unterstützten Methoden sollen durch zusätzliche Anreize und intuitive Zugänge die größten Probleme beim Erlernen von Algorithmen, Datenstrukturen und in weiterer Hinsicht natürlich von ganzen Programmiersprachen ausmerzen und so die Effektivität des Unterrichts erhöhen.

## **1.2 Methodik und Fragestellung**

Mein methodisches Vorgehen wurde von mir in drei Blöcke unterteilt. Als Basis ist es mir ein Anliegen, eine didaktische Theorie zu präsentieren, die mir als Fundament meiner Ausarbeitungen und zugleich als Werkzeug für eine kritische Analyse und Reflexion im Hinblick auf das von Lee Sheldon ausgearbeitete Konzept dient. Den zweiten Block bilden die Untersuchungen, ob und wie Spiele bzw. deren Konzepte und Designs im Sinne der Organisation von Unterricht als dessen Rahmen verwendet werden können, wobei gerade hier die didaktische Theorie zur Analyse und kritischen Betrachtung verwendet werden soll, um die Funktionsweisen und Mechanismen zu extrahieren und zu prüfen. Im dritten Schritt möchte ich einige Spiele präsentieren, die den Einstieg in die Programmierung durch eine Reduktion der Komplexität erleichtern sollen.

Zusammenfassend möchte ich meine Fragestellung nun wie folgt formulieren:

**Bieten Spiele und deren Konzepte eine günstige Möglichkeit zur Reduktion der Komplexität und Förderung der Motivation für einen Einstieg in die Programmierung?**

## **2 KONSTRUKTIVISTISCHE DIDAKTIK**

### **2.1 Warum Konstruktivismus?**

Die Wahl des didaktischen Modells ist für den wissenschaftlichen Diskurs im Rahmen einer Begründung für das eigene Vorgehen und dessen Reflexion ein grundlegendes, doch bei der Fülle und Divergenz an Ansätzen durchwegs kein einfaches Unterfangen. Der Großteil didaktischer Theorien scheint dabei selten eine rezepthafte oder deterministische Lösung, sondern vielmehr allgemeine Vorstellungen mit Bezug auf Unterricht, Lehren und Lernen zu präsentieren, die sich oftmals auf pädagogische, psychologische oder philosophische Ansätze bzw. Überlegungen stützen (vgl. Tulodziecki, Herzig und Blömeke, 2009, S. 211 ff). Der spezifische Fokus jeder Theorie bestimmt ihren Betrachtungswinkel und mithin auch, welche Bereiche von Unterricht genauer unter die Lupe genommen werden. Somit ergibt es sich, dass manche Theorien ihr Augenmerk auf lerntheoretischen Erkenntnissen haben, andere nehmen eine mehr bildungstheoretische Position ein und wieder andere haben erkenntnistheoretische, dialektische oder kommunikative Annahmen als Grundlage.

Ein in den letzten Jahren sehr populäres Modell bietet Kersten Reich, der im Bereich konstruktivistischer Theorien angesiedelt ist. Einer der Vorzüge seines Ansatzes ist meiner Meinung nach, dass wissenschaftliche Erkenntnisse aus anderen Bereichen hierin sehr gut Platz finden können, wie etwa aus lerntheoretischen (kognitive Erkenntnisse aus der Psychologie) oder projektorientierten Modellen (vgl. hierzu Tulodziecki et al, 2009, S. 15 ff). Durch den Bezug zu erkenntnistheoretischen Thesen setzt sein Konzept direkt beim Erkennen der Welt an und somit auch noch vor dem Lernen, das ein Erkennen der Welt immer schon vorausgesetzt hat, wodurch der Fokus der Betrachtungen und Überlegungen etwas breiter gefasst zu sein scheint und mehr Aspekte des menschlichen Handelns als für den Unterricht bedeutungsvoll betrachtet, als es bei einigen anderen Theorien der Fall ist. Die konstruktivistische Sicht auf Didaktik und Unterricht schließt ebenso Betrachtungen von Lehren und Lernen unter dem Einfluss von sozialen, emotionalen, kommunikativen und kul-

turellen Faktoren mit ein, wodurch ebenfalls Untersuchungen von zwischenmenschlichen Beziehungen sowie Kommunikation mit einfließen können.

Des Weiteren konnte ich im Rahmen meiner Ausbildung mehrmals die Erfahrung machen, dass eine aktive Beteiligung der Lernenden am Unterricht, die in der konstruktivistischen Didaktik ein zentraler Bestandteil ist, bei einer guten Unterrichtsvorbereitung und -planung durchaus effizient und förderlich sein kann. Allerdings ist das konstruktivistische Konzept natürlich nicht unumstritten, da die Ergebnisse mancher Studien und Analysen nicht sehr positiv ausfallen: „Manche Untersuchungen belegen sogar Leistungsverlechterungen, wenn zuvor nach konstruktivistischen Prinzipien gelernt wurde.“ (Helmke, 2010, S. 70) Hinzu kommen nach Helmke (2010) auch Bedenken bezüglich der Förderung bei Lernschwächen und der zeitliche Mehraufwand für Vor- bzw. Nachbereitung des Unterrichts. Auch die Ableitung didaktischer Prinzipien aus der Psychologie ist nicht unangezweifelt: „Meines Erachtens lassen sich nicht einfach aus der Kenntnis spezifischer innerpsychischer Vorgänge didaktische Prinzipien und Methoden ableiten.“ (Schüßler, 2005, S. 89) Hinzu kommen freilich auch noch erkenntnistheoretische Grundsatzdiskussionen das Thema Wahrheit und deren Beliebigkeit betreffend, die oftmals in den Bereich wissenschaftstheoretischer Untersuchungen hinein reichen, wobei ich folgendes Zitat zur Verteidigung konstruktivistischer Didaktik am Beispiel Schrödingers Katze<sup>2</sup> für sehr aufschlussreich für seine Sichtweise halte: „Es geht Schrödinger als Naturwissenschaftler ja nicht um die unsinnige Behauptung, dass die Katze in einer tatsächlich luftdichten Kiste überleben konnte, sondern allein um das Phänomen, dass Wirklichkeitskonstruktionen in den Wissenschaften immer von einem Beobachter abhängen, und damit davon, was dieser sehen kann oder will.“ (Reich, 2002, S. 92)

Für den Informatikunterricht halte ich den konstruktivistischen Ansatz dennoch für prinzipiell sehr gut geeignet, da sich der Großteil der Themen und in weiterer Folge die Aufgabenstellungen meist durch selbstständiges Arbeiten der Lernenden

---

<sup>2</sup> Wer das Beispiel nicht kennt, findet unter nachfolgender URL eine kurze Erläuterung von Georg Huhs: [http://physik.htu.tugraz.at/w/images/1/15/Vortrag\\_physik.pdf](http://physik.htu.tugraz.at/w/images/1/15/Vortrag_physik.pdf) [zuletzt abgerufen am 09. Dezember 2012]

auszeichnen, das vielleicht gerade durch den konstruktivistischen Ansatz noch gefördert und weiter ausgebaut werden kann. Der Grund hierfür ist meiner Meinung nach, dass es in der Informatik wesentlich immanenter und zentraler um die Anwendbarkeit des erworbenen Wissens geht, als es in anderen Disziplinen der Fall ist, wodurch die Vermittlung von Kompetenzwissen und mithin konstruktivistische Ansätze hier einen höheren Grad an Effizienz aufweisen könnten, als in anderen Fächern. Hinzu kommt, dass die Thesen der konstruktivistischen Theorien und mithin das Momentum der Konstruktion sich gut mit Spielen vereinen lassen, da das Phänomen des Eintauchens sowie die Wahrnehmung der virtuellen Welt dadurch verständlich und begreifbar gemacht werden können: „When it comes to understanding the experience of computer games from a player-centered approach, constructivist theory provides viable perspectives even if it is limited to its use as a low-range theory.“ (König, 2011, S. 38) Allerdings möchte ich die didaktischen Theorien freilich nicht als strenge Richtlinien, sondern in der Form einer didaktischen Begründung verwenden, die Reflexionen und Analysen auf einer wissenschaftlich fundierten Basis ermöglichen sollen und die gegebenenfalls auch durch andere didaktische Strömungen erweitert und ergänzt wird. Um es mit den Worten von Reich zu beschließen: „Denn Didaktik ist eine Perspektive auf Lerner“ (Reich, 2008, S. 51).

## **2.2 Grundlagen**

Als theoretische Basis diene mir Kersten Reich mit seinen Theorien einer interaktionistisch-konstruktivistischen Pädagogik und konstruktivistischen Didaktik, der den Fokus sehr zentral auf Eigenständigkeit und Selbsttätigkeit der Lernenden legt und pädagogische Klassiker wie etwa Jean Piaget, Paul Watzlawick oder auch Michel Foucault zu einem System vereint, bei dem sehr zentral die Lernenden und weniger die inhaltlichen Schwerpunkte im Vordergrund stehen. Den Grundgedanken konstruktivistischer Theorien bringt er meiner Meinung nach mit folgenden Worten sehr treffend auf den Punkt: „Alle sogenannten Welten, Dinge, sprachlich bezeichneten Objekte, alles, was wir als wirklich aussagen und dem wir Bedeutung, Werte und was auch immer zuschreiben, sind menschliche Konstruktionen von Wirklichkeiten“ (Reich, 2008, S.79). Diese Konstruktionen bilden eine Art Schnitt-

stelle zwischen dem menschlichen Bewusstsein und der Wirklichkeit, sie ermöglichen es uns, dass wir die äußere Welt in uns nachbilden und sie dadurch schlussendlich erkennen können: „Verstehen freilich können wir im Hinblick auf die Erkenntnis nur das, was wir konstruieren“ (Wallner, 1992, S. 97). Schon die Betrachtung der menschlichen Sinneswahrnehmung gibt erkenntnis-theoretische Hinweise dazu, dass wir die Welt in uns als Konstruktion abbilden und nicht eine „wirkliche“ Welt wahrnehmen, da die elektrischen Impulse der Sinne erst im menschlichen Gehirn verarbeitet werden. Dadurch zeigt sich unter anderem, dass wir nicht in die Welt hinein sehen, sondern dass sie auf uns einwirkt und wir sie erst durch die kognitiven Leistungen unseres Gehirns erkennen können. Sehr interessante Experimente hierzu wurden zum Beispiel an der Universität Innsbruck unternommen, bei denen die Versuchsperson eine sogenannte „Umkehrbrille“<sup>3</sup> für einen Zeitraum von 124 Tagen trug, welche die Welt gespiegelt darstellte. Das Ergebnis war, dass sich die Wahrnehmung an die neue Situation anzupassen begann und das Erleben der Welt nach einiger Zeit wie gewohnt war, was verdeutlicht, wie weitreichend der Einfluss des Gehirns auf unsere Wahrnehmung der äußeren Wirklichkeit ist. Abgesehen von den Sinnen unterliegt unsere Erkenntnis, Wahrnehmung und weiters auch unsere Konstruktion sozio-kulturellen Einflüssen, wie moralischen Richtlinien und menschlichen Gesetzen, die sich wiederum ihrerseits als Konstruktionen des menschlichen Geistes auf die Wahrnehmung der Welt auswirken. Die kulturellen Einflüsse sind für Reich daher besonders von Bedeutung: „Reich versteht Konstruktivismus als kulturelle Variante, in der – entsprechend den erkenntnisphilosophischen Grundannahmen – die vielfältigen Weltauslegungen konstruierte 'Wahrheiten' darstellen, die uns teilweise wie Tatsachen erscheinen, letztlich aber auf Verständigungsleistungen einer Gesellschaft basieren und als solche kulturelle Anschlussfähigkeit sichern.“ (Tulodziecki et al, 2009, S. 250)

Diese lern- und erkenntnistheoretischen Ansätze haben nun auch unweigerlich Auswirkungen auf didaktische Überlegungen, da Erkenntnis und Lernen sehr nahe an Erfahrung rücken. Wenn wir nun auf der Suche nach einer didaktischen Umsetzung sind, die sich möglichst produktiv an den Thesen der konstruktivistischen Didaktik

---

<sup>3</sup> Siehe hierzu: <http://www.uibk.ac.at/psychologie/institut/kohler.html>

orientiert, so treten neben dem Ziel der reinen Wissensreproduktion auch noch einige weitere wichtige Elemente auf. Als Kinder lernen wir zum Beispiel, wie auch andere Säugetiere, von Natur aus durch die Interaktion mit unserer Umwelt, wir erforschen und erkunden sie, interagieren noch sehr explorativ und spielerisch mit ihr, wir untersuchen sie und experimentieren mit unserem Umfeld. Dieses frühe Lernen besteht aus zwei wesentlichen Komponenten, nämlich Beobachtung und Imitation, wobei gerade Zweiteres im Zusammenhang mit der Theorie Kersten Reichs und den didaktischen Experimenten von Lee Sheldon von großem Interesse ist, da Imitation eine Nachahmung von Verhalten ist, was an eine evolutionäre Form des Rollenspiels erinnert, wodurch auch in weiterer Folge die Einbettung von Unterricht in ein derartiges Setting eigentlich naheliegender ist, als es am ersten Blick zu sein scheint. Imitation ist daher ein aktiver Prozess und wesentlich weniger passiv, als reine Reproduktion und Wiedergabe von Inhalten.

Die Erkundung im Zusammenhang mit der Interaktion und der damit verbundenen aktiven Auseinandersetzung mit unserer Umwelt, verweist auf einen weiteren, sehr zentralen Punkt der konstruktivistischen Didaktik, nämlich Selbstbestimmung. Exploratives Lernen ist im Wesentlichen ein sehr stark von uns selbst bestimmter, intrinsisch motivierter Prozess, bei dem das Individuum die Kontrolle über das eigene Lernen übernimmt. Schon durch die Hervorhebung der Bedeutung von Interaktion im Lernprozess kristallisiert sich heraus, dass auch das Individuum selbst mehr an Beteiligung und Selbstbestimmung im Lernen erfahren sollte. Den Grund für die Wichtigkeit von Selbstbestimmung und Selbsttätigkeit erklärt Reich wie folgt: „Das Lernen aus innerer Einsicht, aus Selbstzwängen heraus, ist viel effektiver als ein Lernen, das auf Fremdzwänge vertraut.“ (Reich, 2010, S. 160) Bestrafung, als ein Paradebeispiel für Fremdzwang, sei es aufgrund von Druck durch schlechte Noten oder auch sehr subtil bereits durch Punkteabzug für Fehler, kann im Extremfall sogar äußerst unerwünschte Folgen haben, denn dadurch werden möglicherweise "starke Emotionen (Angst) nicht nur auf das bestrafte Verhalten, sondern auf die ganze Lernsituation übertragen" (Herkner, 2001, S.57), was auch die Untersuchungen von Helmke bestätigen (2010, S. 58). Da durch diese Betrachtungen Bestrafungen und andere Formen extrinsischer Motivation, aufgrund der Ermangelung an Förderung individuellen Selbstzwanges und intrinsischer Motivation, etwas an Bedeutung

verlieren, scheint es umso wichtiger, didaktische Mittel und Wege zu finden, um die Eigenmotivation zu fördern, wobei natürlich nicht gänzlich auf extrinsische Anreize verzichtet werden kann oder soll (vgl. hierzu Helmke, 2010, S. 216 f.). Für Lehrende wird es aber bedeutsamer eine Form der Organisation von Unterricht zu finden, bei der sie nicht mehr bloß Wissen in Form von Vorträgen zu übermitteln versuchen, sondern einen didaktischen Rahmen verwenden, der die Selbsttätigkeit ihrer Lernenden fördert, um dadurch eine Annäherung und Anpassung an eine natürlichere und nachhaltigere Lehrstruktur zu erreichen. Kersten Reich bringt diesen Gedanken, der bereits zu Zeiten der Aufklärung aufkam, im Rahmen der Pädagogik sehr gut auf den Punkt: „Die ganze Aufklärung des Pädagogen scheint darin zu bestehen, dass er das Kind zu seinem natürlichen Eigenrecht, zu einer in ihm liegenden Dynamik von Entwicklung kommen lässt, die nur behutsam gefördert, aufmerksam gelenkt und in ihre natürlichen Wege geleitet werden muss“ (Reich, 2010, S.218).

Es liegt hier eine völlig andere theoretische Basis zugrunde, die kaum noch etwas mit dem überspitzten Modell des Nürnberger Trichter gemeinsam hat, denn das Wissen ist in diesem konstruktivistischen Rahmen nun nichts mehr, das in die Köpfe der Lernenden eingeflößt oder eingetrichtert wird, die dabei nur noch sehr passiv anwesend sein müssen, sondern der Prozess wird wesentlich aktiver. Es wird nun durch den lerntheoretischen Fokus auf handlungsorientiertes Lernen viel mehr die Bedeutung der Interaktion und der Selbsttätigkeit in den Vordergrund gestellt, wodurch die Lernenden im Unterricht auch wesentlich aktiver gefordert sind und sein sollen. Der Lehrende ist nun auch nicht mehr bloß der Vermittler von Wissen, sondern gleitet mehr in eine leitende und organisierende Rolle hinüber, bleibt aber dennoch stets Inhaber des Wissens, allerdings mehr in einer dirigierenden, fördernden und lenkenden Position.

Mit dem Blick bereits Richtung Informatik kommt mir aus den Erfahrungen während meines Studiums zum Beispiel „Lego Mindstorms NXT“ als eine gestalterische Möglichkeit für den Informatik Unterricht in den Sinn, denn das System ermöglicht es informatische Kompetenz in der Programmierung zu erlernen und zwar auf eine Art und Weise, die viele Hürden für die erste Begegnung mit Programmierung

ausblendet, so etwa die für viele Neueinsteiger unverständlichen Fehlermeldungen oder eine Vielzahl von syntaktischen Regeln, die für einen Erfolg oft sehr hinderlich sein können. Noch dazu bietet der spielerische Zugang viele Anreize zugunsten einer Steigerung der Selbsttätigkeit. Wie ich selbst im Rahmen eines Seminars beobachten durfte, war es mit Hilfe von Mindstorms sehr gut möglich, die spielerische Neugier zu entfachen, die von einem sehr explorativen Charakter gezeichnet war, sowie die daraus entstandene Motivation mittels Aufgaben und Wettbewerben zu lenken, um einen Lernerfolg sicher zu stellen. Die Beschäftigung mit den Robotern fand augenblicklich nach dem Auspacken statt und entstand spontan, es waren dafür also kaum weitere extrinsische Motive notwendig. In dieser Situation erschien mir der äußerliche Druck sehr gering, da er nur zur Ausrichtung des Spieltriebs auf die zu erreichenden Ziele nötig war, wobei der Lösungsweg natürlich nicht vorgegeben wurde, denn diesen galt es in der Arbeitsgruppe selbst zu finden. Dadurch war der Lernprozess weitestgehend von den Schülern und Schülerinnen bestimmt und es war hierfür ebenfalls auch eigenständiges Arbeiten erforderlich.

Generell ist die Informatik in der glücklichen Rolle, dass sie bereits ein Werkzeug mit sich bringt, das die Möglichkeit eines sehr interaktionistischen Zugangs bietet, nämlich den PC. Der Großteil der Themenbereiche in der Informatik, von der Allgemeinbildung bis hin zum Studium, kann mit dem Hilfsmittel Computer interaktiv gestaltet werden und fordert somit bereits eine Form der Interaktion. Allerdings ist beim Einsatz von Computern in didaktischer Sicht darauf zu achten, dass sie nicht als Selbstzweck verwendet werden, sondern dass sie eine für das Thema geeignete Plattform darstellen. „Wenn Schüler in Zukunft immer mehr mit interaktiven Lernprogrammen – z.B. Vokabeltrainern – am PC arbeiten werden, dann greift dies konstruktivistisch gesehen zu kurz.“ (Reich, 2010, S. 230) Die reine Interaktion mit dem PC, wie hier am Beispiel des Vokabeltrainers, reicht also noch nicht aus, um dem konstruktivistischen Grundgedanken Genüge zu tun, da dem Lernen, wie am Beispiel des Vokabeltrainers zu sehen ist, nach wie vor eine reproduktive und rezeptive Charakteristik inne wohnt, bei der lediglich das Medium der Wissenstransport ausgetauscht wurde. Eine Methode im Sinne Kersten Reichs könnte zum Beispiel ein als Rollenspiel aufgeführter Dialog sein, in dem die neu zu erlernenden Vokabel eingebettet sind, wobei die Ausarbeitung der Unterhaltung den

Lernenden vorbehalten bleibt. Dadurch könnte erzielt werden, dass die synaptische Vernetzung breiter gefächert ist, wie bspw. Helmke (2010, S. 62 f.) bei der horizontalen Informationsverarbeitung beschreibt, und somit geistige Brücken (Eselsbrücken) als Erinnerungshilfen entstehen können. Nach dem kurzen Ausflug in den Bereich des Gedächtnistrainings aber wieder zurück zum Computer als didaktisches Mittel und speziell dem Einsatz von Internet und Hypermedien als Quelle für Recherchen, die nach einer Studie von Ingeborg Schüßler, mit dem Titel „Lernwirkungen neuer Lernformen“, nämlich durchaus auch ein zweischneidiges Schwert sein können, da hierfür das Vorwissen oft ein hinreichendes Kriterium für dessen erfolgreichen Einsatz zu sein scheint, und die Fülle an Information im World Wide Web im Zusammenspiel mit einer unzureichenden Aufbereitung der Anbieter oder mangelnder Vorbereitung der Lehrenden des öfteren zu einer Desorientierung der Lernenden führen kann (vgl. Schüßler, 2004, S.48 ff). Der Grund hierfür liegt in der beschränkten Aufnahmekapazität des Gehirns, quasi dem Cache-Speicher, der nur wenig Information gleichzeitig erinnern kann (vgl. Lehner, 2012, S. 24 f.). Auch in diesem Beispiel erweist sich der PC als wenig förderlich, wenn dieser ein didaktischer Selbstzweck ist und die Einbettung unzureichend geplant und durchdacht ist. Der eigentliche konstruktivistische Gedanke erfährt hier im Bereich der leitenden, strukturierenden Rolle der Lehrenden scheinbar auch oftmals eine mangelhafte Umsetzung, vor allem im Bereich der Vor- und Aufbereitung der zu erarbeitenden oder recherchierenden Inhalte.

Studien hierzu, mit dem Versuch empirischer Evaluationen von konstruktivistischen Methoden, wurden zum Beispiel 2004 von Dr. Ingeborg Schüßler veröffentlicht und "konnten zwar in vielen Fällen die Überlegenheit kooperativer, konstruktivistischer, problemorientierter oder auch multimedialer Lernumgebungen gegenüber traditionellen eher frontal und fremdorganisiert ausgerichteten Lernsettings aufzeigen“ (Schüßler, 2004, S. 62), allerdings haben auch eine Reihe sozio-kultureller und bildungssystematischer Faktoren einen großen Einfluss auf Erfolg oder Misserfolg, wodurch die bloße Verwendung konstruktivistischer Methoden noch kein Garant für Gelingen ist, da sie an die jeweilige Lernsituation und an das jeweilige Publikum angepasst werden sollten. Ich halte es beispielsweise auch bezüglich der Effizienz von Selbstbestimmung im Lernprozess für durchaus möglich,

dass unser aktuelles Bildungssystem hierbei, durch Ermangelung an früher Förderung, eine etwas kontraproduktive Wirkung auf die Lernenden hat. Hier wäre eventuell eine empirische Evaluation der Effizienz von konstruktivistischen Methoden zwischen alternativ-pädagogischen Schulmodellen, wie Waldorf oder Montessori Schulen, die sehr früh damit beginnen selbstbestimmtes Lernen zu fördern, mit staatlichen Institutionen sehr aufschlussreich.

### **2.3 Postulate von Kersten Reich**

Zusammenfassend und reflektierend möchte ich die Postulate von Kersten Reich präsentieren und sie mit drei grundsätzlichen Begriffen seiner Theorien, nämlich Konstruktion, Rekonstruktion und Dekonstruktion erörtern sowie durch die didaktischen Reflexionen von Tulodziecki et al. (2009, S. 252) kombinieren, um dadurch kurz und knapp die Grundpfeiler seiner Theorie zu skizzieren, die mir als Leitfaden und Basis für meine weiteren Ausarbeitungen dienen sollen.

#### **1. „So viel Konstruktion wie möglich!“ (Reich, 2010, S. 122 ff)**

Leitgedanke: *„Wir sind die Erfinder unserer Wirklichkeit!“* (Reich, 2008, S.138)

Da nach Kersten Reich bei rezeptiv angeeignetem Wissen die Wahrscheinlichkeit höher ist, dass die erlernten Inhalte flüchtiger als bei Verwendung von konstruktivistischen Methoden erinnert werden können, was sich mit der reduzierten synaptischen Vernetzung erklären lassen würde, ist das erste Postulat nach Reich wohl wenig überraschend. Konstruktion im Unterricht zu ermöglichen bedeutet den Lernenden die Möglichkeit zu gewähren, dass sie selbst erfinderisch und gestalterisch tätig sein können, was etwa dem Prinzip des Lernens durch selbstbestimmtes Experimentieren oder dem des eigenständigen Begründens entspricht. Dabei sollte die subjektive Weltsicht der Lernenden als Ausgangspunkt genutzt werden, um zur Förderung einer breiteren synaptischen Vernetzung an Erlerntes und an bereits gemachte Erfahrungen anknüpfen zu können, was wiederum für Lehrende bedeutet, dass hier die Notwendigkeit besteht die Lebenswelt der Lernenden zu erkunden und Themen sowie Methoden zu finden, die sowohl die selbstbestimmte Konstruktion von Wissen ermöglichen und zugleich auch so vielschichtig wie möglich an die Erfahrungen und Interessen der Lernenden anknüpfen. Mit den den Worten von

Reich: „Selbst erfahren, ausprobieren, experimentieren, immer in eigene Konstruktionen ideeller oder materieller Art überführen und in den Bedeutungen für die individuellen Interessen-, Motivations- und Gefühlslagen thematisieren.“ (Reich, 2010, S. 119)

Zur besseren Illustration könnten die didaktische Perspektive der Konstruktion im Rahmen einer Metapher sehr oberflächlich mit dem Bau eines sehr archaischen Hauses verglichen werden, und zwar am besten mit dem Bau des ersten Hauses (somit ohne Vorlage), wobei lokal verfügbare Ressourcen wie in unseren Breitengraden zum Beispiel Holzprodukte (Harz, Rinde und Stämme) verwendet werden und die Form des Hauses an die Bedingungen der Umwelt angepasst werden, so zum Beispiel das Spitzdach, um das Gebäude besser vor Niederschlag zu schützen. Dieser Prozess des Baus und der Adaptierung ist selbstbestimmt und selbsttätig, mit einem experimentellen und explorativen Charakter, wodurch die Nähe zum Prozess des Erfindens deutlich wird.

## **2. „Keine Rekonstruktion um ihrer selbst willen!“ (Reich, 2010, S. 133 ff)**

Leitgedanke: „*Wir sind die Entdecker unserer Wirklichkeit!*“ (Reich, 2008, S. 139)

Der Gedanke hinter dem rekonstruktiven Vorgehen ist in unserem Bildungssystem sehr zentral, denn es geht dabei um das Verständnis durch Nachvollziehen und nicht mehr wie bei den Konstruktionen um ein erfinderisches oder experimentelles Lernen. Hierfür sollten die Lernenden jenen kreativen Freiraum geboten bekommen, in dem sie die Lerninhalte selbsttätig erkunden und erfahren können. Im Geschichtsunterricht könnte das beispielsweise bedeuten, dass nicht bloß Fakten und Jahreszahlen zu geschichtlichen Ereignissen auswendig gelernt werden, sondern dass diese in einem von den Lernenden selbst entworfenem Schauspiel aufgeführt werden, wodurch eine intensivere Beschäftigung mit der Thematik, sowie auch eine emotionale Beziehung zum Thema erreicht werden kann, was einem tieferen Verständnis der Materie durchaus zuträglich wäre. Die zentrale Idee ist es also, dass die rezeptive Inhaltsvermittlung zumindest ergänzt, wenn nicht nach Möglichkeit sogar ersetzt wird, durch ein Nach-Spielen, Nach-Empfinden und schließlich ein Nach-Vollziehen der Situation, des ursprünglichen Problems und des Weges, der letztendlich zur Lösung führte, wodurch die Ebene des Inhalts breiter gefächert und durch jene der

Emotionen erweitert wird. Zitat: „Wir fragen nach Motiven und wollen nicht bloß Fakten lernen.“ (Reich, 2010, S. 120)

Rekonstruktion um ihrer selbst willen kann zu einem Kreislauf führen, in dem eigentlich nur noch rein rezeptives Wissen tradiert wird, da das Lernen zu sehr auf die Aneignung von Fakten und Tatsachen beschränkt wird. Es bestünde hier die Möglichkeit, die Rekonstruktionen durch Verwendung passender Methoden in Konstruktionen bzw. Rekonstruktionen zu überführen, wobei aber natürlich auch ein Grund vorliegt, warum sich im Bildungsbereich seit jeher Rekonstruktionen um ihrer selbst willen und mithin rezeptive Wissensvermittlung gehalten haben, denn sie bieten den großen Vorteil der einfachen Überprüfbarkeit durch das simple Abfragen und Prüfen von Fakten. In einem Bildungssystem, in dem Zugänge durch Wissen beziehungsweise Wiedergabe von Wissen beschränkt und kontrolliert werden, scheint sich ein auf Inhaltsvermittlung fokussiertes und weiters ein auf Vermittlung von rezeptiven Inhalten fokussiertes System als wesentlich zweckdienlicher zu erweisen, als konstruktivistische Methoden. Mittlerweile beginnt bloßes Wissen als alleiniger Maßstab aber ein wenig in den Hintergrund zu treten und wird von einer neueren, zentralen Leitidee allmählich abgelöst, nämlich der Kompetenz. Der Wechsel vom "Was?" hin zum "Wie?" ist ein konzeptueller Wandel, der eine interaktionistisch konstruktivistische Didaktik, wie sie Kersten Reich vertritt, begünstigen könnte, da Kompetenz ein "verstanden haben" voraussetzt, das dann aber auch den großen Vorteil hat, dass es auf andere, verwandte Bereiche, wie etwa eine andere Programmiersprache, anwendbar ist. In der Informatik wären im Themenbereich der Programmierung beispielsweise das Verständnis algorithmischer Abläufe und allgemeine semantische Grundlagen ein Kapitel, das als Kompetenzwissen auf fast alle Programmiersprachen angewandt werden kann.

In der Metapher mit dem Hausbau wäre die Rekonstruktion etwa die Nachbildung eines Hauses. Angenommen unser Architekt fährt mit dem ersten Schiff nach Ägypten und sieht dort voller Staunen eine komplett andere Art des Hausbaus, mit anderen Materialien, wie Lehm, und angepasst an andere Umweltbedingungen, wie etwa große Hitze. Unser Architekt darf nun Vorort am Hausbau mitwirken, erfahren, weshalb gerade jene Materialien verwendet werden und kann dadurch rekonstruieren, warum die ägyptischen Häuser anders aussehen und gestaltet sind, als

die seiner Heimat. Wichtig bei der Rekonstruktion des Hauses ist der handlungsorientierte Ansatz, denn der ägyptische Architekt belehrt nicht bloß theoretisch über die Art und Weise des Baus, was rein rezeptive Vermittlung wäre, sondern lädt den europäischen Kollegen dazu ein, sich direkt am Bau zu beteiligen, wodurch dieser wiederum eigene Erfahrungen sammeln kann. Hier ist also die Quintessenz des Prozesses vergleichbar mit dem des Entdeckens.

### **3. „Keine Konstruktionen ohne Verstörungen!“ (Reich, 2010, S. 139 ff)**

Leitgedanke: „*Wir sind die Enttarnen unserer Wirklichkeit!*“ (Reich, 2008, S. 141)

Die Idee hinter der Dekonstruktion ist die Frage, ob es nicht auch noch anders sein könnte (vgl. Reich, 2008, S. 141). Dies zielt meiner Meinung nach vor allem darauf ab zu verhindern, dass Lernende an bestimmte Muster gewöhnt werden und dass ihnen dadurch ihre eigene Kreativität ein Stück weit aberlernt wird. Die Sicht auf die eigenständig und individuell konstruierte Welt sollte durch Bildung nicht auf Lösungen eingeschränkt werden, die von anderen bereits entworfen wurden, sondern es sollte stets hinterfragt werden, ob nicht auch andere Wege möglich gewesen wären. Es geht hierbei wohl weniger um eine breiter gefächerte synaptische Vernetzung der Lerninhalte oder eine tiefere Einbettung in die individuelle Welt, als vielmehr um das Aufrechterhalten des kreativen, individuellen Denkens durch Aufzeigen verschiedenartiger Lösungen und damit zusammen hängend um das Auflockern sozio-kultureller Paradigmen im Bildungsprozess.

Der didaktische Prozess des Wissenserwerbs sollte daher zum Abschluss mittels Dekonstruktion noch in ein Spannungsverhältnis gebracht werden, etwa durch "Kritik, Skepsis, Verfremdung, Ironie" (Reich, 2010, S. 140), da der Lernprozess auch immer ein sozio-kultureller Akt ist und somit verwoben in die Symbolwelten, Ideologien und moralisch-ethischen Systeme sowohl der Lehrenden, als auch der Lernenden, die durch Verstörung und Dekonstruktion davon möglichst entschlackt, sowie kritisch analysiert werden sollen. Betrachtet man im Unterricht den Ablauf geschichtlicher Ereignisse, so könnte beispielsweise ein perspektivischer Wechsel von Opfer und Täter Rollen zur Dekonstruktion verwendet werden.

Ein Mittel der Dekonstruktion wäre zum Beispiel auch die Entfremdung, die in der Metapher dazu führen könnte, dass der Architekt, der den ägyptischen Hausbau erfahren hat, nun in seiner mitteleuropäischen Heimat ein ägyptisches Haus errichtet, das dadurch seinem ursprünglichen sozio-kulturellen Umfeld entrissen und in ein Neues eingebettet wird. Dadurch könnte unter anderem in Erfahrung gebracht werden, was unsere Architekten von den ägyptischen Kollegen lernen und weiter verwenden können oder durch den Kontrast auch klarer werden, warum unsere Häuser auf ihre spezifische Art und Weise errichtet werden. Es besteht auch die Möglichkeit der Entschlackung von sozio-kulturellen Restriktionen, wie zum Beispiel, dass kein Haus das Haus Gottes überragen darf, wodurch kritische Betrachtungen und Analysen hier eine besonders zentrale Rolle spielen.

### **2.3.1 Das Zusammenspiel der drei Grundbegriffe**

Konstruktion, Rekonstruktion und Dekonstruktion haben eines gemeinsam und das ist der Bezug zu der individuellen Wirklichkeit des Menschen (vgl. Reich, 2010, S. 118). Nach meiner Interpretation und Auslegung scheint Reich mit diesen Begriffen den Lernprozess in drei miteinander und aufeinander einwirkende Phasen aufzuteilen, die einander bedürfen. So scheint die Konstruktion vor allem für die anfängliche Erkundung eines Themas gut geeignet, in der unter anderem Hypothesen und Brainstorming sehr willkommen sind, um daraus eine Neugier für die Thematik des Inhalts zu entfachen. Die Rekonstruktion versucht dann zu verständlichen, was bereits an Lösungen vorhanden ist sowie diese nachvollziehbar zu machen, was nicht nur anhand von Fakten, sondern vor allem durch eigenständige Tätigkeiten geschehen sollte. Die Dekonstruktion ist nun der Motor, der beispielsweise die Wissenschaft voran treibt, da durch das ständige Hinterfragen der momentanen Lösungen und Ergebnisse wiederum auch ständig neue Perspektiven und Konstruktionen erschaffen werden. Als kurzes illustratives Beispiel für eine Konstruktion sei „USB 1.0“ gegeben, als eine Erfindung, die eine möglichst störungsfreie Übertragung von Daten sicher stellt. Dass auch diese Erfindung auf einem bestimmten Vorwissen beruhte, sei zum Zwecke der Veranschaulichung vernachlässigt. Möchte sich jemand anderes als der Erfinder mit dieser Technik auseinander setzen, so bedarf es dem Nachvollziehen der Funktionsweise dieser Technologie. Hierbei wird also die Rekonstruktion von Bedeutung, da nicht mehr

erfunden sondern „nach-erfunden“, oder besser: nachvollzogen wird. Die Dekonstruktion setzt nun ein sobald hinterfragt wird, ob es nicht auch anders gehen würde und könnte in diesem Beispiel sogar zur Entwicklung von „USB 2.0“ führen, was an sich wieder eine Leistung der Konstruktion wäre.

Für das Verständnis der drei Begriffe scheint es auch sehr bedeutungsvoll zu sein, dass es sich hierbei um Perspektiven handelt, die in einem Lernprozess sowohl von Lernenden, als auch von Lehrenden bezogen auf das Thema eingenommen werden. Es sind also keine fertigen Lösungen oder Strategien sondern eher flexible Handlungsmuster und Beobachterperspektiven auf die jeweiligen Gegenstände des Unterrichts (vgl. Reich, 2010, S. 118 ff.). Das Zusammenspiel der drei Grundbegriffe soll daher nicht als ein Rezept verstanden werden, dass bereits durch strikte und akkurate Einhaltung einen erfolgreichen Unterricht hervor bringt, sondern in erster Linie als eine Theorie aufgefasst werden, die bemüht ist, dass „die eigene konstruktive Erkenntnistätigkeit“ (Reich, 2010, S. 119) mehr gefördert wird, als es bei rein rezeptiven Methoden und Modellen der Fall ist. Allerdings möchte ich nicht unerwähnt lassen, dass eine Symbiose aus konstruktivistischer Didaktik und kompakter, rezeptiver Vermittlung dadurch nicht ausgeschlossen werden soll. Inhalte, die kurz und prägnant präsentiert werden, können als Basis für konstruktivistische Methoden eine große Zeitersparnis darstellen und daher sollte ihre Bedeutung nicht geschmälert werden, wenngleich sich der Unterricht auch nicht darauf reduzieren sollte.

### 3 MULTIPLAYER CLASSROOM

*„Aber dies zeigt nur einmal mehr,  
dass zu einer guten Arbeit auch ein vernünftiger Plan gehörte.“*

*-Kersten Reich (2010, S. 152)*

#### 3.1 „World of Warcraft“

Bevor ich näher auf das Unterrichtskonzept von Lee Sheldon eingehe, möchte ich erst eine der wohl bedeutendsten Quellen erläutern, die ihm als Inspiration dienten, nämlich „World of Warcraft“, entwickelt von Blizzard Entertainment. Dabei handelt es sich um ein sogenanntes MMORPG - Massively Multiplayer Online Role-Playing Game, also ein Rollenspiel in einer permanenten virtuellen Welt, die von sehr vielen Spielern zeitgleich bewohnt wird. Oftmals tauchte dieser Titel in den Medien auf, die von Spielern berichteten, deren einziger Sinn und Zweck im Leben nur noch dieses eine Spiel zu sein schien, wobei es auch tragischerweise vereinzelt zu Todesmeldungen kam. Meldungen in Online-Magazinen und Zeitungen, etwa „Tödliche Erschöpfung durch Dauerspielen“ (Klaß, 2005), zeigten leider immer wieder auch die Schattenseiten des Erfolges auf.

Das Spiel brach viele Rekorde, die Zahl der globalen Abonnements erreichte oftmals rund 12 Millionen<sup>4</sup>, was ein beachtlicher Erfolg ist. Dabei mag der Gedanke, das teils schockierende Suchtpotenzial eines solchen Spiels produktiv nutzbar zu machen, wohl so manchen Pädagogen oder Manager beschleichen, wobei eine solche Ausarbeitung immer auch die Gefahr beinhaltet zu inakkurat oder zu verspielt umgesetzt zu werden. Die Schwierigkeit hierbei ist es wohl vorrangig, dass es besonders jene Elemente des Spiels heraus zu arbeiten gilt, die im Stande sind die Motivation und Transparenz zu fördern, und zugleich jene auszublenden, die eine eher irritierende und ablenkende Wirkung aufweisen.

---

<sup>4</sup> Quelle: <http://mmodata.net/> [zuletzt abgerufen am 08. November 2012]

### **3.1.1 Spielablauf**

Um die Mechaniken und Funktionsweisen des Spieldesigns zu illustrieren, ist es unumgänglich, zuerst einmal das Prinzip und den Ablauf des Spiels zu erörtern. Zu Beginn wird vom Spieler ein Charakter erstellt, der dann durch die virtuelle Welt gesteuert wird, wobei ausgewählt wird, welcher der beiden verfeindeten Fraktionen (Allianz oder Horde) man als Spieler angehören möchte, welchem der jeweils 5 Völker der Fraktionen der Charakter entstammen soll und welche der 10 Klassen man spielen möchte.

Das erste Ziel ist es, nach der Erstellung des eigenen Spielcharakters, diesen durch Erfüllung von Quests (Aufgaben) oder das Töten von Gegnern auf das höchste erreichbare Level zu bringen, das momentan (19.09.12) noch die Stufe 85 ist. Erlangt man die ersten Stufen noch relativ schnell und einfach, so steigt die benötigte Anzahl an Erfahrungspunkten für den Aufstieg pro Stufe (Level) immer weiter an. Dabei werden auch immer mehr Fähigkeiten, die dem Spielcharakter zur Verfügung stehen, freigeschaltet, sodass er sich stets weiter entwickelt und es immer wieder Zwischenziele zu erreichen gilt. Oftmals erweisen sich die Aufgaben als recht monoton oder langwierig, manche wiederholen sich angesichts des Ablaufs des öfteren, doch gerade diese sehr repetitiven Hürden im Aufstieg des Charakters weisen angesichts der hohen Spielerzahlen auf eine gelungene Motivationsstrategie hin.

Zusätzlich zu den Fähigkeiten gibt es als Anreiz auch noch verschiedene Ausrüstungsgegenstände, die entweder als Belohnung für absolvierte Aufgaben, oder in Form von Beute durch getötete Gegner erbeutet werden können. Diese Gegenstände erhöhen verschiedene Attribute und verbessern dadurch den Charakter, sofern dieser eine bestimmte Mindeststufe erlangt hat, die für das Tragen benötigt wird, und über die generelle Fähigkeit verfügt, einen Gegenstand dieser bestimmten Fertigungsart zu tragen. Mit steigender Stufe des Charakters können immer bessere Gegenstände erbeutet und getragen werden, wodurch die Ausrüstung auch immer mehr an Wert und Bedeutung gewinnt, denn der prozentuale Anteil an den Attributen, den die Ausrüstung im Spiel einnimmt, wächst mit steigender Stufe.

Gegenstände, wie Rüstungen und Waffen, können auch von einigen Spielern entweder käuflich (durch Bezahlung mit der Spielwährung Gold) erworben, oder durch die Spezialisierung auf bestimmte Berufe gefertigt werden. Von den 11 primären Berufen können pro Spielcharakter jeweils nur 2, die 4 sekundären Berufe können hingegen allesamt erlernt und gemeistert werden. Berufe unterliegen einem ähnlichen System, wie der Stufenaufstieg, wobei pro Level des Charakters eine gewisse Menge an Fertigkeitspunkten (bis Level 65: 5 Punkte, danach 10 Punkte) durch Ausübung der Berufe dazu erhalten werden kann. Das Erstellen von Gegenständen einer höheren Stufe benötigt somit auch eine höhere Fertigungsstufe. Gerade die Einbindung von Berufen, einer virtuellen Abwandlung tatsächlicher Arbeit, die meist noch sehr mühevoll und monoton abläuft, ist ein sehr interessanter Bereich, der sich auch in anderen MMORPGs wiederfinden lässt, wie etwa in „Star Wars Galaxies“, was unter anderem die Kommentare von SpielerInnen in den Untersuchungen von Nick Yee (2006) belegen. Hierdurch zeichnen sich durchaus nicht nur die eigentlichen Berufe in den Spielen, sondern auch Meta-Aufgaben, wie die Organisation von Gilden, Teams und Raids aus, wodurch sich ein Spiel eventuell zu einer Form von virtuellem Nebenberuf entwickeln kann.

Auf Stufe 85 ist das Spiel aber bei Weitem nicht beendet, für den größten Teil der Spieler beginnt es wahrscheinlich erst ab diesem Zeitpunkt, da nun verschiedene „Dungeons“, also instantiierte Spielgebiete besucht werden können, die mit steigendem Schwierigkeitsgrad auch verlockendere und bessere Ausrüstungsgegenstände bieten, die bei einigen „Bossen“, sehr mächtigen NPCs (engl.: non-player character, also ein vom Spiel gesteuerter Charakter) nach deren Tod erbeutet werden können. Seit einiger Zeit können Spieler bei den instantiierten Bossen auch Marken erhalten, die dann in weiterer Folge für Gegenstände eingetauscht werden können, sofern man eine ausreichende Menge dieser Marken besitzt.

Interessant anzumerken ist hierbei, dass solche Instanzen auf hoher Stufe niemals als einzelner Spieler zu bewältigen sind, sondern nur mit anderen Spielern, gemeinsam in Gruppen. Möchte ein Spieler nun einen der Dungeons besuchen, so werden daher weitere Mitstreiter benötigt, was als eine Art sozialer Katalysator gesehen werden

kann, denn dadurch besteht der Bedarf, sich mit anderen zusammen zu schließen, zu kommunizieren und miteinander diese Instanzen zu besuchen. Dieser Punkt wird vor allem später von Bedeutung sein, da der soziale Faktor nicht nur für das Spieledesign, sondern auch für eine mögliche didaktische Umsetzung bemerkenswert ist. Um nun die Koordination und Kommunikation zu vereinfachen, hat Blizzard einige Mechaniken und Strategien elaboriert. Eine solche, die schon von Anfang an im Spiel implementiert war, ist die Gilde: eine Gemeinschaft von Spielern, die einige komfortable Funktionen bietet, um sich untereinander zu helfen, sich zu unterhalten und Schlachtzüge („Raids“) in die großen Dungeons des Spiel zu koordinieren.

Diese zwei Abschnitte, das Erlangen der höchsten Stufe und der gemeinsame Besuch der instantiierten Dungeons, bilden mit der sozialen Komponente die tragenden Säulen des Spiels und sind für die weiteren Betrachtungen und Überlegung von großer Bedeutung, da Lee Sheldon viele seiner Ideen aus ihnen schöpft und sie teilweise übernimmt.

### **3.1.2 Kommunikation und Sozialisation im Spiel**

Ein „wesentlicher Bestandteil“ (vgl. hierzu Himmelbauer, 2011, S. 118 ff.), der ausschlaggebend für die Spielmechanik und vor allem für das Funktionieren der Sozialisation im Spiel ist, wäre die bereits erwähnte Kommunikation unter den Spielern. Hierfür hat Blizzard Entertainment einige sogenannter „Channels“ oder Chat-Kanäle eingerichtet, manche für eine Kommunikation die große Gebiete umspannt, so etwa der Channel „Handel“ in dem Spieler, die sich in einer der Hauptstädte befinden, ihre erbeuteten oder hergestellten Waren zum Verkauf anbieten oder nach speziellen Gegenständen fragen können. Andere dieser Kommunikationskanäle, etwa der Channel „Gilde“, ist nur für Mitglieder einer bestimmten Gilde nutzbar, oder auch der Channel „Gruppe“, der jeweils nur für die Mitglieder einer Gruppe zur Verfügung steht.

Durch diese Kommunikationsstrukturen wird die soziale Dynamik unter den Spielern begünstigt und es werden Anlaufstellen für Fragen und Antworten unter Gleichgesinnten geboten. Darüber hinaus besteht die Möglichkeit der Kommunikation mit einem der „Game Master“, Angestellten von Blizzard, die Anfragen über ein Ticketsystem beantworten, was aber meist nur bei Problemen mit dem Spiel an sich, wie etwa defekten NPCs oder anderen auf die Spielmechanik zurückzuführenden Fehlern benutzt wird. Gibt es beispielsweise Komplikationen mit einer der gestellten Aufgaben, so fungieren die Mitspieler und Mitspielerinnen meist als erste Anlaufstelle.

Der Zusammenschluss von Spielern erfolgt entweder über Gruppen, Schlachtzüge oder Gilden, wobei erstere temporäre Bündnisse darstellen, nur die Gilde ist ein über das Beenden des Spiels hinweg permanentes Bündnis. Allen dreien ist gemeinsam, dass sie automatisch eigene Chat-Kanäle erzeugen, um die Kommunikation zu vereinfachen und über mindestens einen oder auch mehrere Leiter verfügen. Abseits dieser Bündnisse besteht auch die Möglichkeit, einen Mitstreiter in die Freundesliste aufzunehmen.

### **3.1.3 Nur ein Spiel**

In seinem Film „Nur ein Spiel“<sup>5</sup> erörtert Michael Schilhansl die facettenreiche Faszination der Rollenspiele, beginnend bei Live Rollenspielen (engl.: Live Action Role Playing) über das Pen and Paper Spiel „das schwarze Auge“ bis hin zu Computerspielen (Gothic 3 und World of Warcraft). In dem Abschnitt zu World of Warcraft kommen einige Spieler zu Wort, die unter anderem die Gruppe als Motivationsfaktor nennen. Aslan, einer der interviewten Spieler meinte beispielsweise, dass es Momente gab, in denen er an das Spiel fast schon wie an Arbeit herantrat (ab 46:10 im Video), wobei auch beim ihm die Gruppe und die Gemeinschaft anscheinend einer der entscheidenden Faktoren waren, trotz dem Fehlen der Unterhaltung durch das Computerspiel, das Mühsal auf sich zu nehmen.

---

<sup>5</sup> Zu finden unter <http://vimeo.com/16064142> [zuletzt abgerufen am 08. November 2012]

Dies könnte ein Hinweis auf eines der grundlegenden Prinzipien sein, die bei dem Konzept von Lee Sheldon eine lernförderliche Wirkung hervor bringen, nämlich, pädagogisch gesprochen, Gruppenarbeiten. Außerdem wären Eigenständigkeit im Arbeitsfluss und eine gelungene Rollenverteilung innerhalb der Gruppe weitere Faktoren, die auf ein didaktisches Design umgelegt werden könnten. Durch die Verteilung der Arbeitslast auf die verschiedenen Klassen im Spiel wird durchaus auch die Eigenständigkeit (hier: die Beherrschung des eigenen Spielcharakters) gefördert, die für das Lernen und den Erfolg eines pädagogischen Konzeptes, wie es bei Kersten Reich erörtert wird (siehe hierzu Kapitel 2 „Konstruktivistische Didaktik“), einen Grundpfeiler darstellt oder darstellen sollte.

Gruppenarbeiten eilt aber teilweise auch der Ruf voraus, dass sie den Lernerfolg von schwächeren Lernenden dadurch hindern, dass diese die Arbeit lieber von besseren Gruppenmitgliedern erledigen lassen, wobei hier auch der Faktor Zeit eine Rolle spielen könnte, da es effizienter ist, wenn diejenigen in der Gruppe die Arbeit machen, die sie schneller und leichter erledigen können. Das Spieldesign hat der Gruppenarbeit voraus, dass es bei Schlachtzügen eine fixe Rollenverteilung gibt, das heißt, wenn jemand als Heiler eingeteilt ist, dann kann nur dieser Charakter-Typus die Rolle erfüllen. Gelingt dem Spieler die Handhabung nicht, so wird er wahrscheinlich nach einiger Zeit aus der Gruppe ausgeschlossen und kommt nicht an das gewünschte Ziel. Durch eine Vielzahl an sogenannten Add-Ons, kleinen Programmen, die in das Spiel integriert werden können, ist es möglich, dass die Leistung der einzelnen Mitstreiter sehr genau protokolliert wird und auch transparent ist. Aus eigener Erfahrung kann ich bestätigen, dass auch dadurch die Motivation, zumindest im Spiel, gesteigert werden kann, wenn zu jeder Zeit ersichtlich ist, welche Leistung man selbst erbracht hat.

### **3.1.4 Erfolgsrezept?**

In einem Online Artikel mit dem Titel „Das Erfolgsgeheimnis von World of Warcraft“ gibt es einige Erklärungen des Vizepräsident Rob Pardo bezüglich Erfolgsrezept des Spieltitels: „Die Spieler müssen einen leichten Einstieg finden. Die

Menüs dürfen nicht überfordern und dem Spieler nicht von Anfang an zu viel Einblick in die Welt schenken. Die Abenteuer, die Quests, sollen schnell erreichbar sein, der Spieler soll den Wunsch entwickeln mehr erreichen zu wollen. Es muss Anreize geben, um in höhere Level aufzusteigen, also die Charakter-Erfahrung auszubauen. Kleinigkeiten wie bessere Monturen oder Waffen dienen diesem Vorhaben.“<sup>6</sup> Demnach steht zu Beginn eine Reduktion der Komplexität, um das Erlernen der Handhabung zu begünstigen. Die Aufgaben oder Quests sollen schnell erreichbar sein und keine unüberwindbaren Hürden darstellen, wodurch eine Art Spielfluss entsteht, in dem sich die Spielerin oder der Spieler fast schon treiben lassen kann. Zusätzlich soll es auch noch Anreize geben, welche die Motivation stets in einem Bereich halten, sodass ein Weiterspielen erstrebenswert erscheint.

Beachtenswert sind auch generell Faktoren, welche die Spieldauer erhöhen, da sie mit großer Wahrscheinlichkeit einen beträchtlichen Einfluss auf die Motivation ausüben. Betreffend Online-Spielen wären hier vorrangig 4 Prinzipien zu nennen: Persistenz, Grenzenlosigkeit, Kommunikation und soziale Beziehungen (Schmidt, Dreyer & Lampert, 2008, S. 53 f.).

So schlüssig und simpel diese Prinzipien auch klingen mögen, deren Umlegung auf ein pädagogisches Konzept kann durchaus einiges Kopfzerbrechen bereiten. Oftmals kann schon der Punkt der Reduktion der Komplexität in einem didaktischen Konzept eine große Herausforderung darstellen.<sup>7</sup> In der Informatik könnte dies für Programmieranfänger zum Beispiel bedeuten, dass Editor beziehungsweise Lernprogramm mit Bedacht gewählt werden, möglichst intuitiv zu bedienen und schnell zu beherrschen sein sollen. Der Fokus soll also durch Editor oder Programm nicht von der Programmierung oder den Algorithmen abgelenkt werden, sondern Lernenden dabei helfen den Blick auf das Wesentliche zu lenken.

---

<sup>6</sup> Quelle: derStandard.at (2007). Das Erfolgsgeheimnis von World of Warcraft.

URL: <http://derstandard.at/2578098> [letzter Zugriff: 07. November 2012].

<sup>7</sup> siehe hierzu Lehner (2012)

### **3.2 Das Unterrichtskonzept von Lee Sheldon**

Lee Sheldon entwickelte ein eher unkonventionell anmutendes Konzept, um seine Studentinnen und Studenten zu unterrichten. Die Inspiration hierfür war das Computerspiel „World of Warcraft“. Der große Vorteil, den Sheldon bei der Anwendung seines didaktischen Konzeptes hat, ist jener, dass sein Fach Spiele-Design ist und somit ist die Nähe zu einem Rollenspiel etwas unkomplizierter oder naheliegender ist, als beispielsweise im schulischen Alltag. Die Einbettung des Unterrichts in eine Geschichte, welche die Lernende fesselt und motiviert ist dabei auch kein triviales Unterfangen. Allerdings ist aus dem Feedback der Studierenden einiges an interessanten Fakten zu entnehmen, die durchaus auch in den regulären Unterricht übernommen werden können oder zumindest hilfreich bei der Reflexion von Methoden, Benotung und Transparenz hinsichtlich einer höheren Motivation im Unterricht sein könnten. Von seinem Konzept zeigt sich Sheldon jedenfalls überzeugt: „Yet, as we shall see, designing classes as games has produced some very interesting results in classrooms ranging from middle school through university-level courses in a variety of disciplines. We have yet to discover a class that cannot be taught in this way. Maybe we never will.“ (Sheldon, 2011, S. 9)

Es sind einige Fallstudien aus anderen Fächern und Institutionen in seinem Buch „The Multiplayer Classroom“ vermerkt, so zum Beispiel auch eine Abwandlung für den Biologie Unterricht, entwickelt von Denishia Buchanan, die in einer High School unterrichtet, in der 80% der Schüler und Schülerinnen unter der Armutsgrenze leben (vgl. Sheldon, 2011, S. 49 ff). Hier konnte eine durchschnittliche Leistungssteigerung von 62% mit der Note D oder höher auf 98% verbucht werden, wobei angemerkt werden muss, dass es sich hierbei um den Vergleich zweier verschiedener Jahrgänge handelt, wodurch sich mitunter auch einige andere Faktoren für die Leistungssteigerung verantwortlich zeigen könnten.

Sheldon berichtet, dass seine Studierenden sich nach dem Umstieg von einem konventionellem System auf sein recht eigenwillig anmutendes Konzept im Durchschnitt verbessert hätten, nämlich von „C“ auf „B“ (vgl. Laster, 2010), wobei die freilich nur ein Fallbeispiel ist und eine weitaus längerfristige und

weitreichendere Untersuchung notwendig wäre, um sich auf aussagekräftige, empirische Belege stützen zu können.

### 3.2.1 Der Lehrplan

Der Lehrplan ist in einer sehr frühen Version auf dem Blog „Gaming the Classroom“<sup>8</sup> zu finden, den Lee Sheldon aufgrund der regen Anfragen nach seinem Unterrichtskonzept ins Leben gerufen hat. Dort finden sich auch Feedback der Studierenden und weiterführende Anmerkungen sowie Reflexionen zu seinem Unterrichtskonzept zur Nachlese. Die Idee zur Entwicklung eines Lehrplans durch einen Game-Designer gab es schon vor Sheldon: „The crux of what I want to say is that game designers have a better take on the nature of learning than curriculum designers.“ (Papert, 1998) Das tägliche Brot von Game-Designern ist es, Menschen zu bewegen, sie mit verschiedenen Formen von Anreizen über schwierige Hürden hinweg zu motivieren, was gerade bei komplexeren Spielen mit einer steileren Lernkurve den Schlussstein bildet und somit das ganze Konstrukt trägt. Daher meint Papert auch, dass sie die besseren Lehrpläne erstellen würden, denn immerhin sind sie sehr gut mit Mechaniken und Techniken vertraut, die Motivation, Neugier und Interesse fördern.

Der Unterricht wird in verschiedene Abschnitte unterteilt, die eine große Ähnlichkeit mit bekannten Untergliederungen aufweisen: „Class time will be divided between fighting monsters (Quizzes, Exams etc.), completing quests (Presentations of Games, Research etc.) and crafting (Personal Game Premises, Game Analysis Papers, Video Game Concept Document etc.).“<sup>9</sup> Die Kategorisierung der Unterrichtsabschnitte wäre also in etwa wie folgt in eine didaktischere Sprache zu übersetzen:

---

<sup>8</sup> URL: <http://gamingtheclassroom.wordpress.com/> [zuletzt abgerufen am 09. Dezember 2012]

<sup>9</sup> siehe <http://gamingtheclassroom.wordpress.com/syllabus/>

- Bekämpfen von Monstern (fighting monsters): verschiedene Prüfungs- und Testmodalitäten
- Erfüllen von Quests: Präsentationen, Recherchen
- Anfertigen (crafting): Erstellung von Konzepten, Prämissen, Dokumenten

Zu Beginn des Semesters werden die Avatare der Studierenden gewählt und benannt, sowie die Gilden anhand der Interessengebiete der einzelnen Studierenden gegründet, wodurch befreundete Studierende sich eher auf unterschiedliche Gilden verteilten und gefördert wurde, dass neue Bekanntschaften und soziale Kontakte entstehen konnten. Jede Gilde besteht aus 6-7 Mitgliedern und wurde einer bestimmten Zone zugewiesen, also Bereiche im Klassenzimmer, in denen jeweils eine Gruppe angesiedelt war. Durch das Rotieren der Gruppen im Raum wurde zu verhindern versucht, dass die Gruppen, die weiter hinten im Raum waren, allzu leicht schummeln konnten (vgl. Sheldon, 2011, S. 35).

„Good morning. Welcome to the first class of the semester. Everyone in this class is going to receive an F.“ (Sheldon, 2011, S. 3) Jeder Studierende beginnt bei 0 Erfahrungspunkten (engl.: experience points, kurz „XP“) und kann durch die Erfüllung von Aufgaben und das „Bekämpfen von Monstern“ immer mehr dieser Punkte anhäufen. Die Anzahl der Punkte entscheidet dann am Semesterende über die ausgestellte Note, wobei Sheldon die Punkte wie folgt auf das amerikanische Notenschema übersetzt, wozu er in diesem Beispiel die prozentuale Notenskala der Indiana University übernommen hatte (Sheldon, 2011, S. 24):

| Level        | XP (experience points) | Letter Grade |
|--------------|------------------------|--------------|
| Level Twelve | 1860                   | A            |
| Level Eleven | 1800                   | A-           |
| Level Ten    | 1740                   | B+           |
| Level Nine   | 1660                   | B            |
| Level Eight  | 1600                   | B-           |
| Level Seven  | 1540                   | C+           |
| Level Six    | 1460                   | C            |
| Level Five   | 1400                   | C-           |
| Level Four   | 1340                   | D+           |
| Level Three  | 1260                   | D            |
| Level Two    | 1200                   | D-           |
| Level One    | 0                      | F            |

*Version „Spring 2010“ (vgl. Sheldon, 2011, S. 6)*

Später wurde diese Tabelle von Sheldon überarbeitet, da ein stetiger Fortschritt in Anbetracht der Stufen erst recht spät möglich wird, was durch die große Punktespanne des ersten Levels bewirkt wird. In der überarbeiteten Version gibt es von der Note F zur Note D insgesamt 5 Stufen oder Levels, wodurch sich insgesamt 16 Level ergeben (Sheldon, 2011, S. 98):

| Level          | XP (experience points) | Letter Grade |
|----------------|------------------------|--------------|
| Level Sixteen  | 1860                   | A+           |
| Level Fifteen  | 1800                   | A            |
| Level Fourteen | 1740                   | A-           |
| Level Thirteen | 1660                   | B+           |
| Level Twelve   | 1600                   | B            |
| Level Eleven   | 1540                   | B-           |
| Level Ten      | 1460                   | C+           |
| Level Nine     | 1400                   | C            |
| Level Eight    | 1340                   | C-           |
| Level Seven    | 1260                   | D+           |
| Level Six      | 1200                   | D            |
| Level Five     | 600                    |              |
| Level Four     | 300                    |              |
| Level Three    | 150                    |              |
| Level Two      | 75                     |              |
| Level One      | 0                      | F            |

*Version „Fall 2010“ (vgl. Sheldon, 2011, S. 27 f.)*

Der Großteil der über das Semester (Spring 2010) geplanten Aufgaben sind Einzelarbeiten („Solo“), was eine maximale Punktezahl von 1325 XP, verteilt auf 8 unterschiedlich gewichtete Aufgaben, für eigenständiges Arbeiten ergibt. Durch die

Gruppenaufgaben („Guild“ oder „Raid“) können 525 XP erlangt werden, wodurch wir eine Zwischensumme von 1850 XP erhalten. 150 XP können dann entweder in Form einer Partner- oder wahlweise einer Einzelarbeit erbeutet werden, woraus sich eine maximale Punkteanzahl von 2000 XP ergibt. Zusätzlich können die Studierenden noch „extra credits“ für Zusatzaufgaben erhalten, wie etwa das frühe Einreichen von Arbeiten oder durch ein anonymes Gruppenfeedback.

Um einen besseren Überblick zu erhalten, möchte ich die Liste der unterschiedlichen Aufgaben, inklusive Punkteverteilung, die auch auf dem Blog „Gaming the Classroom“<sup>10</sup> aufgeführt sind, nach Gruppenmodalität geordnet, anführen:

**Solo:**

- Craft your own game proposal. (Written, 50 pts.)
- Present your game proposal to the class. (25 pts.)
- Craft 3 page report on MMO article (Written, 75 pts.)
- Craft 3 page analysis of MMO-based research topic (Written, 100 pts.)
- Craft 5 page analysis on MMO of your choice (Written, 125 pts.)
- Defeat Five Random Mobs (5 written reading quizzes, 250 pts. Total, 1 extra credit question per quiz)
- Defeat Level Boss (Midterm Exam, 400 pts.)
- Class attendance (300 skill pts. total, 10 to start. 290 additional pts. at 10 pts. per day of attendance)

**Guild:**

- Raid: Guild reading presentation (75 pts. each person, 1 of these per guild)
- Paper Prototype Presentation (50 pts. Each)
- Craft Final Project: Video Game Concept (Written, 400 pts.)

---

<sup>10</sup> Siehe: <http://gamingtheclassroom.wordpress.com/syllabus/>

### **Group/Solo:**

- Pick-Up Group: 2-Player reading presentation (150 pts. each person, cannot team with fellow guild member) **OR**
- Solo: 1-Player reading presentation (150 pts. but easier than above)

### **3.3 Feedback**

Unter dem Eintrag „T366 Multiplayer Game Design Post Mortem“ finden sich einige Punkte aus dem Feedback der Studierenden zu dem Konzept.

Der erste Punkt gilt dem konstanten Aufstieg durch Erfahrungspunkte, der durch die zwei großen Blöcke „Guild: Craft Final Project: Video Game Concept (Written, 400 pts.)“ und „Solo: Defeat Level Boss (Midterm Exam, 400 pts.)“ einen eindeutigen Fokus hatte. Dadurch wurde die Motivation der konstanten Mitarbeit etwas verringert und reduziert. Zusätzlich wurde eine Art der Anerkennung für das Erreichen einer Stufe gewünscht, wie etwa ein Leaderboard oder eine Homepage mit der aktuellen Punkteanzahl und der momentanen Stufe, die zugleich die Privatsphäre der Studierenden schützt und beibehält.

Sheldon machte auch die Erfahrung, dass die Anwesenheit der Studierenden selbst an harschen Wintertagen überdurchschnittlich war: „Out of 700 possible absences (25 students x 28 class stations), there were only 10 absences – including the winter. In upstate New York. In neck-deep snow.“ (Sheldon, 2011, S. 116)

Durch die Punkte wurde die Transparenz der Benotung deutlich erhöht, wodurch klarer ersichtlich war, welche Aufgaben von großer und welche von weniger großer Bedeutung waren: „They felt they could 'game' the class and figure out where to put in the effort (and where they could slack a little bit) to get a good grade.“<sup>11</sup> Natürlich wird einem als Lehrender bei diesem Zitat etwas mulmig zumute, denn schließlich

---

<sup>11</sup> <http://gamingtheclassroom.wordpress.com/t366-multiplayer-game-design-post-mortem/>

möchte wohl niemand, dass nur bestimmte Aufgaben absolviert werden, im perfekten Idealfall soll die Lernenden alle Aufgaben mit Freude meistern, doch ich denke, dass dies sehr selten der Fall ist und bin der Meinung, dass diese selektive Lernen bereits vor Sheldons Konzept existierte und durch selbiges nicht unbedingt verstärkt wird. Zusätzlich wurde allem Anschein nach die Motivation dadurch erhöht, dass die Studierenden von sich aus lieber eine höhere Punktzahl erreichen, anstatt, wie im bisherigen System, eine schlechte Note vermeiden wollten: „Mr. Sheldon says that points system not only feels more like a video game, but also lets students feel like they’re earning points for getting things right instead of losing them for incorrect answers.“ (Laster, 2010) Damit wird auch ein Punkt angeschnitten, der bei vielen didaktischen Diskussionen im Laufe meines Studiums immer wieder angesprochen wurde, nämlich jener, dass Lehrende sehr stark auf Fehler fokussiert sind und womöglich zu wenig darauf, die positiven Errungenschaften der Lernenden hervor zu heben. Dieser strittige Punkt könnte durch eine Umstellung des Benotungssystems gemindert werden, da hierbei nun nicht mehr Fehler, sondern Errungenschaften und Erfolge gezählt werden. Anzumerken ist freilich, dass auch hier Fehler zu einem Abzug in den Gesamtpunkten einer Aufgabe führen und somit nicht das System per se ein Allheilmittel darstellt, doch es begünstigt zumindest eine Betrachtung der Leistung, die mehr auf Erfolge und Errungenschaften beruht. Denkbar wäre hier eine Weiterentwicklung des Ansatzes, bei dem zum Beispiel die Note oder Punkteanzahl eines Tests aus der Summe des Erreichten besteht und nicht aus der maximalen Punktzahl abzüglich der Fehler, was womöglich einen Mehraufwand bei der Testvorbereitung bedeutet, da bei jeder Aufgabe überlegt werden muss, wie viele Punkte für welchen Schritt vergeben werden, was aber auch dazu führen könnte, dass die Benotung etwas objektiviert wird.

Das Konzept der Avatare schien bei den Studierenden gut anzukommen, denn sie bemängelten, dass diese zu wenig genutzt wurden, da durch sie verstärkt ein kulturelles Eintauchen (orig.: „cultural immersion“<sup>12</sup>) statt fand. „It [Anm.: avatar] provides them with a sense of ownership and gives them a stake in the game [...]“ (Sheldon, 2011, S. 38). Es scheint für die Bindung an den künstlichen Avatar bei dessen Namensgebung wichtig zu sein, den Lernenden bis auf anstößige Wörter, alle

---

<sup>12</sup> <http://gamingtheclassroom.wordpress.com/t366-multiplayer-game-design-post-mortem/>

Freiheiten zu lassen, wodurch eine bunte Mixtur an Namen entsteht, die aber jeweils Ausdruck der Individualität der Lernenden sind (vgl. Sheldon, 2011, S. 38). Wie eng die Bindung an die virtuelle Präsenz ist, hängt zusätzlich noch von einigen anderen Faktoren, wie dem körperlichen Idealbild (je größer die Differenz zum Ideal, umso stärker die Bindung) oder auch psychologischen Aspekten, wobei hier eine geringere Differenz und mithin eine größere Ähnlichkeit die Intensität der Bindung steigert (vgl. Ducheneaut, Wen, Yee & Wadley, 2009). Das Spiel mit Avataren kann sogar so weit getrieben werden, dass es Auswirkungen auf das reale Verhalten von Menschen aufweist, wobei der ausschlaggebende Faktor scheinbar die Einbindung des Gesichts einer Person in die virtuelle Präsenz ist (vgl. Fox, Bailenson & Binney, 2009).

Die Gruppenorganisation in Form von Gilden fand anscheinend auch Anklang, da diese als eine Art Bündnis gesehen wurde, wodurch sich die Teilnehmer als „Teil von etwas“ empfanden (orig.: „a part of something“<sup>13</sup>). Das kompetitive Spiel zwischen den Gilden beziehungsweise Gruppen während der Vorbereitung zum Examen waren ebenfalls ein Pluspunkt, der anscheinend anspornend wirkte. Dabei bemerkte Sheldon eine Parallele zwischen dem Spiel und der Arbeitsweise seiner Studierenden, da sie ähnlich flexible Strategien und Hilfsmittel anwendeten, wobei auch hierdurch eine Leistungssteigerung zu dem üblichen System vermerkt wird.

Bei den Vorträgen erwies sich die Methode als motivierender, bei der nach jedem Abschnitt eine Frage zu diesem gestellt wurde, als jene, bei der die Fragen am Anfang gestellt wurden. Auch die methodische Umstrukturierung, dass die Studierenden statt dem Tutor vortragen, schien sich positiv auf die Aufmerksamkeit ausgewirkt zu haben. Eine Gilde trug in einer späteren Iteration des Konzeptes eine Präsentation in Form des TV-Formates „So You Want to Be a Millionaire?“ vor, hierzulande besser bekannt unter dem Titel „Wer wird Millionär?“. Durch diese Art des Vortrages, konnte der Inhalt scheinbar sehr gut erinnert werden: „On the midterm, the material covered by So You Want to Be a Millionaire? was nailed by almost everyone who had been present on the day of the game.“ (Sheldon, 2011, S. 74)

---

<sup>13</sup> URL: <http://gamingtheclassroom.wordpress.com/t366-multiplayer-game-design-post-mortem/>

Abschließend wird angemerkt, dass der Enthusiasmus der Studierenden mit dem neuen Konzept generell höher war, als mit der klassischen Methode und dass oftmals der Wunsch bestand, dieses Unterrichtskonzept auch in anderen Fächern anzuwenden:

*„Overall the students were uniformly enthusiastic about the class as game approach. Many wished that other of their courses could be taught the same way; and thought the techniques could be used with just about any subject matter.“<sup>14</sup>*

### **3.4 Reflexion & Anmerkungen**

*„We'll come back to that word, balance, again and again in the following pages.*

*One of the hardest jobs a game designer has  
is to balance the many elements of a game.“*

*-Lee Sheldon (2011, S. 16)*

Die Gewichtung und Verteilung der Punkte könnte etwas breiter gefächert oder besser kategorisiert werden, um allzu große und gewichtige Blöcke zu vermeiden. Das könnte dazu führen, dass die Leistung und Motivation etwas konstanter und weniger auf einzelne Termine fokussiert ist. Ganz klassisch wären hier als weniger gelungene Unterrichtsstile beispielsweise jene zu nennen, bei denen die Note zu einem Großteil aus einigen wenigen Tests besteht, da freilich verstärkt für diese Termine gelernt, die aktive Mitarbeit im normalen Unterricht aber eher vernachlässigt wird. Durch kleinere Tests, abwechslungsreichere Testmethoden beziehungsweise fortlaufende Überprüfung der Mitarbeit wären bereits einige Möglichkeiten gegeben, um die Lernenden etwas beständiger zu motivieren und zusätzlich wäre somit auch einer Überforderung durch eine Anhäufung von Tests und Schularbeiten in einem zeitlich zu engem Rahmen vorgebeugt.

---

<sup>14</sup> URL: <http://gamingtheclassroom.wordpress.com/t366-multiplayer-game-design-post-mortem/>

In einer schulischen Situation, beispielsweise in einer AHS sind mehrere Themenkomplexe über ein Semester hinweg zu bearbeiten. Hier würde es sich unter anderem anbieten, jeden dieser Abschnitte mit einem Test oder einer praktischen Ausarbeitung zu beschließen. Dadurch entstehen kleinere und logisch strukturierte Blöcke, die das Gefühl eines inkrementellen Fortschreitens fördern und es können zusätzlich verschiedene Methoden zur Benotung heran gezogen werden, sofern diese Überprüfungen nicht nur auf normalen Tests basieren. Es können auch Aufgaben mit Bonuspunkten verwendet werden, um die Motivation zu fördern und um die Möglichkeit zu bieten, dass schlechte Leistungen in einzelnen Abschnitten verbessert werden. Damit das Punktesystem seine volle Wirksamkeit bezüglich Transparenz entfalten kann, sollte der aktuelle Punktestand der Lernenden zu jeder Zeit einsehbar sein, was entweder mittels einer Lernplattform oder einer anonymen Punktetabelle realisiert werden könnte: „Incremental rewards are terribly important for keeping players engaged.“ (Sheldon, 2011, S. 43)

Es könnte auch ein weiterer Punkt, der für den Erfolg von Spielen herausragend ist, in ein Klassenzimmer übernommen werden: Spiele bieten die Möglichkeit, durch oftmaliges Versuchen einen Lernerfolg zu erreichen. Hierbei besteht selten ein Leistungsdruck, ähnlich einem Test oder einer Prüfung, und man hat die Möglichkeit, Schritte zu wiederholen, erneut zu versuchen und zu variieren. Gibt es nun viele verschieden Test- und Prüfungsmodalitäten über ein Semester verstreut, so könnte auch das einen positiven Effekt auf die Verminderung des Leistungsdrucks haben. Sheldon meint hierzu: „In an MMO, there is an unlimited supply of repeatable tasks to guarantee that players can always level up. In an MC, there must be enough assignments to replicate that aspect of gameplay (many opportunities for XP) and rewards (many opportunities for incremental rewards).“ (Sheldon, 2011, S. 58) Im Multiplayer-Classroom (kurz: MC) soll also genau das gewährleistet sein: die Möglichkeit, die eigene Leistung oft und auf unterschiedliche Weise unter Beweis stellen zu können.

Wie bereits erwähnt, könnte das Punktesystem auch dahin gehend ausgebaut werden, dass zur Berechnung der Note die Summe der erfolgreich absolvierten Schritte und nicht der Abzug von Punkten für Fehler heran gezogen wird. Auch wenn dies eher ein kosmetischer Punkt zu sein scheint, könnte er eventuell, vor allem bei schwächeren Lernenden, etwas mehr das Vertrauen in die eigenen Fähigkeiten steigern und vielleicht auch besser vermeiden, dass Lernende resignieren, da ihnen weniger oft vor Augen geführt wird, was sie alles nicht können, sondern das, was sie geleistet haben, augenscheinlicher hervor gehoben wird. Unter Umständen könnte es dadurch auch gelingen, dass Leistungsdruck und Stress ein klein wenig gemindert werden. Dies kann vor allem durch die Kommunikation des Unterrichtskonzeptes vonseiten der Lehrenden unterstützt und gefördert werden.

Vor Tests könnte der Prüfungsstoff nochmals aufgearbeitet werden, etwa in einem Frage-Antwort Spiel, wie zum Beispiel „Wer wird Millionär?“ oder durch andere anspornende oder kompetitive Unterrichtsmethoden. Dabei könnten die Lernenden Zusatzpunkte erhalten, falls die Vorbereitung als Bonus-Konzept und nicht als fixer Bestandteil der Unterrichts und der Benotung betrachtet wird.

Die Gilden und somit die Organisation von längerfristigen Gruppen, wurde anscheinend mit Wohlwollen seitens der Studierenden aufgenommen. Ein Problem, das sich daraus ergeben könnte, wäre, dass schwächere Lernende sich eher bedeckt halten und weniger aktiv mitarbeiten. Eine Methode, dem entgegen zu wirken, wird von Sheldon auf seinem Blog unter dem Abschnitt Syllabus<sup>15</sup> bereits angeführt, nämlich die anonyme Vergabe von Extra Punkten („extra credits“) in einem Peer-Review innerhalb der Gruppe für die jeweiligen Mitglieder (vgl. Sheldon, 2011, S. 40). Diese Methode könnte aber durch das soziale Gefüge und persönliche oder private Konflikte korrumpiert werden, also entweder Absprachen, dass alle die vollen Punkte erhalten, oder zerstrittene Parteien, die nicht ihre Arbeit, sondern ihre Beziehung zueinander bewerten, wodurch sie eher mit Bedacht eingesetzt werden sollte. Auch für Sheldon waren diese Faktoren von Interesse, doch scheinbar kam es zu keinen Absprachen in den Gilden: „In no guilds in the class did everyone earn 100 pts.“ (Sheldon, 2011, S. 40)

---

<sup>15</sup> URL: <http://gamingtheclassroom.wordpress.com/syllabus/>

Eine klare Rollenverteilung innerhalb der Gruppe wäre ebenfalls eine Strategie, um die einzelnen Leistungen etwas besser überprüfen zu können und zu unterbinden, dass sich einige Lernende vor der Arbeit drücken. Sheldon hat hierfür innerhalb der Gruppen Rollen vergeben, etwa Begriffe aus dem Berufsumfeld eines Spieledesigners wie „Designer“, „Writer“ und dergleichen (vgl. Sheldon, 2011, S. 25) oder fantasievollere Rollen aus den Spieluniversen, wie „Mages“, „Rangers“ etc. (vgl. Sheldon, 2011, S. 100ff).

Zum anderen könnten die Gilden semi-konstant organisiert werden, sodass eine Gilde nur für ein Projekt besteht und danach neue Gilden formiert werden, was auch noch den Vorteil bieten würde, dass es zu einer Erweiterung des sozialen Gefüges der Lernenden kommt, da sie nicht immer in der selben Gruppe arbeiten, sondern sich auch vermehrt mit denjenigen auseinander setzen, mit denen sie andernfalls weniger Kontakt haben, andererseits würde aber die soziale Bindung an die Gilde nicht so stark ausgebaut werden können, wenn diese nur ein temporäres Bündnis darstellt.

Einen anderen Lösungsweg gab es noch von Aaron Pavao, einem Mathematik und EDV Lehrer an der Waunakee Community High School: „[I] implemented a 'teamwork bonus' system wherein if all of the members of a team do well on an individual test, they get an XP bonus. This leverages social pressure to encourage students to encourage one another to prepare.“ (Pavao zit. nach Sheldon, 2011, S. 117). Hier wird versucht über die soziale Komponente die Motivation zu steigern, was durch einen Bonus an Erfahrungspunkten erreicht wird, den eine Gilde erhält, wenn alle Mitglieder ein gutes Testresultat abliefern. Die soziale Komponente bei Sheldons Konzept spielt wohl durchaus eine tragende Rolle, die auch von Spielern von „World of Warcraft“ immer wieder betont wird. In dem Seminar „didaktisches Design“, das ich im Sommersemester 2010 besucht hatte, präsentierte ich Untersuchungen betreffend der Auswirkungen der Ausrüstungsgegenstände eines Charakters auf dessen Attribute, was prompt eine Diskussion auslöste, in der Spieler sehr deutlich betonten und darin auch überein stimmten, dass weniger die Ausrüstung, als vielmehr die Bindung an die Gruppe, eine wesentliche Motivation

zum Spielen war. Die sozialen Faktoren im Spiel scheinen tatsächlich eine tragende Rolle bei der Motivation zu sein, was auch empirisch unterstrichen werden kann (siehe hierzu Cypra, 2005, S. 90 ff.). Auch die Betrachtung sportlicher Aktivitäten kann dies nochmals untermauern, da hier oftmals entweder der kompetitive Faktor, die Bindung an ein Team oder sogar beides die Leistung des Einzelnen steigern kann.

Betreffend verwendeter Sprache und der Einbettung in eine virtuelle Fantasiewelt: Die Bereiche, die mir als angehenden Lehrer etwas seltsam und ungewohnt anmuten, sind die Sprache und die erschaffene Welt, auch wenn dies Faktoren sind, welche der Gestaltung des Unterrichts als Rollenspiel förderlich sein und die Motivation der Lernenden steigern können. Für einen erfahrenen Geschichtenerzähler wie Sheldon, der weiß, wie er sein Publikum fesselt und der ein Gespür für jene Mechaniken besitzt, die in der Lage sind, Menschen in eine fiktionale Welt eintauchen zu lassen, ist es wohl etwas natürlicher und naheliegender, sowie wahrscheinlich auch ein Stück weit unproblematischer, eine solche umrahmende Welt zu kreieren. Diese muss nicht zwingend eine Online-Welt sein, er sie für spätere Iterationen seines Multiplayer-Classrooms verwendete, sondern kann auch im Stil eines „Pen & Paper“ Rollenspiels verwirklicht werden, bei denen ein Erzähler die Heldengruppe durch eine abenteuerliche Welt geleitet. Ein Beispiel für eine sehr kompakte Form der Rollenspielmethode gibt es von Keller et al. aus ihrem Kurs zur Einführung in die Kryptografie, wobei folgende Aufgabe im Stil eines Rollenspiels präsentiert wurde: „The Spartans want to send a secret message from Sparta to Athens. In this message, a strategy for the upcoming battle is described. The battle begins in 3 days.“ (Keller, Scheuner, Serafini & Steffen, 2011, S. 194) Das Rollenspiel ist hier nur in sehr elementarer Form verwendet worden und könnte sehr einfach erweitert und ausgebaut werden, indem der Kurs möglicherweise als ein Wettstreit zwischen zwei Teams organisiert oder die umrahmende Geschichte vertieft wird, um die emotionale Aktivierung zu erhöhen und somit das Gefühl der Involvierung zu steigern. Je reichhaltiger und breiter diese fiktionale Welt jedoch ausgebaut wird, umso tiefer kann das Eintauchen der Lernenden in selbige vonstatten gehen, was die Bindung an das Lernspiel erhöht: „These rich virtual worlds are what make games such powerful contexts for learning.“ (Shaffer et al., 2004, S. 4)

Ein Konzept zu verwenden mit dem man sich selbst nicht wohl fühlt, ist jedoch nicht die Intention meiner Reflexion. Es gibt meiner Ansicht nach genug Methoden und Mechaniken, die Sheldon verwendet, sodass die Einbettung des Unterrichts in eine virtuelle oder erzählte Welt auch als optionales Modul angesehen werden kann, das zwar, wenn es stimmig umgesetzt wird, wohl noch etwas mehr Anreize bietet, das aber nicht zwingend notwendig für ein funktionierendes Konzept nach Sheldon ist. Für all jene, die weniger vertraut mit den virtuellen Welten von Spielern und sozialen Netzwerken sind, aber dennoch Gefallen an einigen Umsetzungen finden, möchte ich nachfolgend noch versuchen, das Konzept von Lee Sheldon in ein modulares, didaktisches Baukastenmodell zu extrahieren, damit genau jene Elemente davon verwendet werden können, die einem zusagen, wobei aber gleichzeitig kein Zwang besteht Methoden zu verwenden und zu implementieren, mit denen man sich selbst nicht anfreunden kann.

### 3.5 Didaktische Analyse

*„The structure of the course allows the students  
to take ownership of their learning.“*

*-Marie-Pierre Huguet (zit. nach Sheldon, 2011, S. 128)*

Marie-Pierre Huguet, eine ehemalige Lehrerin, die am Rensselaer Polytechnic Institute die Stelle als „project leader for course support of distributed education courses, and consultant to faculty on issues of technology integration, blended learning, and instructional design“<sup>16</sup> bekleidet, konnte den Unterricht von Sheldon beobachten und ihn aus didaktischer Sicht ein wenig durchleuchten (vgl. Sheldon, 2011, S. 125 ff). Ein Satz, der mir mit Kersten Reichs konstruktivistischer Theorie im Hinterkopf ins Auge gestochen ist, war folgender: „Piaget is omnipresent in the environment Lee has created. Lee intuitively understands the critical role that experiences play in his students' learning. He is aware that they construct their own knowledge in response to these experiences.“ (Marie-Pierre Huguet zit. nach Sheldon, 2011, S. 127) Es erinnerte mich an die Worte von Reich: „Selbst erfahren, ausprobieren, experimentieren, immer in eigene Konstruktionen ideeller oder

---

<sup>16</sup> Quelle: <http://www.rpi.edu/coursedevelopers/MPH.htm> [zuletzt abgerufen am 08. November 2012]

materieller Art überführen und in den Bedeutungen für die individuellen Interessens-, Motivations- und Gefühlslagen thematisieren.“ (Reich, 2010, S. 119) Die Grundlage von Sheldons Konzept ist allem Anschein nach eine sehr konstruktivistisch orientierte Theorie, die er sehr intuitiv verwendet. Explizit erwähnt wird dies jedoch erst von Marie-Pierre Huguët in ihren Beobachtungen. Durch das didaktische Design in Sheldons „Multiplayer Classroom“ wird nach Huguëts Ansicht ein weiterer wichtiger Punkt für den Erfolg von Unterricht in konstruktivistischer Sicht, nämlich die Selbsttätigkeit der Lernenden und ihre aktive Auseinandersetzung mit dem Lehrstoff, erreicht. Sheldon selbst hält sich eher im Hintergrund und fungiert als eine Art „Game Master“ (vgl. Sheldon, 2011, S. 36). Dadurch lässt er den Studierenden mehr Freiraum für ihre Aktivitäten und erreicht, dass sie selbst eine aktivere Rolle einnehmen, da sie sich das Wissen sehr häufig selbstständig oder in einer Gruppe erarbeiten müssen.

Der Großteil der Methoden, die verwendet werden, lassen sich auch im konstruktivistischen Methodenpool<sup>17</sup> von Kersten Reich wiederfinden und nachlesen. Am augenscheinlichsten ist wohl die Verwendung des Rollenspiels als Umrahmung für den Unterricht, sowie der häufige Einsatz von Gruppenarbeiten, die sich beide wie ein roter Faden durch das Konzept von Sheldon ziehen. Zusätzlich werden von ihm sehr gerne Methoden wie Quiz bzw. das konstruktive Wissensspiel mit teils kompetitivem Charakter (eine Adaption des Showformates „Wer wird Millionär?“), Projektarbeit, Referat und Präsentation sowie in neueren Iterationen auch Elemente von Fantasiereise, Web-Tools oder auch Stationenlernen verwendet. Allerdings möchte ich an dieser Stelle auch anmerken, dass die Vergabe von Punkten einen eher behavioristischen Charakter aufweist und an die klassische Konditionierung nach Pawlow erinnert (vgl. hierzu Herkner, 2001, S.49 ff). Auch wenn die Konditionierung im heutigen Diskurs als überholt gilt, so ist sie zum Beispiel hinsichtlich Lernmotivation ein nützliches Werkzeug und wird von Helmke in dieser Hinsicht sogar als „unersetzbar“ (2010, S. 60) beschrieben.

Durch die bereits erwähnten Punkte lässt sich der Erfolg des Konzeptes sehr gut entschlüsseln und mithin auch ein wenig demystifizieren. So basiert der Kern sehr

---

<sup>17</sup> Zu finden unter: [http://www.uni-koeln.de/hf/konstrukt/didaktik/frameset\\_uebersicht.htm](http://www.uni-koeln.de/hf/konstrukt/didaktik/frameset_uebersicht.htm)

stark auf einem handlungsorientierten Konzept von Unterricht, das sich gut mit dem theoretischen Ansatz von Kersten Reich vereinbaren lässt. Daher besteht auch die Möglichkeit die wesentlichen Elemente des zunächst etwas fantastisch anmutenden Unterrichtskonzepts zu extrahieren und für eine breitere Basis an Lehrveranstaltungen nutzbar zu machen. Für all jene, denen das Rollenspiel-Setting und dessen Terminologie seltsam anmuten mag und denen dessen Implementierung in den Unterricht Unbehagen bereitet, kann es dadurch auch durchaus weggelassen werden. Auch wenn die komplette Adaptierung des Konzepts auf einmal ein zu gewagter Schritt ist, kann sich langsam heran tasten, indem zum Beispiel erst der Fokus auf die konstruktivistischen Methoden übernommen wird und dann zusätzlich, in einer weiteren Iteration, die transparente Benotung in Form von Punkten, wobei auch hier auf die Bezeichnung von Sheldon mit Begriffen wie „Erfahrungspunkten“, „experience points“ oder kurz „XP“ verzichtet werden kann. Ein für mich sehr wesentlicher Punkt an dem Konzept ist jedoch, dass es stets zur Diskussion mit den Lernenden offen steht, dass es am Schluss oder auch zwischendurch ein Feedback oder ein Resümee gibt, sodass es stets weiter entwickelt und verbessert bzw. an die Bedürfnisse und Wünsche der Zielgruppe angepasst werden kann. Allein schon dieser Schritt könnte zu einer Verbesserung des Unterrichts führen. Interessanterweise existiert diese Form der Rückmeldung bereits an den Universitäten, zumindest betreffend der Universität Wien und der Technischen Universität Wien weiß ich aus eigener Erfahrung davon, dass für Lehrveranstaltungen regelmäßig das Feedback der Studierenden eingeholt wird, doch im Schulsystem dürfte hier noch einiger Aufholbedarf bestehen. Effiziente und zweckdienliche Verbesserungen können nur dort erreicht werden, wo Klarheit darüber herrscht, was gewünscht und benötigt wird. Hier könnte jeder Lehrende selbst ein wenig der breiten, von Kersten Reich geforderten, demokratischen Basis in den Unterricht einfließen lassen. Mit Blick Richtung Pisa-Studie und Optimierung des Bildungssystems könnten die hier gewonnenen Daten ebenfalls sehr nützlich sein, da die Defizite des Systems direkt an der Wurzel erhoben werden können, wodurch sich die Anpassungen womöglich etwas direkter an den Bedürfnissen der eigentlichen Zielgruppe orientieren könnten. Außerdem könnte es vorbeugend gegen ein Festfahren in einem bestimmten Unterrichtsstil wirken und für ein wenig mehr Flexibilität und Kommunikation zwischen den Parteien der Lehrenden und Lernenden sorgen.

Abschließend und zusammenfassend möchte ich die wesentlichen Kernelemente des Unterrichtskonzepts von Lee Sheldon auflisten, damit es eine übersichtliche Unterteilung in die wichtigsten Themenbereiche oder Module gibt, die je nach Wunsch und Bedarf vereinzelt oder in Kombination implementiert werden können.

- Benotung mittels Punkten
- konstruktivistische Unterrichtsmethoden
- handlungsorientierter Unterricht
- Kommunikation, Feedback & Resümee (der Lernenden)
- Setting (Rollenspiel, Terminologie)
- soziale Katalysatoren (Gruppenarbeiten, kompetitive Gruppenaufgaben)

*„If nothing else, the multiplayer classroom will appeal to every miserly school system in the land. It uses the language and principles of video games to engage student with little to no out-of-pocket costs at all.“ (Sheldon, 2011, S. 18)*

## 4 DIDAKTISCHE SPIELE

### 4.1 Abwägung der Alternativen

Vor Beginn meiner Ausarbeitungen, wurden einige Möglichkeiten didaktischer Programmierumgebungen geprüft, wobei zwei in die engere Auswahl kamen, nämlich Lego Mindstorms NXT und BYOB, welches eine Erweiterung von Scratch ist. Ein Knock-Out Kriterium war dabei die Verwendung einer grafischen Umgebung, da hierdurch der Zugang zur Programmierung schon einigermaßen vereinfacht wird: „Hence we propose visual programming environments than can be understood and mastered easily and quickly“ (Lonati et al, 2011, S. 221). Beide Konkurrenten bieten einen intuitiven Zugang zur Programmierung und nutzen grafische Repräsentationen zur Darstellung von Befehlen. Die Programmierung erfolgt dabei nicht durch Tippen, sondern durch das Aneinanderreihen von Programmblöcken, wodurch sehr schnell erste Erfolge erzielt werden können, was einen durchaus positiven Effekt auf die Motivation der Lernenden hat. Ebenso ist beiden gleich, dass sie mitunter auch auf didaktische Zwecke hin entworfen wurden und interessanterweise wurzeln beide in der Forschungsarbeit des MIT Media Lab, das nicht nur Scratch<sup>18</sup>, die Grundlage für BYOB, sondern auch den Grundstein für Legos Mindstorms NXT (mit-)entwickelt hat<sup>19</sup>.

Aus der Sicht von jemandem, der bereits einige Jahre programmiert hat, können sich teilweise Hürden beim Einstieg in die beiden Programme bieten, da die Zugänge zu manchen Algorithmen, vor allem bei Lego Mindstorms NXT eher gewöhnungsbedürftig sind. So hatte ich bei meinen ersten Gehversuchen mit den Robotern aus dem Hause Lego so manches Problem bei der Umsetzung einer wiederholten Bewegung mittels einer Schleife, was sich aber durchaus mit der Usability des Programmeditors begründen lässt und womöglich weniger mit einer zu abstrakten Programmierstruktur. Bezüglich Aufbau hat BYOB aber den großen Vorteil, dass es sich viel näher an üblichen Algorithmen orientiert, als die Konkurrenz und somit der

---

<sup>18</sup> Quelle: <http://ilk.media.mit.edu/projects.php?id=783> [zuletzt abgerufen am 08. November 2012]

<sup>19</sup> Quelle: <http://www.media.mit.edu/sponsorship/getting-value/collaborations/mindstorms>  
[zuletzt abgerufen am 08. November 2012]

Wechsel auf eine normale Programmier- oder Skriptsprache wesentlich reibungsloser und fließender vonstatten geht. Auf der anderen Seite kann Mindstorms NXT mit der Möglichkeit punkten, dass es per Live-CD mittels Java programmiert werden kann, wodurch es sich auch für das Erlernen fortgeschrittener Programmierung bestens eignet, wohingegen die Blöcke in BYOB momentan mittels Squeak, einer auf Smalltalk basierenden Programmiersprache, bearbeitet werden können, die mir als weiterführende Sprache weniger geeignet scheint, wobei sich das in Zukunft wahrscheinlich ändern wird, da die Entwickler von BYOB das Projekt ab Version 4.0 in „Snap!“ umbenennen werden und in diesem Zuge auch HTML5 und JavaScript als Basis verwenden wollen, was sich meiner Meinung nach zur Überführung in eine praxisnahe Programmierung sehr gut eignen kann. Für Lernende, die noch keine oder kaum Erfahrung mit Algorithmen, Datenstrukturen oder Programmierung haben, bietet BYOB zusätzlich eine wesentlich breitere Basis, als Mindstorms NXT, da es eine Vielzahl an Begriffen und Strukturen mitbringt, die für eine spätere weiterführende Ausbildung ein sehr vorteilhaftes Grundgerüst bilden, was wahrscheinlich auch durch den Umstand begünstigt wird, dass es nicht wie Mindstorms NXT eine recht enge Bindung an die Steuerung eines Roboters hat, und sich daher besser an den Bedürfnisse von Anfängern orientieren kann.

Ein weiterer wichtiger Punkt sind natürlich die Kosten der beiden Systeme, da sie schließlich im Bildungssystem eingesetzt werden sollen, in dem nicht immer die gewünschten finanziellen Mittel vorhanden sind. Hier kann ganz klar BYOB punkten, da es gratis als Open-Source Software zur Verfügung steht, wohingegen Mindstorms NXT mit satten €299<sup>20</sup> zu Buche schlägt. Auch der zeitliche Aufwand für den Auf- und Abbau ist bei Mindstorms höher, als bei BYOB, das lediglich einen PC benötigt. Ein weiterer Vorteil von BYOB ist, dass es sowohl für Windows, Linux als auch Macintosh Plattformen verfügbar ist. Mindstorms NXT gibt es zwar für Windows und Macintosh, allerdings lief es in meinen Versuchen lediglich unter Windows einwandfrei, denn unter Apples OS X kam es trotz Patch zu Problemen. Da die nächste Version von BYOB (Namenswechsel auf Snap! 4.0) mittels JavaScript und HTML5 umgesetzt werden soll, ist auch zukünftig eine Plattformunabhängigkeit

---

<sup>20</sup> Quelle: <http://shop.lego.com/de-AT/LEGO-MINDSTORMS-NXT-2-0-8547?p=8547>

[letzter Zugriff am 08. November 2012]

sicher gestellt, wenn sie nicht sogar in Hinblick auf Smartphones und Tablets noch erweitert wird. Nach den Angaben der Entwickler ist die nächste Version zwar noch im Pre-Alpha Status (recherchiert im September 2012), doch wer möchte kann sich auf der Projektseite bereits jetzt einen Einblick verschaffen<sup>21</sup>.

Bei den didaktischen Möglichkeiten zur Aneignung der grundlegenden Prinzipien von Programmierung bietet BYOB im Gegensatz zu Mindstorms NXT angesichts der formalen Struktur eine größere Nähe zu dem gewohnten Aufbau eines Programms, wobei schon der Faktor der Ausrichtung der Befehle, die bei BYOB in üblicher Weise vertikal ist, bei Mindstorms NXT jedoch horizontal und mehr einer Zeitlinie, vergleichbar mit einem Videoschnittprogramm, ähnelt, eine meiner Meinung nach gelungenere Umsetzung darstellt. Durch das Wegfallen eines zu strengen Fokus auf die Steuerung von Hardware, kann BYOB gezielter auf didaktische Bedürfnisse eingehen und sich mehr Freiraum für eine konkretere grafische Repräsentation einräumen. Bei der grafischen Umsetzung der Baustein-Metapher ist BYOB daher sehr nahe an der formalen Struktur etlicher Programmiersprachen, wohingegen Mindstorms NXT eher an Bauklötze, als an Bausteine erinnert, und der strukturelle Aufbau dadurch eine größere Distanz zu gewohnter Programmierung aufweist. Gerade bei der Betrachtung der syntaktischen Umsetzung beider Lernumgebungen, zeigt sich der Vorteil von BYOB in dieser Hinsicht, denn durch die unterschiedlichen Formen der Blöcke passen nur jeweils solche Bausteine in- bzw. aneinander, wie es auch die Syntax der meisten nicht-grafischen Programmiersprachen zulassen würde. Somit kann von Beginn an eine korrekte Gliederung von Programmstrukturen erlernt und eingeübt werden, was den Umstieg auf andere Sprachen erleichtert. Mich beschäftigte einige Zeit recht intensiv die Frage, ob es in BYOB wirklich zweckdienlich ist, dass die Unterscheidung von Variablentypen fehlt, wodurch es unter anderem möglich ist, dass Strings auch problemlos durch die Operatoren „<“ und „>“ vergleichbar sind. Einerseits ist es eine nachvollziehbare Vereinfachung, da ein Vergleich auf der Ebene der ASCII Kodierung keine große Schwierigkeit darstellt, andererseits sollte hierbei von Lehrenden bedacht werden, dass für Anfänger und Anfängerinnen die Verknüpfung von logischen Operatoren mit Buchstaben und Wörtern unverständlich erscheinen

---

<sup>21</sup> zu finden unter: <http://byob.berkeley.edu/#snap4.0> [letzter Zugriff am 08. November 2012]

mag, auch wenn es aus Sicht der Entwickler ein logisch nachvollziehbarer Schritt ist, vor allem, wenn BYOB nicht Selbstzweck, sondern eine Vorbereitung auf eine Programmiersprache sein soll, für die eine Unterscheidung der Typen von Variablen von Bedeutung ist. Da ich nach reiflicher Überlegung aber zu keiner vernünftig umsetzbaren Lösung gekommen bin, die außerdem noch im Stande ist die Intuitivität der Lernumgebung nicht zu gefährden, halte ich die konsequente Einhaltung des Prinzips „keep it simple“ für durchwegs zweckdienlich.

Wirklich überzeugt hat mich letztendlich aber erst die Möglichkeit auch fortgeschrittene Algorithmen und Datenstrukturen in BYOB programmieren zu können, was bis hin zur Entwicklung eigener Programmiersprachen reicht, wie etwa Block-basierte Version von „C“ (siehe hierzu Federici, 2011, S. 192 ff.). So war es mir nach dem Ausmerzen einiger anfänglicher Hürden möglich, einen Heapsort Algorithmus zu realisieren, den ich selbst erst zu Beginn meines Studiums auf der Technischen Universität Wien kennen lernte. An manchen Stellen musste der Algorithmus zwar leicht an die Mitteln von BYOB angepasst werden, da es zum Beispiel nicht mit For- oder While-Schleifen dienen kann, sondern einzig eine Repeat-Schleife bietet und obendrein bei den Indizes von Listen nicht wie in der Informatik üblich bei 0, sondern bei 1 zu zählen beginnt. Da jedoch die For-Schleife selbst nachgebaut werden kann und der Zugriff auf die Liste mit einem simplen Inkrement des verwendeten Index auch das erwünschte Element liefert, waren diese anfänglichen Probleme beseitigt, doch dazu später mehr. Ich muss an dieser Stelle gestehen, dass ich selbst anfangs die breit gefächerten Einsatzmöglichkeiten von BYOB etwas unterschätzt hatte.

## **4.2 Kurzvorstellung**

Einleitend möchte ich kurz BYOB ein wenig näher vorstellen, damit besser nachvollzogen werden kann, wie es funktioniert und wie damit gearbeitet werden kann. Die nachfolgende Abbildung zeigt die Oberfläche von BYOB, die sehr übersichtlich und schlicht gehalten ist.

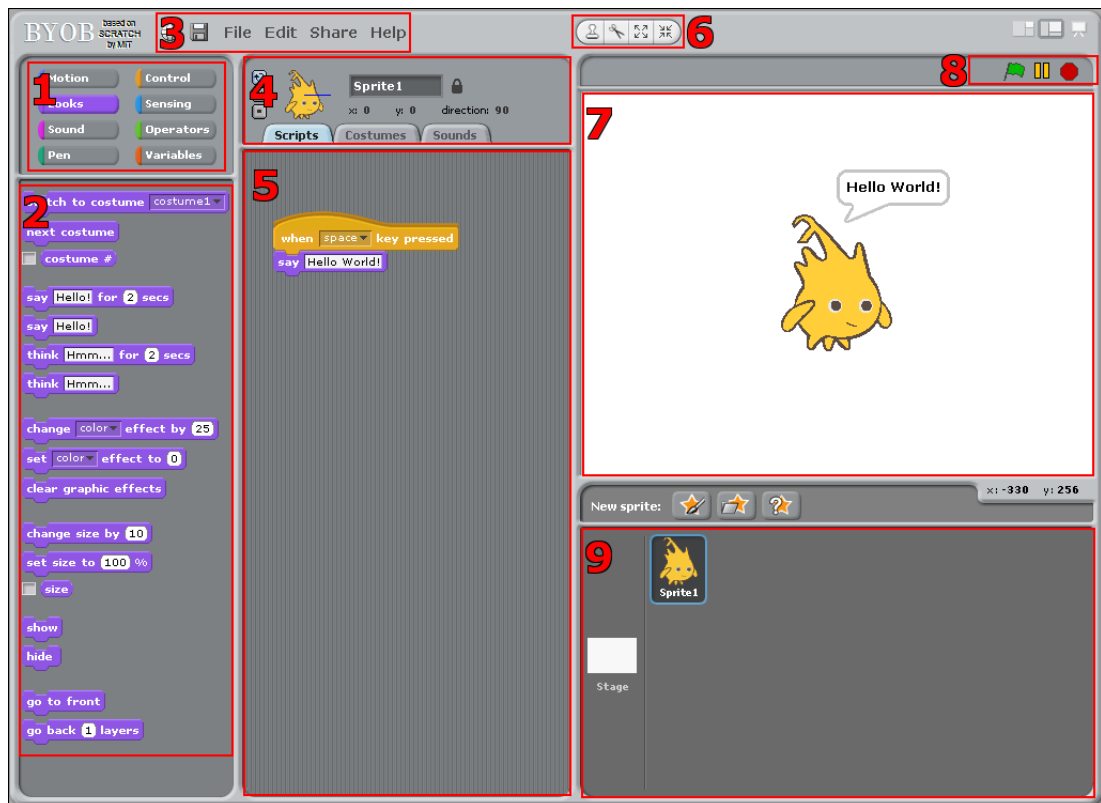


Abbildung 1: Das Programmfenster von BYOB

- |                       |                        |                               |
|-----------------------|------------------------|-------------------------------|
| 1 ... Blockkategorien | 4 ... aktueller Sprite | 7 ... Stage                   |
| 2 ... Blöcke          | 5 ... Skriptbereich    | 8 ... Start/Pause/Stop Events |
| 3 ... Menü            | 6 ... Toolbar          | 9 ... Sprite-Liste            |

In der obersten Leiste findet sich zunächst das Programm-Menü (3) und die Toolbar zur manuellen Bearbeitung der Sprites (6). Weiters ist das Fenster im Wesentlichen in drei Bereiche aufgeteilt, welche in Spalten organisiert sind. Links findet sich das Menü für die Blöcke, das unterteilt ist in die Kategorien (1) und die dazugehörige Liste der Blöcke (2). In der mittleren Spalte werden zunächst Details zum momentan ausgewählten Sprite angezeigt (4), wie etwa der Name, Blickrichtung, X- und Y-Koordinaten der aktuellen Position, sowie drei Reiter mit denen zwischen den Skripts, Kostümen und Sounds gewechselt werden kann. Darunter findet sich der Anzeigebereich für den ausgewählten Reiter, der aktuell den Skriptbereich (5) darstellt, auf dem sich das „Hello World!“ Programm befindet, das im Detail wie folgt aussieht:



**Abbildung 2: "Hello World!" in BYOB**

Zum Starten des Programms wurde hier das Drücken der Leertaste gewählt, wobei aber auch andere Tasten oder wahlweise die grüne Flagge (8) zum Start des Programms möglich sind. In der rechten Spalte finden sich die Kontrollen zum Ablauf des Programms (8), die von links nach rechts Start (falls die grüne Flagge als startender Event verwendet wird), Pause und Stop (das alle Methoden und Skripts beendet) darstellen. Unterhalb findet sich die „Stage“ (7), auf der die verwendeten Sprites dargestellt werden und ihre Befehle ausführen. Auf der Abbildung ist der standardmäßig vorgegebene Sprite zu sehen, wie er nach einem Druck auf die Leertaste das erwünschte „Hello World!“ ausgibt. Im rechten unteren Eck findet sich schließlich noch die Liste zur Auswahl der Sprites.

Die Programmierung ist denkbar einfach und funktioniert durch Drag & Drop eines Blocks aus dem Untermenü (2) in den Skriptbereich (5) des gewünschten Sprites. Durch die unterschiedlichen Formen können nur solche Blöcke in- oder aneinander gefügt werden, die auch tatsächlich miteinander funktionieren und ausgeführt werden können, wodurch verhindert wird, dass gewisse Kombinationen zu syntaktischen Fehlern führen. Für versierte und fingerflinke Programmierer ist diese Art des Arbeitens wahrscheinlich nicht sehr erstrebenswert, da das ständige Klicken und Ziehen der Blöcke langsamer von der Hand geht, als das Tippen per Tastatur, die gerade durch Shortcuts ein sehr mächtiges Instrument zur Eingabe darstellt. Wenn es um den Umbau oder die Ausbesserung von etwas größeren Programmen geht, kann der Arbeitsfluss von BYOB dadurch schnell mühsam werden. Andererseits ist es für Anfängerinnen und Anfänger wohl wesentlich weniger unvorteilhaft, wenn nicht sogar förderlicher, da syntaktische Tippfehler von vorne herein auf breiter Basis ausgeschlossen werden.

## 4.3 Ausarbeitungen

### 4.3.1 BYOB & Heapsort

Zu Beginn der Präsentation meiner Ausarbeitungen möchte ich ein Projekt vorstellen, das für mich von großem Interesse war, nämlich die Realisierung des Heapsort Algorithmus mit BYOB. Das Programm fällt zwar nicht in die Kategorie der didaktischen Spiele, allerdings möchte ich es dennoch nicht unerwähnt lassen, da es sehr anschaulich illustriert, was BYOB zu leisten im Stande ist, wo Abstriche gemacht werden mussten und Anpassungen nötig waren. Der Algorithmus, der mir als Vorlage diente, stammt von Hans Werner Lang (2006, S. 19) und wurde auch auf der Homepage der FH Flensburg<sup>22</sup> veröffentlicht.

In der Klasse des Algorithmus der FH Flensburg werden zu Beginn zwei globale Variablen deklariert, nämlich ein Integer-Array „a“ und ein Integer „n“. Dies kann in BYOB durch das Erstellen einer Liste „a“ und einer Variablen „n“ gelöst werden.

```
public class HeapSorter
{
    private int[] a;
    private int n;
}
```

**Text 1: Variablen in Java**



**Abbildung 3: Variablen in BYOB**

Die einzige Methode der Klasse, die als public definiert wurde, ist „sort()“, an die auch das Array beim Aufruf als Parameter übergeben wird. Die zuvor deklarierten Variablen werden hier nun mit den Werten initialisiert und zwar mit dem übergebenem Array und dessen Länge, woraufhin der eigentliche Heapsort Algorithmus gestartet wird. In BYOB musste lediglich die Länge der Liste in der Variablen „n“ gespeichert und der Block „heapsort“ aufgerufen werden.

<sup>22</sup> Siehe <http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/heap/heap.htm> [letzter Zugriff am 08. November 2012]

```

public void sort(int[] a)
{
    this.a=a;
    n=a.length;
    heapsort();
}

```

**Text 2: sort()**



Abbildung 4: sort() in BYOB

Im Anschluss daran wird zuerst der Heap mittels der Methode „buildheap()“ erzeugt, der dann abschließend sortiert werden kann. Eine sehr zentrale Rolle spielt hierbei die Methode „downheap()“, denn ihre Aufgabe ist es, ein Element zu versickern, also die Heap Eigenschaft wieder herzustellen und das größte Element an die Spitze des Baumes und mithin an die vorderste Stelle des Arrays (Java) bzw. der Liste (BYOB) zu bringen. Doch zunächst einmal zur Methode „heapsort()“.

```

private void heapsort()
{
    buildheap();
    while (n>1)
    {
        n--;
        exchange (0, n);
        downheap (0);
    }
}

```

**Text 3: Heapsort in Java**



Abbildung 5: heapsort in BYOB

Hier zeigen sich bereits einige notwendige Anpassungen, die durch die Indizes der Listen in BYOB, welche bei 1 anstatt wie die Indizes der Arrays bei 0 beginnen, sowie durch das Fehlen einer „while“-Schleife zustande kommen, die sich aber anstandslos durch eine „For“-Schleife ersetzen ließ, welche aber für ein besseres Verständnis problemlos nachgebaut werden hätte können. Für die „For“-Schleife der Methode „buildheap()“ wurde dies zum Zweck der Illustration auch bewerkstelligt. Auch hier kommt durch den Unterschied bei der Indizierung (+1 bei BYOB) erneut zu einer Anpassung im Programmcode.

```
private void buildheap()
{
    for (int v=n/2-1; v>=0; v--)
        downheap (v);
}
```

**Text 4: buildheap in Java**

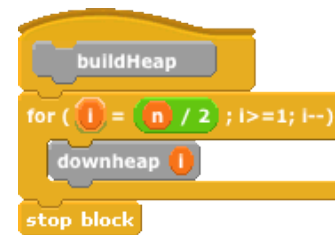


Abbildung 6: buildHeap in BYOB

Der Algorithmus für die Schleife wurde über einen kleinen Umweg mittels einer „Repeat“-Schleife realisiert und sieht wie folgt aus<sup>23</sup>:

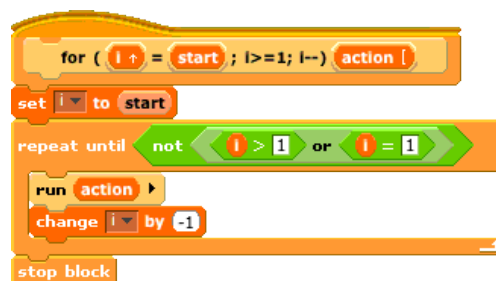


Abbildung 7: For-Schleife mit BYOB

Das Herzstück des Algorithmus stellt die Methode „downheap()“ dar.

```
private void downheap(int v)
{
    int w=2*v+1;
    while (w<n)
    {
        if (w+1<n)
            if (a[w+1]>a[w]) w++;

        if (a[v]>=a[w]) return;
        exchange (v, w);
        v=w;
        w=2*v+1;
    }
}
```

**Text 5: downheap in Java**

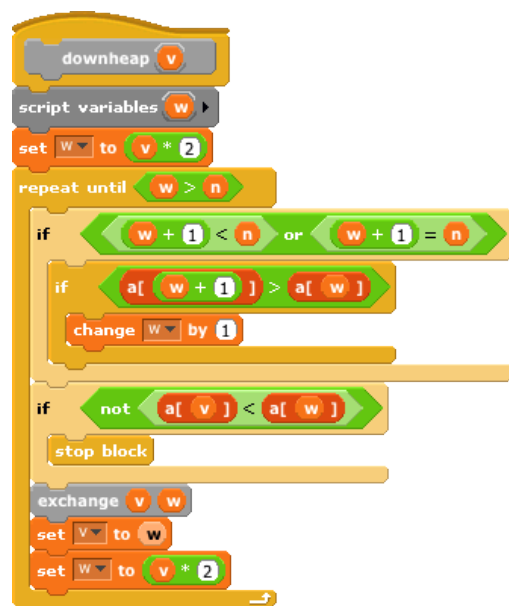


Abbildung 8: downheap in BYOB

<sup>23</sup> Übernommen von Harvey & Mönig (o.D.)

Erneut zeigen sich auch hier nötige Anpassungen aufgrund der verschobenen Indizes und des Fehlens einer „while“-Schleife in BYOB, die sich auch auf die Bedingungen der If-Abfragen auswirken. Der Abbruch mit Hilfe von „return“ musste mittels „stop block“ umgesetzt werden. Zur besseren Lesbarkeit wurde von mir ein eigener Block für die Zugriffe auf Elemente des Arrays erstellt, der wie folgt aussieht:



Abbildung 9: Array-Syntax in BYOB

Die letzte Methode, „exchange()“, dient ganz simpel dem Tausch zweier Elemente und konnte ohne große Abweichungen oder Anpassungen umgesetzt werden.

```
private void exchange(int i, int j)
{
    int t=a[i];
    a[i]=a[j];
    a[j]=t;
}

// end class HeapSorter
```

**Text 6: exchange in Java**

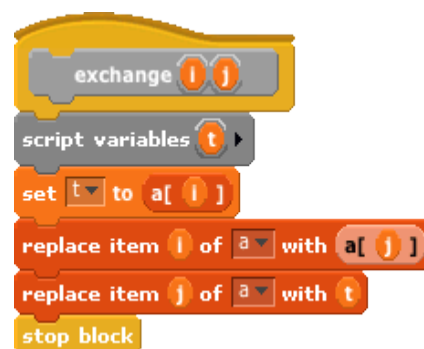


Abbildung 10: exchange in BYOB

## 4.4 Didaktische Spiele

Mit der Sammlung an Programmen und Aufgaben für BYOB möchte ich die Möglichkeiten demonstrieren, die es speziell für Anfänger und Anfängerinnen in der Programmierung gibt, um den Einstieg zu vereinfachen ohne dabei auf wesentliche Aspekte im Bereich Datenstrukturen und Algorithmen verzichten zu müssen. Dazu sind die erarbeiteten Projekte in spielerische Umgebungen eingebettet worden, deren Fokus aber dennoch so zentral wie möglich auf den inhaltlichen Schwerpunkten liegen. Es sollen schnell erste Erfolge erzielt werden können, um anfänglichen Frust zu vermeiden und möglichst sogar durch eine motivierende, interaktive Lernumgebung zu ersetzen. Meine Herangehensweise war nun, dass ich eine wesentliche

Komponente bei der Übersetzung von Spiel in Lernumgebung der Programmierung unterstellt habe, und zwar die Steuerung, um so die Interaktivität gleichzeitig zur Programmierung verwenden zu können. Es wurde also versucht den Lernenden ein Umfeld zu bieten, das die Anreize von Spielen mit dem Erfolg in der Programmierung verknüpfen kann und so die Motivation fördert. Die zu erreichenden Ziele und Rahmenbedingungen wurden so gewählt, dass sie meist von einfachen Lösungen bis hin zu recht komplexer Programmierung einen möglichst großen Spielraum zur Verfügung stellen, um Lehrenden den Einsatz einer Reihe von Algorithmen zu ermöglichen. Zusätzlich bieten die didaktischen Rahmenprogramme alle Vorteile, die BYOB von sich aus bereits mit bringt.

Marc Prensky hat im Rahmen von digitalen Lernspielen bzw. Digital Game-Based Learning einige Faktoren ausgearbeitet, die seiner Meinung nach wichtige Elemente darstellen und dafür Sorge tragen, dass Spiele in der Lage sind das Engagement der Spielerin oder des Spielers möglichst hoch zu halten. Diese möchte ich kurz in übersetzter und zusammengefasster Form vorstellen, da sie mir als Richtlinien und Leitfäden bei der Erstellung meiner didaktischen Spiele dienten, um die Motivation der Lernenden durch die Lernspiele möglichst gelungen zu fördern. Wenig überraschend werden von Prensky (2001, S. 106) als erste Faktoren Spiel und Spaß, gefolgt von Regeln und Zielen, die zur Orientierung und Motivation beitragen sollen. Spiele sollen weiters klarerweise interaktiv und adaptiv sein, sowie ein Feedback zum aktuellen Status bereit stellen und den Sieg klar auszeichnen. Wettbewerb bzw. Herausforderung, sowie Wettkampf und soziale Interaktionen sollen, ähnlich wie es in sportlichen Disziplinen der Fall ist, die Leistung steigern. Um auch Kreativität zu fördern, sind vor allem die möglichen Problemlösungsstrategien von Interesse sowie die Präsentation und Aufbereitung für den Bereich der emotionalen Beteiligung, wobei ich gerade zur Steigerung der Emotionalität und Einbettung in eine Geschichte oder ein Rollenspiel persönlich eher zur Verwendung des Unterrichtskonzepts von Sheldon tendiere, welches vielleicht gerade in Kombination mit didaktischen Lernspielen eine weitere Steigerung einiger der genannten Faktoren und Elemente bewirken oder auch übernehmen könnte, da diese sich in einigen Punkten überlappen.

Leitfragen für die Entwicklung und den Einsatz von Spielen im Rahmen von Digital Game-Based Learning bietet auch Philipp Daferner, der sie wie folgt zusammen fasst: „Das bedeutet, die Spezifikation sollte bereits hier um didaktische und lerntheoretische Elemente erweitert werden:

- Welche Inhalte sollen mit dem Spiel vermittelt werden?
- Wie werden diese Inhalte vermittelt?
- Was können Schülerinnen von diesem Spiel lernen, was können sie nicht lernen?
- Wie wird bei diesem Spiel auf unterschiedliche Lernniveaus eingegangen?
- Was sollten die Schülerinnen bereits vor dem Spielen des Spiels wissen?

Obwohl durch den Lerninhalt zusätzliche Schwierigkeiten und Aufgaben auf eine Spieleentwicklerin zukommen, darf das eigentliche Ziel des Spiels nicht aus den Augen verloren werden: der Spielspaß.“ (Daferner, 2010, S. 51 ff)

Die Inhalte des jeweiligen Spiels und mithin der dafür geeignete Einsatzbereich, sowie was gelernt werden kann, ist in den jeweiligen Abschnitten zu finden. Die Art der Vermittlung ist bereits geklärt, da diese mittels BYOB und den darin entwickelten Spielen erfolgen wird, in denen die Spielfiguren durch Algorithmen gesteuert werden müssen. Den unterschiedlichen Lernniveaus wird durch die Anknüpfung an bereits erlernte Inhalte versucht gerecht werden, wodurch auch das Vorwissen geregelt ist, das gegebenenfalls auch nicht vorhanden sein muss, wenn es sich um ein Spiel handelt, das für den direkten Einstieg geeignet ist.

#### 4.4.1 Catch me if you can

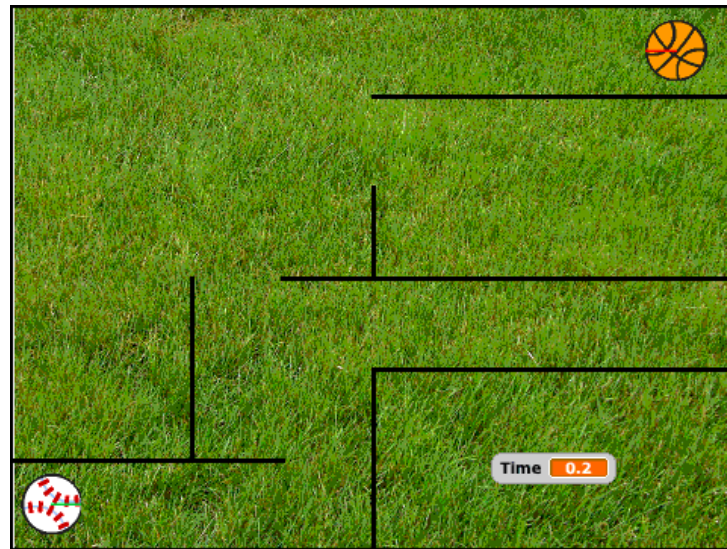


Abbildung 11: Catch me if you can

##### Einleitung

Hierbei gibt es ein Spielfeld, ähnlich einem kleinen, sehr übersichtlichen Labyrinth, und zwei programmierbare Spielfiguren, einen Jäger („Hunter“), der von dem orangen Basketball rechts oben dargestellt wird, und einen Gejagten („Hunted“), der weiße Baseball links unten. Generell besteht die Möglichkeit, dass beide Figuren von jeweils einem Spieler gesteuert werden können, wenn zum Beispiel ein Wettkampf zwischen zwei Algorithmen ausgeführt werden soll. Andererseits kann auch eine Spielfigur bereits programmiert sein, was beispielsweise sehr leicht durch eine zufällig gesteuerte Bewegung möglich ist, wobei auch ein Timer zur Verfügung steht um die Effizienz der Algorithmen zu messen. Zum Zweck der Einübung und Erkundung von BYOB kann der Algorithmus des Gejagten Balls natürlich auch zur Gänze entfernt werden, sodass nur der Weg durch das Labyrinth programmiert werden muss. Das Spiel endet, sobald sich die Farbe der Spielfiguren (orange und weiß) berühren, was ein eindeutiges Indiz dafür ist, dass der Gejagte gefangen wurde. Die Textur des Labyrinths kann jederzeit ausgetauscht und durch eine eigens erstellte ersetzt oder auch vollkommen weg gelassen werden, wobei im zweiten Fall darauf zu achten ist, dass der Timer ausgeblendet wird, da dieser im Anzeigefeld die Farbe orange hat und dadurch die Siegbedingung seinerseits erfüllen kann.

## Aufbau

Gestartet wird die Applikation durch das Klicken der grünen Fahne. Das Objekt „Stage“ dient als Hintergrund-Ebene, setzt den Timer und aktualisiert ihn in einer Endlosschleife. Außerdem kann hier die Bewegungsgeschwindigkeit des Jägers „Hunter“ über die globale Variable „HunterSpeed“ gesteuert werden. In der Abbildung wird dem Jäger ein Bonus von 20% gewährt.

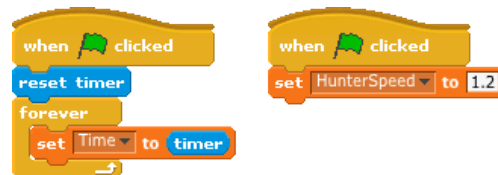


Abbildung 12: Stage

Die Regeln des Spiels sind in dem von mir erstellten Block „PlayCatch“ festgelegt. Hier wird das Spielfeld initialisiert, die Spielfiguren werden auf ihre jeweilige Anfangspositionen gestellt, die Siegbedingung in einer Repeat-Schleife festgelegt und die Algorithmen der Spieler ausgeführt. Außerdem wird zu Beginn noch sicher gestellt, dass die Figuren auch genau 50 Pixel groß sind.



Abbildung 13: PlayCatch

Wesentlich für die Funktionalität sind die beiden Bewegungsblöcke „moveMe“ und „moveMeBack“, die beide eine Abfrage implementiert haben, die verhindert, dass einfach durch die Wände des Labyrinths gefahren werden kann. Bei jedem Schritt vor- oder rückwärts findet eine Kollisionsabfrage zwischen Spielfigur und der Textur

des Labyrinths statt, die eine Bewegung nur dann ermöglicht, wenn keine Kollision stattfindet. Für den Sprite „Hunter“ wird aufgrund der höheren Bewegungsgeschwindigkeit, die durch eine Erhöhung der Schrittweite realisiert wurde, die Länge der gefahrenen Strecke durch eine Reduktion der Durchläufe der Schleife angepasst, wozu in beiden Fällen der Wert der globalen Variablen „HunterSpeed“ verwendet wird, damit zwar schneller, aber nicht weiter als gewünscht gefahren wird.

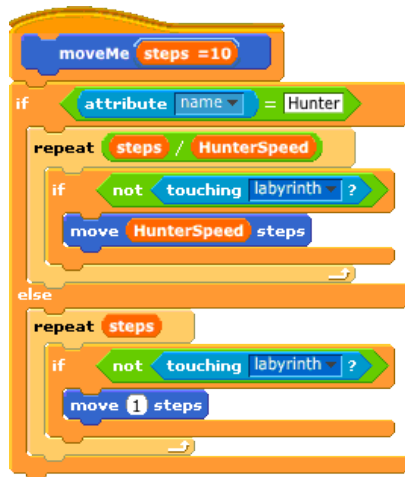


Abbildung 15: moveMe

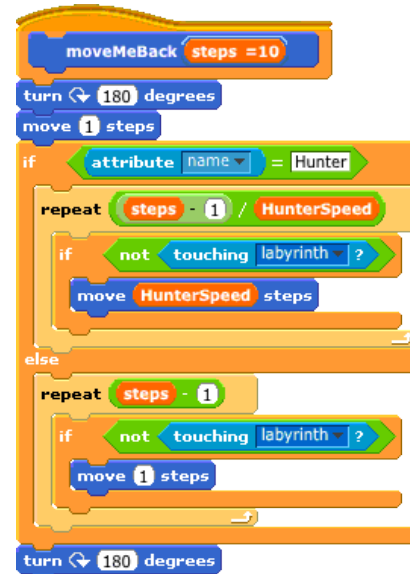


Abbildung 14: moveMeBack

Als Beispiel zur Veranschaulichung, wie der Einsatz des Rahmenblocks „PlayCatch“ und der Bewegungsblöcke aussehen könnte, möchte ich kurz eine zufallsgenerierte Bewegung präsentieren, die beispielsweise der Gejagte Sprite „Hunted“ beinhalten könnte. Hierbei wird so lange geradeaus gefahren, bis der Sprite des Labyrinths berührt wird, woraufhin zufällig nach rechts oder links abgebogen wird und der Algorithmus von vorne beginnt.

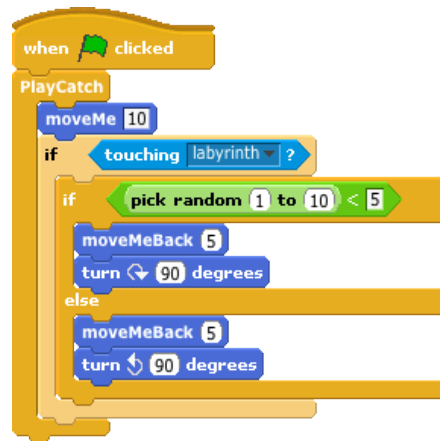


Abbildung 16: Zufallsgenerierte Bewegung

### Einsatzmöglichkeiten

Persönlich würde ich dieses didaktische Spiel eher zu Beginn und zur Erlernung der Prinzipien von BYOB verwenden, als für fortgeschrittene Programmierung, denn das war auch der Gedanke hinter der Entwicklung. Da die Bewegung durch die Repeat-Schleife im Block „PlayCatch“ wiederholt ausgeführt wird, ist lediglich eine If-Abfrage nötig, um sie zu steuern und wie aus der zufallsgenerierten Bewegung ersichtlich ist, reicht dies theoretisch schon aus, um das erste Spiel in BYOB zu meistern. Ein weiterführender Ausbau durch Verwendung des 2-Spieler-Modus oder einer kompetitiven Zeitmessung mit Bestenlisten ist durchaus denkbar und erwünscht, allerdings wäre es im zweiten Fall eventuell sinnvoll, wenn der flüchtende Sprite „Hunted“ eine einigermaßen fixe Route zugeteilt bekommt, da die Verwendung von Zufallsalgorithmen dazu führen könnte, dass die Lernenden sich dadurch benachteiligt fühlen, was jedoch auch durch das Einbinden von Wahrscheinlichkeiten zu einem erweiterten Lerneffekt ausgebaut werden könnte.

Eine weitere mögliche Lösung wäre, dass der Sprite „Hunter“ in der Mitte des Spielfeldes patrouilliert, da die Wahrscheinlichkeit sehr groß ist, dass der Gejagte früher oder später dieses Gebiet passiert. Hierzu wird aber bereits die Verwendung einer Schleife benötigt.

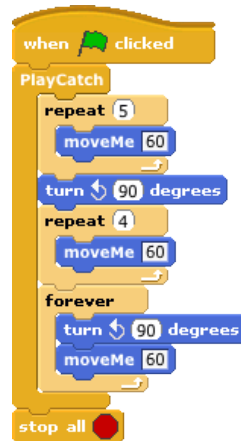


Abbildung 17: Bewegung "Hunter"

Das Labyrinth kann auf Wunsch auch getauscht werden, wobei zusätzlich die Möglichkeit zur zufälligen Auswahl implementiert wurde. Mittels Leertaste können die Kostüme des Sprites „labyrinth“ der Reihe nach durch geschalten werden, mit „r“ wird eines der Kostüme zufällig erwählt.

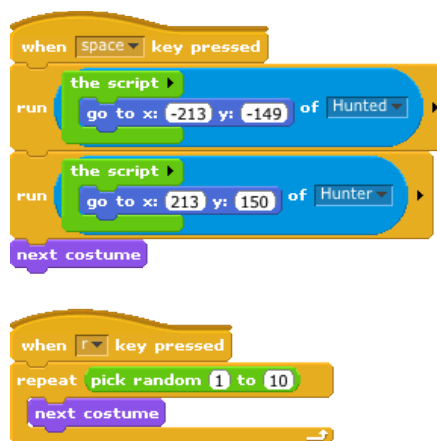


Abbildung 18: Wechsel des Labyrinths

#### 4.4.2 Circuit

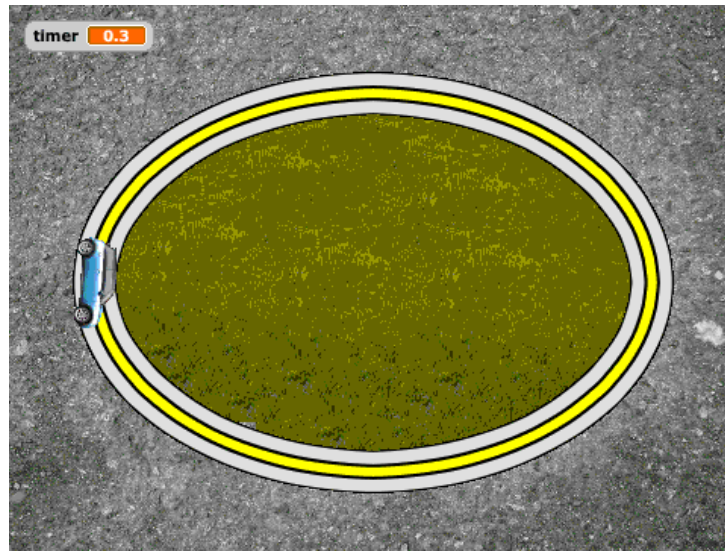


Abbildung 19: Circuit

Das Spielprinzip ist denkbar einfach: Ziel des Spiels ist es, eine Runde auf dem vorgegebenem Parcours zu fahren, ohne dass dabei die Bahn verlassen wird. Wieder steht ein Timer zum Zweck der Zeitmessung zur Verfügung um optional auch einen kompetitiven Modus verwenden zu können. Die Fahrbahn ist absichtlich leicht oval gestaltet, um den Schwierigkeitsgrad ein wenig zu erhöhen. Die Erweiterung auf ein Rennen über mehrere Runden wäre ebenfalls eine Option zum Ausbau beziehungsweise zur Erweiterung.

#### Aufbau

Auch der Aufbau ist hierbei wesentlich simpler, als im vorigen Lernspiel. Der Timer wird wieder durch einen Klick auf die grüne Fahne gestartet, das Skript selbst ist der „Stage“ zugewiesen, wo wiederum die Variable für die Schrittlänge des Spielers zu finden ist, die hier „steps“ genannt wird. Sowie der Timer zurückgesetzt wurde, erfolgt ein Broadcast für alle anderen Sprites, der als Startsignal dient, wodurch sichergestellt wird, dass erst danach das Rennen auch tatsächlich starten kann.



Abbildung 20: Stage

Das Sprite „Circuit“, welches die Rennbahn darstellt, beinhaltet ein Skript, das bei Erreichen des Checkpoints, der horizontal gegenüber der Startposition liegt, den Zähler für die globale Variable „checkpoint“ um 1 erhöht. Der Checkpoint dient zur Absicherung und Erleichterung, sodass die Rundenzahl sehr einfach über die Konditionen der Siegbedingungen gesteuert werden kann und um auf Nummer sicher zu gehen, dass auch wirklich die Strecke gefahren wird. Die Variable „halftime“ hat den Sinn, dass der Wert von Checkpoint nicht bei dessen Durchqueren um mehr als den Wert 1 erhöht wird, daher wird vor der nächsten Erhöhung die Zeitspanne der ersten halben Runde gewartet.

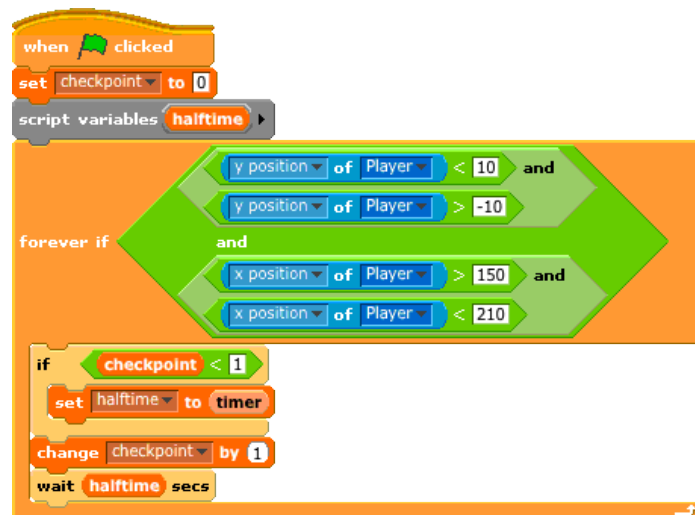


Abbildung 21: Circuit

Zu Beginn des Rennens werden wieder Positionierung, Größe und Blickrichtung festgelegt, sowie die Siegbedingungen geregelt. In der folgenden Abbildung ist das Rennen nach einer Runde absolviert, da der Checkpoint dazu nur einmal passiert werden muss (checkpoint=1). Der Timer wird dann ausgeblendet, da er während dem Glückwunsch zur Absolvierung inklusive Anzeige der Rennzeit weiter laufen würde.

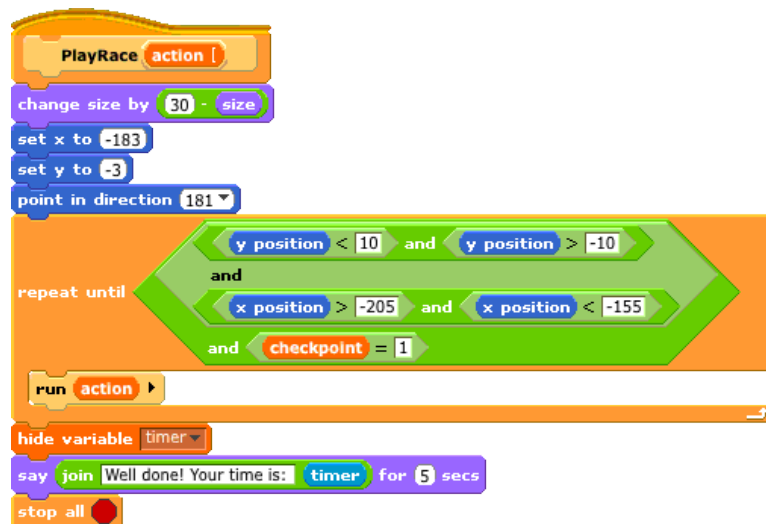


Abbildung 22: PlayRace

Auch bei diesem Lernprogramm steht wieder ein eigens dafür entwickelter Bewegungsblock zur Verfügung, damit auch möglichst bei jedem Schritt überprüft werden kann, ob der Spieler sich noch auf der Rennbahn befindet, oder diese bereits verlassen hat. Wie zuvor erfolgt auch hier die gewohnte Berechnung zur Anpassung der Schleifendurchläufe an die gewählte Schrittlänge.



Abbildung 23: MoveMe

Dadurch, dass bei jedem Schritt eine Überprüfung statt findet und die Bewegung in einzelne Schritte unterteilt wird, ist diese zwar langsamer, aber besser zu kontrollieren und mögliche Abkürzer werden vermieden.

## Einsatzmöglichkeiten

Durch den sehr kompetitiven Charakter eignet es sich sehr gut zur Festigung bereits erlernter Befehle, wobei sich Schleifen, eventuell in Kombination mit Abfragen, sehr gut eignen um den Parcours zu meistern. Eine sehr einfache Lösungsmöglichkeit, die nur Schleifen verwendet, wäre zum Beispiel folgende:



Abbildung 24: einfache Lösung

### 4.4.3 Donkey Kong



Bei dem nächsten Spiel, das ich präsentieren möchte, geht es im Prinzip recht simpel darum, den Fässern durch Bewegung nach links oder rechts auszuweichen. Die Spielfigur, die oben auf dem Balken steht („TheKong“ genannt), versucht die Figur

am Boden („TheHero“), die grundsätzlich vom Spieler programmiert wird, mit einem Fass zu treffen. Als kleinen Anreiz gibt es für den Spieler einen Punkt pro ausgewichenem Fass. Generell verfolgt dabei die Fässer werfende Spielfigur jene am Boden, wobei die Frequenz des Abwurfs zufällig geschieht, damit eine Flucht nicht allzu leicht vonstatten geht. Der Start erfolgt wieder durch das Klicken auf die grüne Fahne.

## Aufbau

Der Sprite „Stage“ hat in diesem Spiel eine zentrale Rolle und dient als eine Art Optionsmenü, in dem einige wichtige Variablen eingestellt und dadurch an die jeweiligen Wünsche angepasst werden können. Die möglichen Einstellungen betreffen die Bewegungsgeschwindigkeit und Schrittweite der Figur „TheKong“ („KongSpeed“ und „KongSteps“), sowie die Fallhöhe der Fässer auf der die zufallsgenerierte Abwurfsequenz des nächsten Fasses gestartet wird („dropLevel“) und der Grenzwert („dropIfLessThan“), der festlegt, welcher Wert unterschritten werden muss, damit das Fass tatsächlich abgeworfen wird, wobei die Spanne der Zufallswerte 1-100 beträgt und der Wert in diesem Fallbeispiel die Bedingung „< 3“ erfüllen muss.



Abbildung 25: Stage

Zu Beginn des Spiels werden die Figuren auf die Startpositionen gestellt, ihre jeweilige Größe angepasst, die Fässer werden initialisiert und die Punkte werden auf 0 gesetzt. Durch einen einzigen Block („moveKongToPlayer“) kann dabei „TheKong“, also die Figur, welche die Fässer wirft, automatisiert gesteuert werden, denn prinzipiell besteht natürlich die auch Möglichkeit, dass beide Figuren selbst programmiert werden.

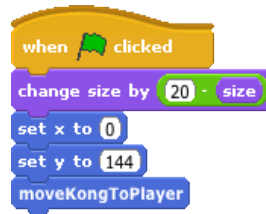


Abbildung 26: TheKong

Der Algorithmus der Spielfigur „TheKong“ ist in der nächsten Abbildung zu sehen.

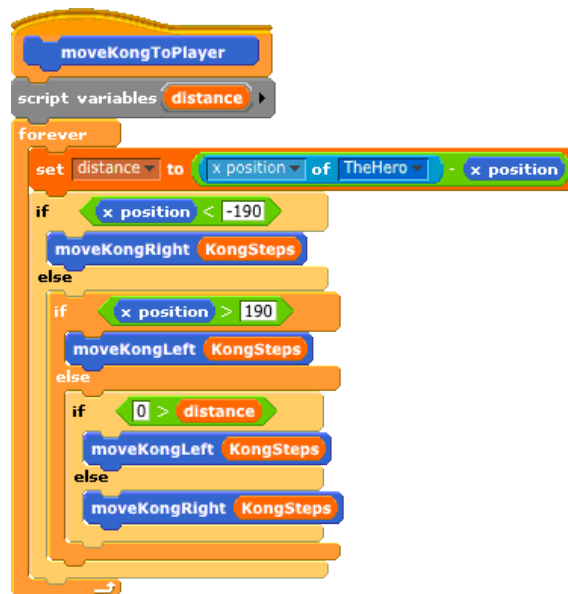


Abbildung 27: moveKongToPlayer

Dem Bewegungsraum wurden auf der X-Achse linker- und rechterhand Grenzen gesetzt (-190 beziehungsweise 190, wobei der Rahmen, den BYOB vorgibt, eine Abmessung von -240 bis 240 hat), da es in den Testläufen andernfalls oft passieren konnte, dass „TheKong“ sich zu lange an den Rändern aufhielt und den Verfolgten aus den Augen verlor. Die Bewegung wird von einer Endlosschleife umrahmt, wobei es ausschlaggebend für die Richtung der Figur „TheKong“ ist, wohin sich der Spieler bewegt, was durch eine Abfrage von dessen X-Position geschieht. Die Namensgebung der Skriptvariablen „distance“ ist zugegebenermaßen etwas irreführend, da sie nicht die Distanz, sondern eigentlich den Richtungsvektor speichert. Ist diese Variable also negativ, dann ist dies ein Indiz dafür, dass sich die Figur des Spielers links von der Figur „TheKong“ befindet und vice versa. Die Variable „KongSteps“ erlaubt eine Anpassung der Schrittlänge für alle Bewegungsblöcke, die im Sprite „Stage“ geschieht, und dient somit lediglich einer

komfortableren Adjustierung. Hinter den Blöcken „moveKongLeft“ und „moveKongRight“ befinden sich die Algorithmen zur Bewegung in die gewünschte Richtung, die zum Zweck der Übersichtlichkeit und besseren Lesbarkeit in eigene Blöcke verpackt wurden.



**Abbildung 29:**  
**moveKongLeft**

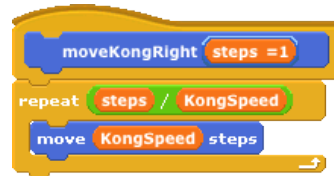


**Abbildung 28:**  
**moveKongRight**

Für die Bewegung und die Anpassung der Schrittweite an die eingestellte Geschwindigkeit wird die selbe Formel verwendet, die schon im vorigen Spiel zum Einsatz kam und regelt, dass die gefahrene Länge trotz erhöhter Geschwindigkeit konstant bleibt. Diese kann für die Figur „TheKong“ wieder im Sprite „Stage“ angepasst werden.

Die Fässer sind in 4 eigenen Sprites realisiert, deren Algorithmen aber zu einem großen Teil recht ähnlich sind. Da die Bedingung für einen möglichen Abwurf ist, dass der Vorgänger unter die Y-Position 0 gesunken ist, und erst dann in einer Schleife auf den zufallsgenerierten Auslöser gewartet wird, benötigt das erste Fass eine kleine Anpassung, da es beim ersten Abwurf keinen Vorgänger gibt. Hierzu wurde zusätzlich die auslösende Bedingung implementiert, dass der Punktestand 0 beträgt.

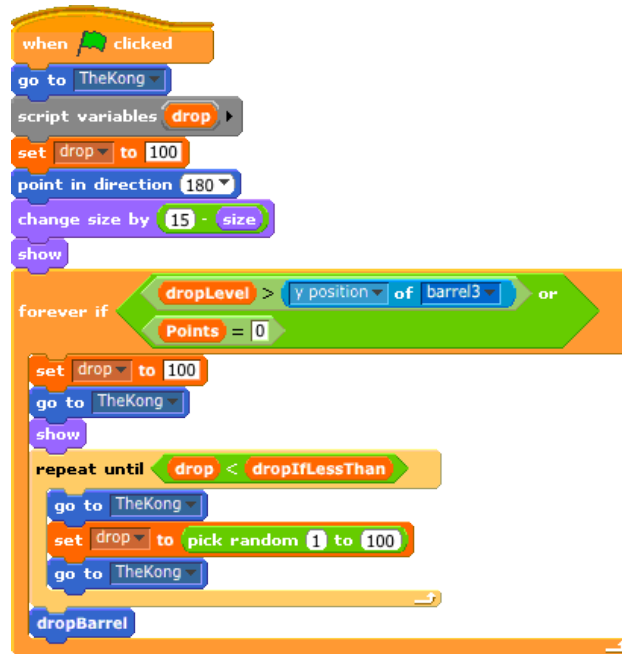


Abbildung 30: Barrel

Das erste Fass („barrel“) wird erstmal initialisiert, wobei Richtung, Größe sowie X- und Y-Position durch den Block „go to TheKong“ angepasst werden und das Sprite sichtbar geschaltet wird. Da hier der Sonderfall auftritt, dass es beim ersten Durchlauf keinen Vorgänger gibt, kann nicht einfach wie bei den anderen Fässern darauf gewartet werden, dass das vorige Fass unter die im Sprite „Stage“ eingestellte Grenze fällt, die hier auf den Wert 0 gesetzt wurde. In der Schleife wird das Fass so lange an die Position von „TheKong“ bewegt, bis der Skritvariablen „drop“ ein Wert unter dem gewählten Grenzwert zugewiesen wird, erst dann erfolgt der eigentliche Abwurf durch „dropBarrel“, das wie folgt aufgebaut ist:

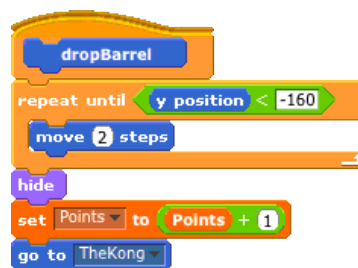


Abbildung 31: dropBarrel

Der Fall ist eine Änderung der Y-Koordinate, der so lange vonstatten geht, bis das Fass aus dem sichtbaren Bereich gefallen ist, woraufhin es ausgeblendet und repositioniert, sowie Punkteanzahl um 1 erhöht wird. Für die anderen Fässer wurde der Algorithmus etwas abgeändert, da nicht mehr der fehlende Vorgänger beim ersten Durchlauf beachtet werden muss. Folglich ändern sich bei den restlichen 3 Fässern nur noch die Werte für den jeweiligen Vorgänger.



Abbildung 32: Barrel1

Für die Bewegung des Spielers gibt wieder eigene Blöcke, einen für das Gleiten nach rechts und einen nach links, wobei der Unterschied lediglich im Vorzeichen des Blocks „move -1 steps“ sowie in der Grenze des Bewegungsraums liegt.



Abbildung 33: moveLeft

Da die Funktionen aller Teilnehmer festgelegt sind, fehlt eigentlich nur noch der Rahmen des Spiels, in dem dessen Beendigung geregelt wird.

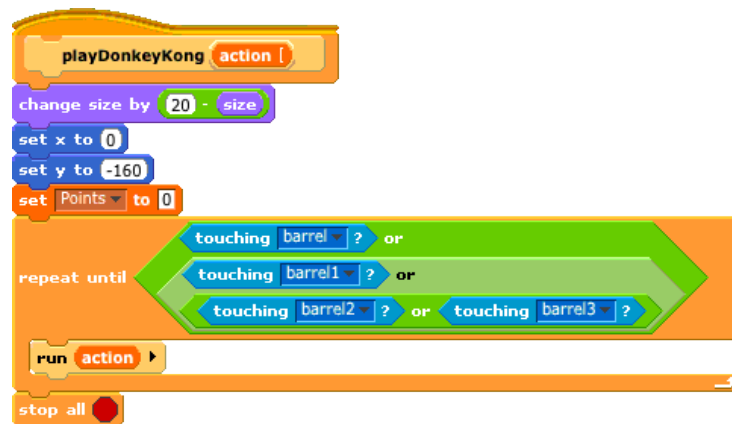


Abbildung 34: playDonkeyKong

Es handelt sich dabei wieder um einen C-förmigen Block, der den Algorithmus des Spielers „TheHero“ in einer Schleife ausführt. Nach der üblichen Initialisierung wird das Spielziel als Abbruchbedingungen einer Repeat-Schleife realisiert, in der festgelegt wird, dass der Spielcharakter keines der Fässer berühren darf.

### Einsatzmöglichkeiten

Das Lernspiel ist vorrangig zur Festigung der Kenntnisse entwickelt worden, wobei die Lernenden eigentlich schon sehr selbstständig an die Entwicklung der Algorithmen heran gehen sollten. Durch den Handlungszwang, der durch den Abwurf der Fässer entsteht, könnte auch die potenzielle Frustration etwas höher sein, wenn das eigene Programm nicht gleich funktioniert, wodurch es sich weniger als Erkundung von BYOB und zur Erlernung dessen Prinzipien eignet. Ist aber eine grundsätzliche Orientierung in der Lernumgebung vorhanden, besteht die Möglichkeit diese hiermit etwas zu festigen, sodass auf einem möglichst solidem Fundament aufgebaut werden kann. Bezüglich Algorithmen würde ich es persönlich als Einführung in Schleifen, die auch mit If-Abfragen gemischt werden können, verwenden. Dabei können auch die Distanz zu den Fässern oder die X-Position von „TheKong“ als Orientierungshilfen mit hinein genommen werden, wofür es bereits vorgefertigte Blöcke gibt.



Abbildung 35: optionale Erweiterungen

Eine sehr einfache Lösungsmöglichkeit wäre es, wenn nur Schleifen verwendet werden, was zum Beispiel wie folgt aussehen könnte.



Abbildung 36: Lösung mit Schleifen

Für eine weiterführende Beschäftigung wurden noch zwei weitere Blöcke entwickelt, die dem lernbegierigen Spieler erweiterte Handlungsmöglichkeiten bieten. Der erste Block „distanceToBarrelY“, der als ein Reporter-Block realisiert wurde, liefert den Wert der Distanz von der Spielerfigur zum dem Fass mit der niedrigsten Y-Position, also demjenigen, welches am tiefsten gefallen ist.

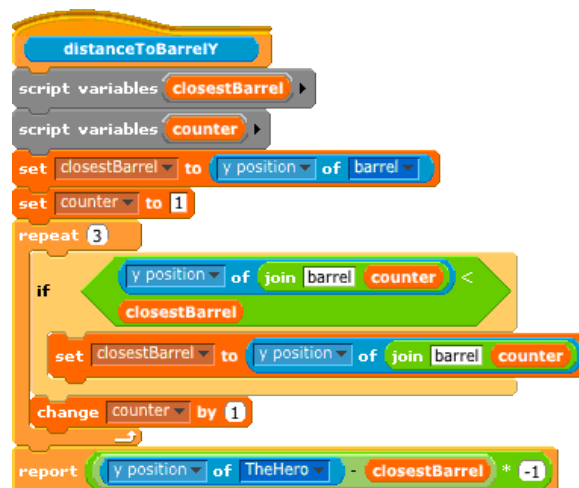


Abbildung 37: distanceToBarrelY

Der zweite Block „closestBarrel“ liefert nicht die Distanz, sondern den Namen des Fasses, das die kürzeste Distanz zur Spielfigur aufweist.

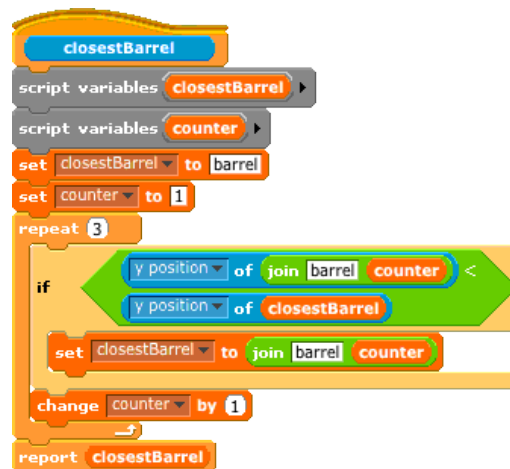


Abbildung 38: closestBarrel

Mit Hilfe dieser Blöcke können schon weitaus effizientere Algorithmen erarbeitet werden, die auch in einer durchschnittlich höheren Punktezahl resultieren.

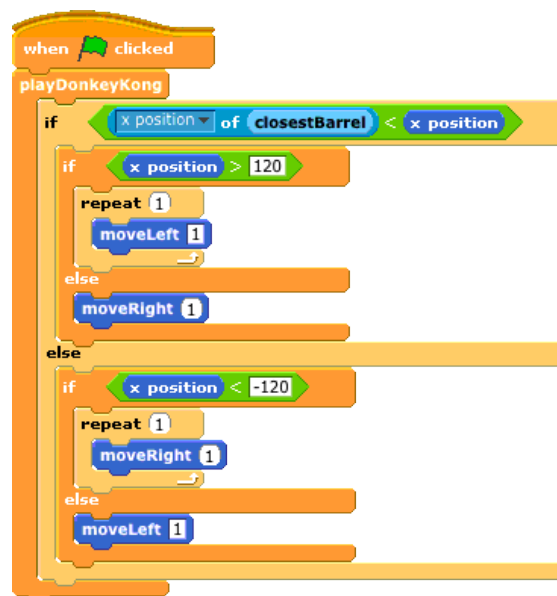


Abbildung 39: fortgeschrittene Lösung

Bei diesem Algorithmus werden einerseits der Bewegung links und rechts Grenzen gesetzt, damit nicht allzu lange an den Rändern verweilt wird, und andererseits wird durch die Verwendung des Reporters „closestBarrel“ die Bewegung an den Fassabwurf ein wenig besser angepasst und versucht, den Fässern auszuweichen.

#### 4.4.4 PacMan

Das Lernprogramm „PacMan“ wurde von mir in drei verschiedenen Schwierigkeitsgraden entwickelt, nämlich „easy“, „medium“ und „hard“, wodurch sich das Paket sehr gut zur Anknüpfung und Erweiterung der bereits durch die andere Lernprogramme erlernten Inhalte eignet. Die Idee hinter dem Paket ist, dass die Zeit, die zur Erklärung des Spielprinzips benötigt wird, möglichst reduziert wird und die Inhalte mehr in den Vordergrund gerückt werden. Dadurch soll die Effizienz des Lernspiels gesteigert und der Fokus mehr auf die Algorithmen gelenkt werden. Alle zuvor präsentierten Programme sollen gewährleisten können, dass ein ausreichend solides Fundament vorhanden ist, auf das aufgebaut werden kann. Voraussetzung für die möglichst erfolgreiche Anwendung dieses Pakets ist die grundlegende Bedienung von BYOB, die Lernenden sollen also in der Lage sein, ein Programm erstellen zu können. An Algorithmen würde ich das generelle Verständnis für die Funktionsweisen von Schleifen und bedingten Anweisungen empfehlen. Sofern das algorithmische Verständnis vorhanden ist, können zum Zweck Einführung in BYOB einzelne Lernprogramme dazu verwendet werden, um die Programmierweise von BYOB kennen zu lernen und erste Schritte darin zu unternehmen, es brauchen dazu dann nicht mehr alle Module absolviert werden. Die ersten beiden Labyrinth, „easy“ und „medium“, sollen Möglichkeiten aufzeigen, wie ein Algorithmus, der vorwiegend auf Schleifen basiert, mittels Variablen und Listen wesentlich kompakter umgesetzt werden kann. Im dritten Schritt, auf der Stufe „hard“, können die ersten beiden Module wiederholt werden, wenn erwünscht, und darauf aufbauend wird die Möglichkeit geboten, dass Rekursionen zur Lösung verwendet werden.

Das Spielprinzip orientiert sich sehr stark an dem bekannten Videospiel, wobei die gegnerischen Gespenster weggelassen wurden, da dadurch ein wesentlicher Faktor für eine zu hohe Komplexität ausgeklammert werden konnte. Das Ziel ist es wie gewohnt, alle Punkte auf dem Spielfeld einzusammeln, das die Form eines Labyrinths hat. Die Labyrinth können sehr einfach durch Eigenkreationen ersetzt werden, um eine gewisse Flexibilität zu ermöglichen und damit sie, falls nötig, dem Wissensstand und Fortschritt der Lernenden angepasst werden können. Die

Entwicklung eigener Labyrinth kann aber natürlich auch zu Test- und Prüfungszwecken eingesetzt werden.

#### 4.4.4.1 PacMan easy

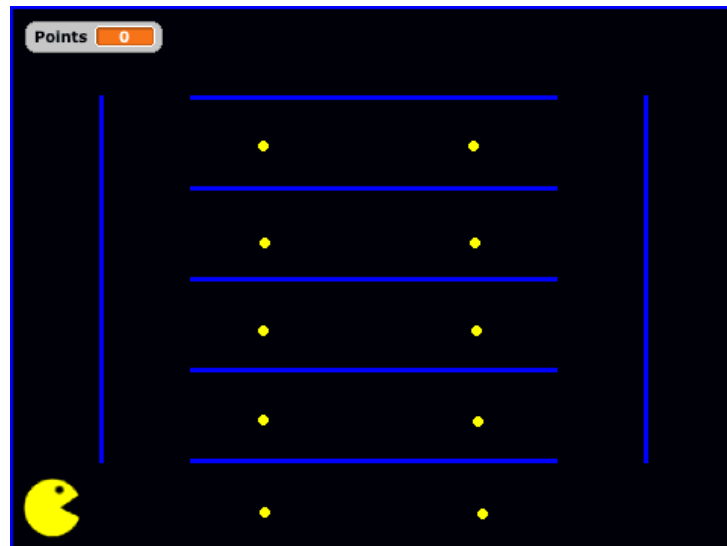


Abbildung 40: PacMan - easy

Das Ziel des Spiels ist es, mit der gelben Spielfigur „PacMan“ links unten die gelben Punkte einzusammeln. Das Raster, das dem Spielfeld zugrunde liegt, besteht aus Quadraten zu je 60x60 Pixeln, was zur besseren Orientierung auf dem Spielfeld beitragen soll. Im ersten Modul soll den Lernenden gezeigt werden, wie rein auf Schleifen basierende Algorithmen mittels der Verwendung von Variablen und Abfragen kompakter umgesetzt werden können. Ob der Algorithmus mit Repeat-Schleifen bereits vorgegeben ist, oder mit den Lernenden gemeinsam erarbeitet wird, lasse ich im Ermessen des Lehrenden.

#### Aufbau

Der Aufbau des Spiels beruht diesmal auf Schichten (engl.: „Layer“), da die Punkte, die aufgesammelt wurden, vom Spielfeld verschwinden sollen. Um sicher zu stellen, dass alle Sprites, die benötigt werden, auch sichtbar sind, werden sie bei der Initialisierung mit dem Block „go to front“ in den Vordergrund gerückt. Als eine Art

Vorhang dient ein Sprite namens „Background“, der lediglich eine schwarze Fläche ist, hinter der die gesammelten Punkte versteckt werden. Der Spielblock ist wieder eine C-förmige Repeat-Schleife, in der die Befehle ausgeführt werden, nachdem die Positionierung der Spielfigur abgeschlossen ist. Zuvor wird allerdings noch ein Broadcast an alle Sprites versendet, der sicherstellt, dass der Punktestand fertig initialisiert ist, damit das Spiel nicht gewonnen ist, bevor es überhaupt begonnen hat, da die Siegbedingung auch nochmals im Sprite „Stage“ geregelt ist. Für jeden Punkt, der eingesammelt wird, erhält der Spieler 10 Punkte, sodass ein Erreichen von 100 Punkten das Spiel beendet.



Abbildung 41: PlayPacMan

Der Glückwunsch zur erfolgreichen Beendigung des Spiels wurde, wie bereits erwähnt, im Sprite „Stage“ realisiert und ist ebenfalls erreicht, wenn der Spieler 100 Punkte sammelt. Hier kommt auch der Broadcast zum Tragen, da er das auslösende Ereignis für das Skript darstellt. Der Clou hierbei ist, dass der Spielfigur, aufgrund der erfüllten Bedingung der Abfrage im Sprite „Stage“, sprichwörtlich die Worte der Gratulation in den Mund gelegt werden, da das Skript mit der „say“ Anweisung vom Objekt „PacMan“ ausgeführt wird, bevor die Ausführungen aller Sprites gestoppt werden.



Abbildung 42: Stage

Die Punkte, die es zu sammeln gilt, haben denselben Algorithmus. Zu Beginn werden sie in den Vordergrund gerückt, damit sie von der Spielfigur berührt und somit als gesammelt verbucht werden können, was in der If-Anweisung geregelt ist. Berühren sie „PacMan“, rücken sie in den Hintergrund, erhöhen den Punktestand um +10 und warten dann bis die Figur sie nicht mehr berührt, da andernfalls der Punktestand bereits nach dem ersten einzusammelnden Punkt erreicht ist und das Spiel beendet wird.

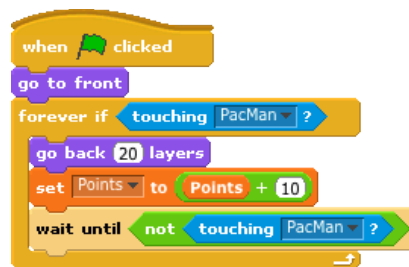


Abbildung 43: Punkte

Es gibt auch hier wieder einen eigenen Bewegungsblock („moveMe“), damit garantiert ist, dass die Spieler sich nicht völlig frei bewegen können, sondern sich an den Grenzen des Labyrinths (der Sprite mit der verwendeten Grafik wurde „playground“ genannt) orientieren müssen. Die Abfrage erfolgt auch hier bei jedem einzelnen Schritt.

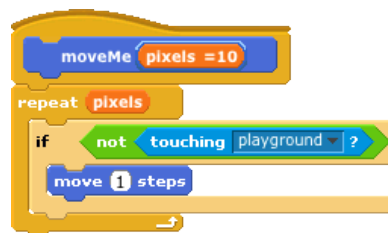


Abbildung 44: moveMe

### Einsatzmöglichkeiten

Wie bereits erwähnt liegt der Fokus bezüglich Einsatz sehr klar auf der Weiterentwicklung der grundlegendsten Fähigkeiten in der Programmierung und im Umgang mit BYOB. Es soll daher möglichst direkt an dem Wissensstand angeknüpft werden und bereits bekannte Inhalte sollen adaptiert werden, wodurch die Komplexität niedrig gehalten werden kann und gleichzeitig die neuen Inhalte in den Vordergrund treten. Zu diesem Zweck wird von einem Algorithmus, der rein mit

Repeat-Schleifen realisiert wurde, ausgegangen, der dann in einem weiteren Schritt mit Variablen gelöst werden soll. Die Funktionsweise ist also bei beiden die gleiche, nur die Art der Umsetzung ändert sich. Der algorithmisch einfachste Lösungsweg ist das Durchfahren des Labyrinths mittels Schleifen, was zusätzlich durch die genormte Rasterung erleichtert wird, da die Bewegung in das nächste Feld eine Länge von 60 Schritten hat. Die folgende Abbildung zeigt bereits eine verschachtelte Schleife, wobei diese auch aufgelöst und zerlegt werden könnte.



Abbildung 45: Lösung mit Schleifen

Es zeigt sich hier bereits sehr deutlich ein Muster, da nach der ersten Geraden alle anderen die gleiche Länge von 5x60 Schritten haben. Der Bogen, der zur Erreichung der nächst höheren Ebene gefahren werden muss, ist ebenfalls immer gleich lang, es ändert sich hierbei nur die Richtung, in die gedreht wird, wodurch die Schleifen sich zusammenfassen ließen. Dieser Algorithmus sollte von den Lernenden bereits problemlos verstanden und interpretiert werden können, da Schleifen zur Genüge in den vorherigen Lernspielen erlernt und eingeübt werden konnten.

Die darauf folgende Aufgabe besteht nun darin, den Code weiter zusammen zu fassen, da die vertikale Bewegung noch immer doppelt vorkommt. Zur Lösung empfiehlt sich hierbei eine bedingte Anweisung, damit einmal im und einmal gegen den Uhrzeigersinn gedreht wird. Es wird eine Variable „i“ mit dem Wert 1 initialisiert, die dann in der Schleife mit -1 multipliziert wird, um so bei jedem

Durchlauf die Richtung wechseln zu können, da der Wert von „i“ zwischen 1 und -1 hin und her springt.



Abbildung 46: fortgeschrittene Lösung

#### 4.4.4.2 PacMan medium

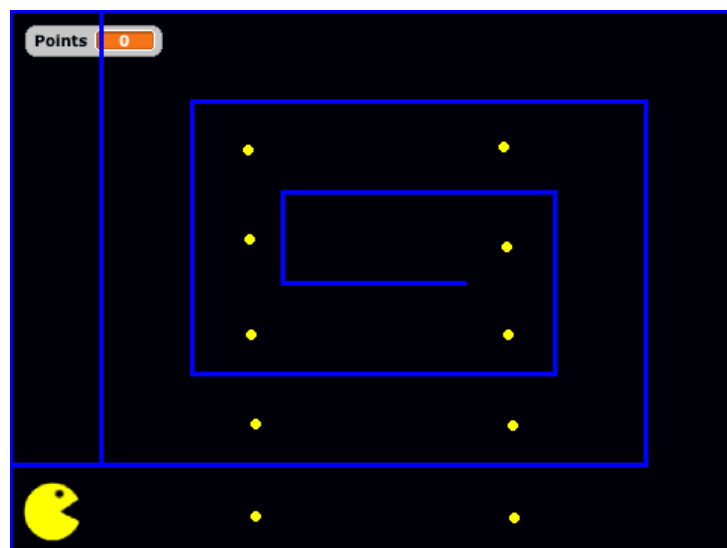


Abbildung 47: PacMan medium

In der nächsten Stufe wurde das Labyrinth so angepasst, dass die reine Verwendung von Schleifen nun nicht mehr sehr zielführend ist. Durch die ständig wechselnden

Längen der Strecken ergäbe sich dadurch eine lange Aneinanderreihung von Blöcken, die auch nicht mehr so leicht in einer Schleife zusammengefasst werden könnten. Zwar bleibt die Richtung, in die gedreht wird konstant, da es sich hierbei um ein spiralenförmiges Labyrinth handelt, doch die horizontal beziehungsweise vertikal zu fahrenden Längen werden ausgehend von den Startwerten immer um 1 verringert.

## Aufbau

Im Aufbau ändern sich eigentlich nur kleine Details, so zum Beispiel im Block „PlayPacMan“, denn dort wurde nun die Repeat-Schleife entfernt und gegen einen einfachen „run“-Block ersetzt, der nun für die Ausführung des Algorithmus zuständig ist. Außerdem wurde ein aufmunterndes „Try again!“ eingebaut, sollte nach Ausführung des Codes der nötige Punktestand nicht erreicht sein.

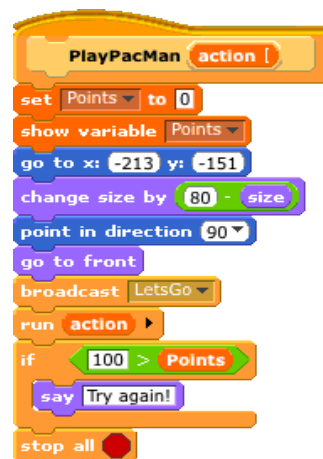


Abbildung 48: PlayPacMan

Eine kleine Änderung gibt es auch für „playground“, der das Labyrinth darstellt. Dieses kann nämlich nun durch drücken der Leertaste gewechselt werden.



Abbildung 49: playground

Hier stehen die Grafiken vom Modus „easy“ und vom Modus „medium“ zur Verfügung, die als separate „costumes“ (Kostüme) definiert wurden. Dadurch kann ohne großen Aufwand der erste Modus nochmals besprochen und geübt werden, falls dazu Bedarf besteht.

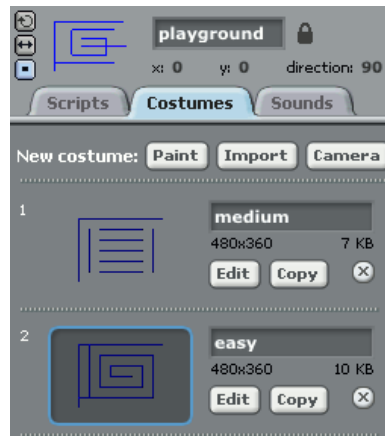


**Abbildung 50: Costumes**

### **Einsatzmöglichkeiten**

Auch in diesem Modus besteht wieder die Möglichkeit an das zuvor Erlernte anzuknüpfen. Das Spielfeld bietet von rein auf Schleifen basierten Algorithmen, über die Verwendung von Variablen bis hin zur Einführung in Listen ein sehr breites Spektrum. Aufgrund des spiralförmigen Aufbaus ergeben sich für die horizontalen Strecken die Längen 7, 6, 5, 4 und 3, sowie für ihre vertikalen Pendants die Längen 5, 4, 3, 2 und 1. Eine naheliegende Lösung ist daher die Verbindung von Schleifen mit zwei Variablen, die mit den Startwerten 7 (horizontal) und 5 (vertikal) initialisiert werden. Diese Werten müssen lediglich iterativ um 1 vermindert werden und schon sind die Längen der Strecken bereit zur Verwendung. Da es insgesamt 10 Geraden zu fahren gilt, werden die Bewegungsanweisungen noch in eine Schleife eingebettet, die nach 5 Durchläufen beendet ist. Der fertige Algorithmus wird in der nachfolgenden Abbildung dargestellt.

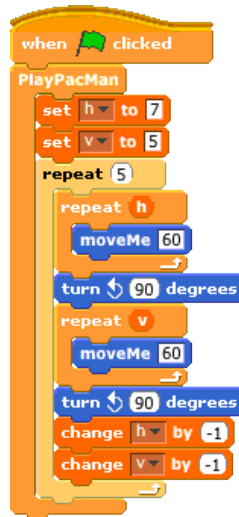


Abbildung 51: Schleifen und Variablen

Aufbauend auf die Wiederholung von Variablen soll nun ein Algorithmus erstellt werden, der Listen zur Bewältigung der Strecke verwendet. Eine andere Betrachtung der Streckenlängen ergibt sich, wenn sie hintereinander geschrieben werden: 7, 5, 6, 4, 5, 3, 4, 2, 3, 1. Die Differenz der Werte beträgt jeweils -2 und +1, was sich entweder sehr einfach als Variable oder auch als eine kleine Liste implementieren lässt. Zu Beginn werden daher zwei Variablen „steps“ und „i“ mit den Werten 7 für die anfänglich Streckenlänge und 1 für den ersten Zugriff auf die Werte in der Liste, die dann selbst erst einmal gelöscht werden muss, damit alle vorherigen Werte entfernt werden. Nachdem die Werte der Differenzen in die Liste eingefügt wurden, kann die Bewegung beginnen, wobei die Schrittlänge mittels Listeneinträgen berechnet wird.

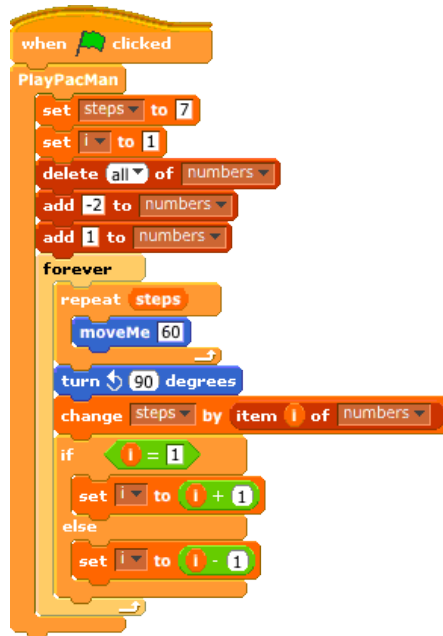


Abbildung 52: Algorithmus mit Liste

Eine weitere Möglichkeit in diesem Modus mit Listen zu arbeiten ist, dass die Längen der Strecken in einer Liste gespeichert werden, die dann schrittweise abgefahren werden. Dafür wird neben der Liste mit den Werten nur noch eine Variable zum Zugriff auf die Indizes der Elemente benötigt, die nach jedem Durchlauf um 1 erhöht wird, wobei die Anzahl der Durchläufe durch die Länge der Liste bereits festgelegt ist.

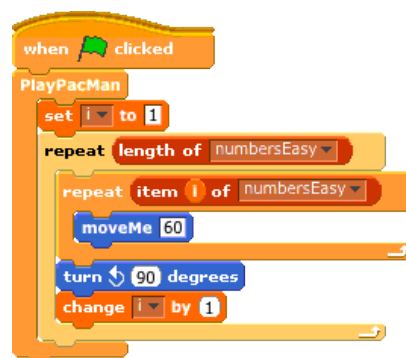


Abbildung 53: Algorithmus mit Liste

#### 4.4.4.3 PacMan hard

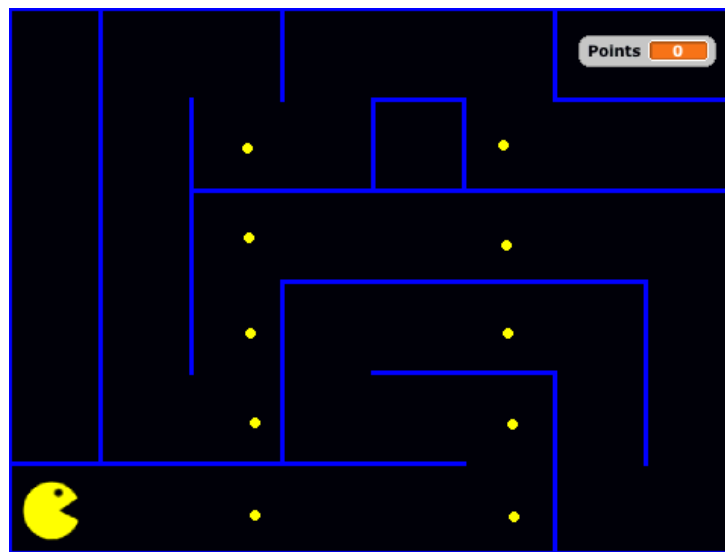


Abbildung 54: PacMan hard

Das Labyrinth in diesem Modus ist absichtlich nicht mehr so einfach berechenbar, doch auch hier soll wieder an bereits erlernte Algorithmen angeknüpft werden können. Durch das Drücken der Leertaste kann auf die beiden vorherigen Modi zurückgegriffen werden, falls diese wiederholt oder geprüft werden sollen. Diesmal sind die Richtungen sowie die Wegstrecken möglichst chaotisch, sodass auf etwas komplexere Algorithmen zurückgegriffen werden muss. Der Aufbau ist identisch mit dem Modus „medium“, daher werde ich diesen Punkt überspringen.

#### Einsatzmöglichkeiten

An den erlernten Inhalten des vorherigen Lernspiels, in dem mit Listen gearbeitet wurde, soll nun angeknüpft werden. Eine Möglichkeit dieses Labyrinth mittels Listen bewältigen zu können, ist, die Längen der Strecken sowie die Blickrichtung der Spielfigur in zwei Listen zu speichern und diese iterativ zu durchlaufen. Zu Beginn wird eine Skript Variable „j“ mit dem Wert 1 initialisiert, da die Indizes von BYOB nicht wie gewohnt bei 0, sondern bei 1 beginnen. In einer Schleife wird nun immer zuerst die Länge der Strecke gefahren und im Anschluss daran die Blickrichtung geändert, wobei anzumerken ist, dass die Werte für rechts und links „90“ bzw. „-90“

betragen und für auf- und abwärts „0“ bzw. „180“, was etwas ungewohnt anmuten möchte.

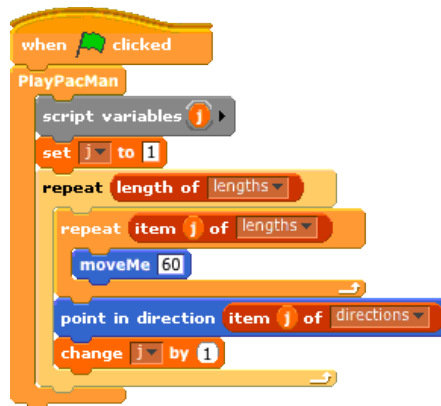


Abbildung 55: Lösung mit Listen

Wieder soll dieser Algorithmus im nächsten Schritt mit anderen Mitteln, diesmal einer rekursiven Lösung, umgesetzt werden. Dafür wird zum ersten Mal ein eigener Block erstellt werden, der von mir „movement“ getauft wurde. Diesem Block werden beim ersten Aufruf drei Parameter übergeben, und zwar die ersten beiden Elemente der Listen, die auch hier weiter verwendet werden können, und der Zähler für die Iteration, der beim ersten Aufruf den Wert 1 hat.



Abbildung 56: rekursive Lösung

Der Block selbst ähnelt im Aufbau sehr stark dem vorigen Algorithmus, da die Repeat-Schleife, die Anweisung für die Blickrichtung und das Inkrement fast unverändert übernommen werden können. Anstatt der umrahmenden Schleife folgt nun jedoch der rekursive Aufruf des Blocks mit den neuen Parametern. Im Vergleich zur iterativen Lösung wäre dieser Ansatz wahrscheinlich wohl nicht die erste Wahl für die zu bewältigende Aufgabe, doch besteht gerade hierdurch auch die Möglichkeit, dass ein schon sehr fortgeschrittenes Konzept, wie eine Rekursion, durch möglichst geringe Abweichung vom Algorithmus implementiert werden kann,

wodurch eine erhöhte Chance besteht, dass die Funktionsweise von Rekursionen leichter verstanden werden könnte, da die Änderungen marginal sind und daher der Fokus zentraler auf dem neuen Inhalt liegt.

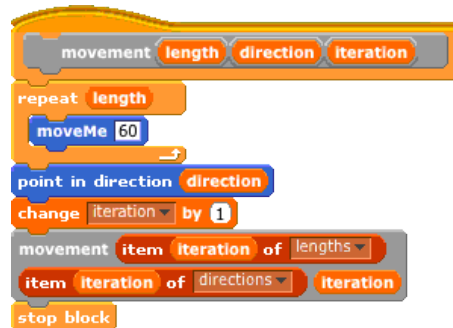


Abbildung 57: Der Block "movement"

#### 4.4.4.4 PacMan master

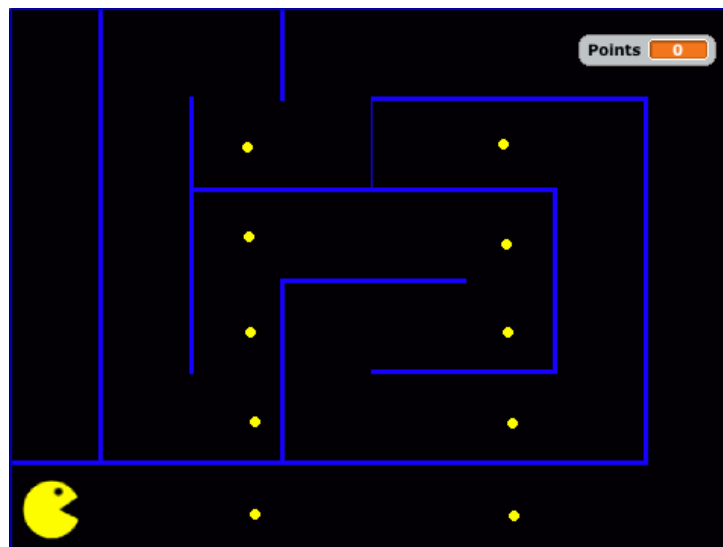


Abbildung 58: PacMan master

Bei dem Vierten und Letzten Kursmodul ist nun die Anforderung, dass ein zufällig gewähltes Labyrinth absolviert werden muss, wodurch iterative Algorithmen also schon im Vorhinein ausscheiden. Hierfür sind klarerweise auch schon recht fundierte Kenntnisse in der Programmierung und mithin auch ein durchdachtes Vorgehen von Nöten. Wichtig ist aber, dass es immer nur einen Weg gibt, der durchfahren werden

kann, sodass eine Orientierung ohne zu aufwändige Programmierung noch möglich ist.

Die Skripte der Sprites für die Punkte sowie der Stage sind unverändert geblieben. Der Sprite „Playground“ beinhaltet nun aber drei unterschiedliche Kostüme für die verschiedenen Labyrinth.



Abbildung 59: Skripte von "Playground"

Ebenso gibt es nun zwei Skripte in diesem Sprite, wobei das Erste dazu dient, dass eines der Labyrinth beim Start zufällig ausgewählt wird. Mit dem Zweiten kann die Spielfigur durch Drücken der Leertaste auf die Startposition zurück gesetzt und das nächste Labyrinth gewählt werden, wodurch das Testen des Algorithmus auf den drei Strecken etwas erleichtert werden soll.

### Einsatzmöglichkeiten

Da dieses Kursmodul schon sehr fordernd ist, würde ich es eher im Bereich der Begabtenförderung ansiedeln, als im allgemeinen Unterricht. Es könnte zum Beispiel als Zusatzaufgabe, als abschließende Gruppenaufgabe oder, bei einer durchwegs sehr guten Gruppe, als normale Übung verwendet werden.

Die Funktionsweise meiner Beispiellösung ist folgende: Die Spielfigur fährt solange geradeaus (Blickrichtung rechts), bis sie auf eine Wand des Labyrinths trifft. Anschließend fährt sie zurück, dreht sich zunächst (Blickrichtung: nach oben) und beginnt dann mit der Endlosschleife. Da der Wert von „check“ bei 0 liegt, dreht sie sich nun eigentlich unnötigerweise zweimal um 180°, bevor sie weiter fährt, da „check“ dabei nicht erhöht wird. Dieser kleine Umweg verkürzt jedoch den Algorithmus ein wenig und stellt sicher, dass auch zu Beginn alle möglichen Wege überprüft werden können.

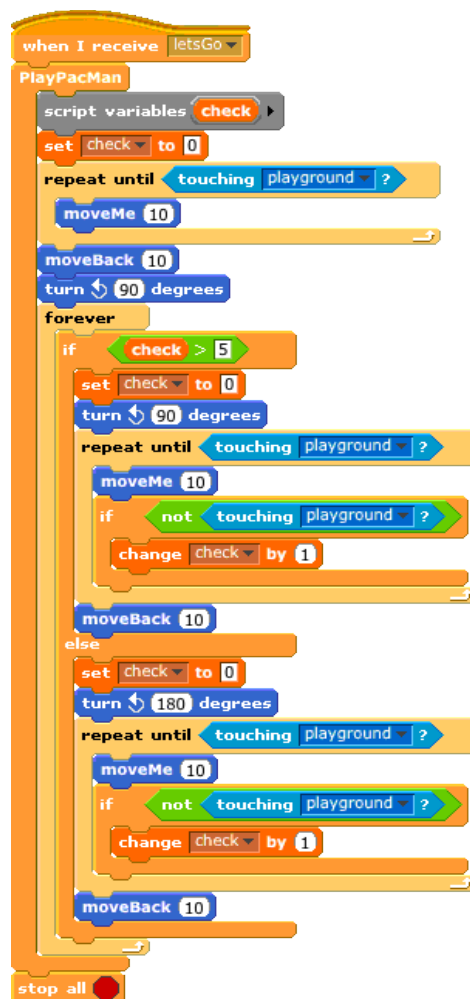


Abbildung 61: Lösungsmöglichkeit "PacMan master"

Die Variable „check“ überschreitet den Wert 5, wenn in ein neues Feld gefahren wird, also ein freier Weg gefunden wurde, da die Breite der Quadrate im Labyrinth 60 Pixel beträgt und nach allen 10 Pixel „check“ inkrementiert wird. Nun wird so lange geradeaus gefahren, bis eine Wand berührt wird, woraufhin der Weg linker

Hand überprüft und wenn möglich sogleich befahren wird. Ist dies nicht möglich, so bleibt als einzige Möglichkeit für ein Weiterkommen der Weg zur Rechten.

#### **4.5 Didaktische Reflexion**

Abschließend zu meinen Ausarbeitungen, möchte ich diese noch aus didaktischer Sicht hinterfragen und kurz durchleuchten. Einer der Punkte, die mir im Spannungsverhältnis zwischen meinen didaktischen Spielen und der konstruktivistischen Didaktik auffiel, ist, dass sich Freiheit und Selbstbestimmung bei den Programmen in einem recht engen Rahmen bewegt. Um hier etwas gegenzusteuern, würde ich zum Abschluss eine Art Kreativ-Modul empfehlen, in dem den Lernenden mehr Freiraum für die Umsetzung ihrer eigenen Ideen eingeräumt wird und wobei sie eigene Projekte finden sollen, die sie, wahlweise alleine oder in kleinen Gruppen, ausarbeiten können. Es sollte dabei ein Rahmen für erforderlichen Umfang und Zeitaufwand vorgegeben werden, doch die Themen der Aufgabenstellungen können eigentlich einen sehr breiten Bereich abdecken und sich von kleinen Tools über Spiele bis hin zu sehr simplen „Robotern“ erstrecken, je nach Gruppengröße und Lernerfolg. Dieses freie Arbeiten wurde von mir auch deshalb an das Ende verschoben, damit das Wissen über die verschiedenen Möglichkeiten, welche die Programmierung mit BYOB bietet bzw. nicht bieten kann, vorhanden ist. Dadurch soll erreicht werden, dass die Übersetzung von einer Idee in die sprachliche Beschreibung des Ablaufs und letztendlich in einen funktionstüchtigen Algorithmus erfolgreich vonstatten geht. Die Rolle der Lehrenden kann hier sehr deutlich auf eine lediglich strukturierende oder dirigierende Position reduziert werden, was sich vor allem dann anbietet, wenn dieses Abschlussmodul gleichzeitig eine Art Test oder Prüfung darstellen soll.

Bezüglich didaktischer Dekonstruktion wäre meine Empfehlung das Mittel der Verfremdung, und zwar in dem Sinne, dass die erlernten Inhalte in ein neues System, hier also eine neue Programmiersprache überführt werden. Es wäre zum Beispiel möglich, einen simplen Algorithmus zuerst in BYOB zu realisieren und diesen im Anschluss daran in eine neue Sprache, etwa Java, zu übersetzen. Die Basis der erlernten Konzepte kann dazu verwendet werden die Syntax von Programmier-

sprachen fokussierter zu lehren, da auf das Vorwissen der Semantik aufgebaut werden kann. Dieser Ansatz wird gerade dann umso wertvoller, wenn für das Lernen von Konzepten eine zentralere Rolle im Bildungssystem gefordert oder gewünscht wird, da das Verständnis grundlegender Konzepte von Skript- und Programmiersprachen, wie bedingte Anweisungen, Schleifen oder Variablen, bereits vorhanden sein sollte.

Die einzelnen Module können auch mit anderen Unterrichtsmethoden verknüpft werden. So würde es sich zum Beispiel anbieten, dass die Module zu den „PacMan“-Kursen in einer Gruppen- oder Projektarbeit bearbeitet werden, die wiederum durch das Konzept von Lee Sheldon umrahmt werden könnten. So wäre es denkbar, dass es Erfahrungspunkte für das Absolvieren eines Labyrinths gibt und die einzelnen Kursmodule in eine Geschichte, ähnlich die eines Rollenspiel eingebettet werden. Die Gruppen der jungen Programmierer könnten in dieser Fantasiereise beispielsweise auf „PacMan“, einem verirrtten gelben Vielfraß treffen, den sie durch die Irrwege leiten müssen. Hierbei können auch alle Schwierigkeitsstufen zur Bearbeitung offen stehen, damit Begabteren eine zusätzliche Beschäftigung geboten wird, während etwas Schwächere mehr Zeit in die Bewältigung der Kernthemen investieren können.

## **5 THEORIE & PRAXIS**

Ich möchte auch noch kurz einige Wort zu dem Zusammenhang von Kersten Reichs Theorie sowie Lee Sheldons Konzept mit der praktischen Umsetzung der Spiele verlieren. Mir geht es dabei einerseits um die für Reich sehr zentralen Begriffe Konstruktion, Rekonstruktion und Dekonstruktion, da sie die wesentlichen Aspekte der konstruktivistischen Didaktik sehr essentiell erfassen und in sich vereinen, andererseits um die Möglichkeit einer Einbettung im Rahmen des didaktischen Entwurfs von Lee Sheldon. Der Grundgedanke hinter meinen Ausarbeitungen war dabei die modulare Zergliederung und Aufbereitung der einzelnen Bereiche, sodass nun die Möglichkeit besteht diese je nach entsprechenden didaktischen Anforderungen und Bedürfnissen zusammen zu fügen und dadurch vor allem jene

vorgestellten Methoden und Konzepte zu verwenden, die sowohl als besonders förderlich für die Bedingungen des Unterrichts erachtet werden, als auch den persönlichen Präferenzen entsprechen, da andernfalls die Gefahr besteht, dass die Authentizität der Lehrenden unterwandert wird.

## **5.1 Konstruktivistische Didaktik**

**Konstruktion:** Ein sehr komplexer Punkt im Rahmen meiner Überlegungen für praktische Lernspiele war genau an dieser Stelle, denn wie könnte es möglich sein, dass so streng funktionale Prinzipien, wie die von Programmiersprachen, mit dem recht freien Grundgedanken eines Leitmotivs wie dem des Erfindens in Einklang gebracht werden, vor allem mit der speziellen Berücksichtigung, dass es sich in dieser Lernsituation um den Einstieg in die Programmierung handelt? Das Problem konnte nicht zur Gänze gelöst werden, einerseits ist das Erfinden einer Lösungsmöglichkeit zwar durch die Selbstbestimmung durchaus gegeben, andererseits wird die Konstruktion jedoch nicht in vollem Umfang ermöglicht, da der Rahmen der Spiele, die auf eine möglichst effiziente Erreichung des Lernzieles hin ausgelegt sind, eine im weiten Rahmen freie Betätigung sehr wohl beschränken. Das Prinzip der Konstruktion wird durch die Aufforderung zur kreativen Lösung und damit zusammenhängend durch das experimentierende Handeln gefördert, da im Großteil der Spiele mehrere Algorithmen gültige Lösungen darstellen, die durch versuchendes Vorgehen im Sinne von „Trial & Error“ entwickelt werden können, wobei durch die Spiele auch der Vorteil geboten ist, dass Fehler hierbei keine Konsequenzen bezüglich der Note mit sich bringen und somit einen förderlichen Beitrag zum Unterricht darstellen, da sie zum einen gute Lehrmeister und zum anderen klare Anzeichen für Un- bzw. Missverständnisse sein können. Jedoch muss ich an dieser Stelle eingestehen, dass wahre Konstruktion beispielsweise erst im Rahmen des Gestaltens eines eigenen Spiels erzielt werden würde, da sich hierbei der erfinderische Geist weitläufiger entfalten kann.

**Rekonstruktion:** Das Entdecken von bereits bekanntem Wissen, dessen Erkundung und Erforschung, ist bei einer Einführung in die Programmierung wohl das

effizienteste Prinzip von Kersten Reich, da es möglichst schnell zu den gewünschten Resultaten führt. Gerade um rezeptiver Wissensvermittlung bei der Rekonstruktion entgegen zu wirken, können Spiele sehr hilfreich sein, da sie eine kreative Art der Selbstbestimmung fördern und ebenso Anreize bieten, die zu einer selbsttätigen Beschäftigung mit der Materie führen. Die möglichen Vorteile bei der Verwendung von Spielen wurde bereits erörtert, doch gerade bei einem Einstieg in die Programmierung, bei dem es vorrangig gilt die Grundstruktur und das allgemeine Vokabular zu erlernen, erachte ich einen spielerischen Zugang für sehr hilfreich, da dadurch das Maß an Überforderung gemildert und bei bedachtem Einsatz sogar die Eigenmotivation der Lernenden gefördert werden kann. Das Leitmotiv des Erkundens kann dabei durch die Verwendung der Spiele mittels deren explorativem Charakter sehr harmonisch vereint und umgesetzt werden, da die Algorithmen und Befehle ja nicht bloß erklärt und vorgeführt, sondern vielmehr in ihrer Funktionsweise entdeckt und nachvollzogen werden. Das Ziel der Spiele ist so gestaltet, dass es sehr schnell einsichtig ist, wodurch eine zu strikte Instruktion vermieden werden kann. Damit soll erzielt werden, dass klar ersichtlich ist welche Aufgabe es zu bewältigen gilt und zugleich soll verhindert werden, dass der eigenständigen Erkundung des Lösungsweges zu enge Grenzen gesetzt sind. Die Rekonstruktion wird also versucht mittels eigenständiger, explorativer Betätigung im Rahmen eines Spiels, das klar erkennbare Grenzen und Ziele hat, die eine Orientierung an einem Lernziel erleichtern, umzusetzen.

**Dekonstruktion:** Die Verwirklichung des dekonstruktivistischen Gedankens in einem Spiel für Anfängerinnen und Anfänger ist meiner Ansicht nach nicht sehr zweckdienlich, da hierfür bereits ein möglichst umfassendes Verständnis benötigt wird, da nur etwas anders gedacht werden kann, was bereits verstanden wurde, oder anders: nur ein bereits verstandener Lösungsweg kann kritisiert und bezweifelt werden, da andernfalls die Basis für derartige Überlegungen fehlt. Dieser Lösungsweg in Form eines passenden Algorithmus soll aber gerade erst durch die Spiele erlernt und verstanden werden. Dekonstruktion kann aber auch auf sehr einfache Weise außerhalb der Spiele Einzug in den Unterricht halten, da mit Sicherheit nicht jeder Lernende die gleiche Lösung hervor bringt und somit stets auch diverse Möglichkeiten zur Diskussion stehen. Da es sich bei den Lösungen aber

um individuelle Kreationen handelt, ist die Methode des Zweifelns in diesem Zusammenhang wohl weniger förderlich, da hierfür Fallbeispiele notwendig wären, die, wenn sie aus der Gruppe der Lernenden entnommen werden, nur allzu leicht die Motivation drastisch senken könnten, sollten die fehlerhaften Versuche Einzelner vor der Gruppe veröffentlicht werden. Dies könnte wiederum zu einer Hemmung der aktiven Teilnahme am Unterricht führen und somit einer konstruktivistischen Didaktik, die sehr zentral auf Selbstbestimmung und Selbsttätigkeit im Lernprozess beruht, im Wege stehen. Allerdings wäre ein respektvoller Vergleich verschiedener Lösungen sehr wohl möglich und durchaus empfehlenswert, da die dekonstruktivistische Grundfrage „Wäre es auch anders möglich?“ (vgl. Reich, 2008, S.141) dadurch auf die Ergebnisse der Lernenden angewandt werden kann. Dazu können auch Vorschläge eingebracht werden, die keine fertige Lösung, sondern nur die Idee für einen Algorithmus darstellen, welche dann gemeinsam erörtert und diskutiert werden kann. Wie bereits erwähnt kann Dekonstruktion aber auch erreicht werden, indem das erlangte Wissen in eine neue Programmiersprache überführt wird, da auch auf diesem Wege das bereits Erlernte in ein neues Spannungsverhältnis gesetzt wird.

## **5.2 Multiplayer Classroom**

Eine didaktische Symbiose des Konzepts von Lee Sheldon mit den Spielen ist ohne große Hindernisse möglich, wobei ich eine solche hier nur im Rahmen möglicher Überlegungen und mithin nicht als eine zwingende Methodik präsentieren möchte, da dies dem Grundgedanken der flexiblen Anpassung an die subjektiven Welten der Lernenden widersprechen würde, wodurch das Konzept des Multiplayer Classroom auch als didaktische Umrahmung des Unterrichts gedacht ist und nicht direkt in die Spiele implementiert wurde. Es können hierfür durchaus mannigfaltige und exotische Fantasiewelten erschaffen werden, die je nach Publikum ihre Berechtigung haben und gerade durch die Anpassung an die Lernenden ihre volle Wirkung entfalten können. Allerdings stellt die Verwendung einer Fantasiewelt, wie bereits erwähnt, keinen Zwang dar, da es natürlich auch möglich ist, dass eine derartige Unterrichtsform auf größeren Widerstand stößt, als das gewohnte Prinzip und somit ungeeignet wäre.

Die Reihe „PacMan“ kann zum Beispiel in die Geschichte um einen bösen Hexenmeister eingebettet werden, der die Avatare der Lernenden mit Hilfe seiner schwarzen Magie in einen Kerker verbannt, aus dem sie zwar ausbrechen können, doch der einzige Weg führt nun durch die verwinkelten Labyrinth, den glücklicherweise der gelbe Helfer namens „PacMan“ finden hilft. Zum Abschluss könnte ein Kampf gegen den finsternen Hexenmeister statt finden, was im Sinne Sheldons eine Prüfung umschreiben würde. Eine solche Geschichte kann natürlich auch über den Stoff eines ganzen Semesters ausgeweitet werden, sofern Zeit und Kreativität dies zulassen.

Solche Fantasiereisen lassen sich recht schnell aus abgewandelten Plots aktueller cineastischer Kassenschlager ableiten und bieten daher auch den Vorteil, dass hierbei an die Erlebniswelten der Lernenden angeknüpft werden kann, wobei, ein entsprechendes Publikum voraus gesetzt, auch andere fiktionale Quellen heran gezogen werden können. Hierbei erweist sich die Ressource Zeit abermals als ein Hindernis, da für die Lehrenden, zusätzlich zu den notwendigen Vorbereitungen, auch noch die Beschäftigung mit den Themenbereichen der Geschichten hinzu kommt. Ich erachte die Aufbereitung eines solchen Schauspiels zwar als eine sehr gute Möglichkeit, aber nicht als zwingend für den Erfolg von Unterricht, da es zusätzliche Anreize bieten kann, aber nicht muss, sollten diese von den Lernenden nicht angenommen werden.

Die Vergabe von Punkten, auch wenn sie nicht als Erfahrungspunkte betitelt werden, halte ich dennoch für sehr wesentlich, da hierdurch eine transparentere Orientierung ermöglicht wird. Eine fixe Vergabe von Punkten ist in den Spielen nicht vorgesehen und daher auch nicht implementiert, da ich der Meinung bin, dass dies zu starre Vorgaben hervor bringen und eine flexible Adaptierung an die verschiedenen möglichen Unterrichtsformen und Bedingungen erschweren würde. Die Benotung mittels Punkten als Erfahrungspunkten kann in ein umrahmendes Rollenspiel eingebettet werden, wobei der aktuelle Stand der individuellen Leistung auch mittels der Stufe der jeweiligen Avatare anstatt mit herkömmlichen Noten präsentiert werden kann<sup>24</sup>. Da BYOB das Importieren eigener Grafiken ermöglicht, könnten die Spiele

---

<sup>24</sup> Für nähere Informationen siehe Kapitel „Der Lehrplan“.

durch eigene Fotos oder Bilder personalisiert werden und die Möglichkeit bieten, dass so eigene Avatarbilder oder Gildenwappen eingebunden werden.

Eine ähnliche Offenheit möchte ich auch bezüglich der Sozialformen (Einzel-, Partner- und Gruppenarbeit) beibehalten. Die Spiele können je nach Leistung der Lernenden mit den verschiedenen Sozialformen kombiniert werden um dadurch eine möglichst effiziente Adaptierung an die Lernpräferenzen sicher zu stellen, die jedoch von Gruppe zu Gruppe sehr stark variieren mögen. In leistungsschwächeren Klassen würde ich persönlich bei der Einführung in die Programmierung mit Hilfe der BYOB Spiele eher zu Gruppen- und Partnerarbeiten tendieren, als in leistungsstarken Klassen. Vor allem Lernende mit „ungünstigen Lernvoraussetzungen benötigen offenbar ein deutlich stärkeres Ausmaß an Anleitung, Unterstützung und Strukturierung“ (Helmke, 2010, S. 70), was erneut darauf verweist, dass eine zu rezepthafte Ausarbeitung und Festlegung der praktischen Spiele deren eigentlichen Sinn verfehlen würde, da gerade die Möglichkeit der Adaptierung an ein Unterrichtsszenario sehr wertvoll ist.

## 6 CONCLUSIO

Spielerische Konzepte und Methoden im Informatikunterricht und speziell für Anfängerinnen und Anfänger in der Programmierung können einen durchwegs positiven Effekt auf den Lernerfolg haben, falls sie zweckdienlich, mit Bedacht und mithin nicht als Selbstzweck eingesetzt werden. Sie können zusätzliche Anreize bieten, für mehr Transparenz sorgen und die Eigeninitiative der Lernenden fördern, doch sind sie keineswegs der „Heilige Gral“ der Didaktik.

Für Lehrende besteht ein erhöhter Aufwand in der Vor- und Nachbereitung ihres Unterrichts, da Planung und Reflexion eine größere Rolle spielen, was auch gleichzeitig mehr Flexibilität fordert, da eine Anpassung an die Bedürfnisse für einen möglichst positiven Lernerfolg unabdingbar zu sein scheint. Zusätzlich sind auch einige der präsentierten Möglichkeiten gerade erst dabei erkundet und analysiert zu werden, wodurch die Menge an empirischen Fakten und Untersuchungen oftmals sehr überschaubar ist, was leider gerade dann noch mehr ins Gewicht fällt, wenn es darum geht das eigene Vorgehen rechtfertigen und begründen zu müssen. Dadurch kann es auch eine etwas abschreckend wirken, gerade im schulischen Betrieb, wenn von Seiten der Eltern und der Schulleitung eine Begründung gefordert wird, speziell bezüglich dem Unterrichtskonzept von Lee Sheldon. Durch sorgfältige Vorbereitung und Dokumentation kann hier aber in vielen Fällen wahrscheinlich ausreichend vorgebeugt werden. Diese Überlegung war für mich unter anderem auch schon im Vorfeld der Ausarbeitungen ein wichtiger Punkt, weshalb ich mich dafür entschied, eine fundierte didaktische Basis als Fundament meiner Untersuchungen zu wählen, da durch diese das eigene Handeln begründet werden kann. Mit gründlicher Planung und dem nötigen Eifer könnte das Konzept von Sheldon also durchaus eine günstige Möglichkeit zur Förderung von Motivation und Reduktion von Komplexität darstellen, so weit es sich an den stichprobenartigen empirischen Belegen ablesen lässt.

Die anfänglichen Bedenken, dass die spielerischen Anreize eine zu stark ablenkende Wirkung erzeugen und somit eine erfolgreiche Symbiose verhindern würden, konnten abgeschwächt werden. Zwar sehe ich diese Bedenken generell noch immer

als gerechtfertigt an, doch nicht als unüberwindbares Hindernis, sondern mehr als mahnenden Begleiter, der dazu eingesetzt werden kann, dass die Vorbereitung sorgfältiger ausgeführt und der Einsatz der Methoden sowie Konzepte reflektierter durchgeführt wird. Es wurde für mich im Laufe meiner Ausarbeitungen auch immer klarer ersichtlich, dass ein Erlernen von grundlegenden Konzepten für den Einstieg in einen Themenbereich sehr wertvoll sein kann, da es eine solide Basis bieten kann, von der aus eine konkrete Anwendung wahrscheinlich einfacher zu realisieren ist, als wenn zeitgleich die Konzepte und deren Umsetzung verstanden werden müssen. Speziell in der Programmierung ist diese Ausgangsbasis umso zweckdienlicher, da gerade hier eine Vielfalt an Sprachen existiert, deren Grundlagen sich meist recht ähnlich sind. Daher erachte ich BYOB für den Einstieg als eine sehr günstige Möglichkeit, der durch den Einsatz von didaktischen Spielen eventuell noch zusätzliche Anreize bezüglich Motivation hinzugefügt werden können. Die Spiele bieten aber auch noch einen anderen Vorteil, den ich nicht unerwähnt lassen möchte, es ist nämlich sehr schnell ersichtlich, was getan werden muss. Bei der Spielreihe „PacMan“ ist zum Beispiel sehr schnell klar wohin unser gelber Freund geführt werden muss, dadurch ist die Aufgabenstellung bereits anschaulich definiert und somit wird eine zusätzliche Hürde zumindest abgeschwächt.

Abschließend möchte ich noch anmerken, dass die Gestaltung von Unterricht, gleich welcher Art sie sein mag, vielleicht gerade dann am wirkungsvollsten ist, wenn sie auch von Seiten der Lehrenden mit Freude am Unterrichten aufbereitet wurde.

## **7 ABSTRACT (DEUTSCH)**

Die Basis meiner Untersuchungen beruht auf den Theorien zur konstruktivistischen Didaktik von Kersten Reich, einem sehr handlungsorientiertem Konzept, dessen zentraler Fokus auf Selbstbestimmung im Lernprozesses liegt. Dies bildet die Basis für die weiteren Reflexionen und kritische Betrachtungen der Möglichkeiten, Spiele im Informatikunterricht, speziell für den Einstieg in die Programmierung, einzubinden.

Darauf basierend wird das Konzept von Lee Sheldon untersucht, der nach dem Spieldesign von „World of Warcraft“, einem Online-Rollenspiel aus dem Hause Blizzard Entertainment, unterrichtet, dessen Elemente er als Umrahmung für den Unterricht übersetzte, wobei die Lernenden in die Rolle von Spielern versetzt werden und der Unterricht nach den Mechanismen ähnlich eines Spiels gestaltet wird. In diesem Kapitel war es ein großes Anliegen, die Funktionsweisen zu analysieren und extrahieren, da dieses Konzept, zumindest den veröffentlichten Fallstudien nach zu urteilen, überwiegend erfolgsversprechend zu sein scheint. Es zeigte sich in der Analyse sehr deutlich, dass die grundlegenden Mechaniken konstruktivistisch geprägt waren, sowie eine weitreichende Überschneidung mit den von Kersten Reich präsentierten Unterrichtsmethoden.

Im dritten Abschnitt wurden Spiele mittels BYOB entwickelt (als CD beigelegt), die einen Einstieg in die Programmierung erleichtern und fördern sollen. Das Ziel hierbei war es, die bereits vorhandenen Vorzüge durch eine klar ersichtliche Aufgabenstellung und motivierende, spielerische Elemente weiter auszubauen, um der Reduktion der Komplexität im Einstieg auch weitere Anreize hinzufügen zu können. Dabei sind die Spiele so entwickelt worden, dass eine Anknüpfung an bereits erlernte Inhalte möglich sowie gewünscht ist.

## **8 ABSTRACT (ENGLISH)**

My research is based on the theories of constructivist didactics by Kersten Reich, a very action-oriented approach. The central focus is the self-determination of the learning process. This theory forms the basis for further reflections and critical observations of possible ways to include games in computer science education, especially concerning getting started with programming.

Based upon the first part the concept of Lee Sheldon is being evaluated. He teaches according to the game design of "World of Warcraft" (an online role-playing game developed by Blizzard Entertainment), whose elements he translated for the use of a setting for his class. The students become players of the classroom and the lessons are designed similar to the mechanisms of a game. It was a great concern to analyze and extract its functioning, as this concept seems to be mostly promising (at least judging by the published case histories). The analysis shows very clearly, that the used mechanics were basically of constructivist nature and furthermore that an overlap with the teaching methods presented by Kersten Reich is obvious.

The third section is concerning the games I developed using BYOB to ease getting started with programming. The goal hereby was to enrich the existing benefits like reduction of complexity by constructing a clearly visible task, utilizing additional motivation factors and therefor game-like elements. The games have been developed in a way that enables a link to the previously learned content.

## 9 LITERATURVERZEICHNIS

- Cypra, O. (2005). *Warum spielen Menschen in virtuellen Welten? Eine empirische Untersuchung zu Online-Rollenspielen und ihren Nutzern*. Diplomarbeit. Johannes Gutenberg Universität Mainz. URL: <http://www.staff.uni-mainz.de/cyprao/diplom-arbeit.pdf> [zuletzt abgerufen am 12. Februar 2013]
- Daferner, P. (2010). *Digital Game Based Learning. Die Begeisterung als Motivationsgrund beim Lernen*. Diplomarbeit. Technische Universität Wien. URL: <http://www.ub.tuwien.ac.at/dipl/2010/AC07807585.pdf> [zuletzt abgerufen am 12. Februar 2013]
- Federici, S. (2011). A minimal, extensible, drag-and-drop implementation of the C programming language. In: *Proceedings of the 2011 conference on Information technology education (SIGITE '11)*. 191-196. West Point, NY: ACM.
- Fox, J., Bailenson, J. & Binney, J. (2009). *Virtual Experiences, Physical Behaviors: The Effect of Presence on Imitation of an Eating Avatar*. Presence: Teleoperators & Virtual Environments. 18(4), 294-303.
- Futschek, G. (2012). *Programmieren lernen mit BYOB*. Vortrag auf der Tagung Informatiktag 2012 der OCG. URL: [http://www.ocg.at/sites/ocg.at/files/medien/pdfs/futschek\\_5\\_11.pdf](http://www.ocg.at/sites/ocg.at/files/medien/pdfs/futschek_5_11.pdf) [zuletzt abgerufen am 09. Dezember 2012].
- Futschek, G. & Moschitz, J. (2011). Learning Algorithmic Thinking with Tangible Objects Eases Transition. In: *Proceedings of the 5th international conference on Informatics in Schools: Situation, Evolution and Perspectives (ISSEP'11)*. LNCS 7013, 155-164. Berlin: Springer Verlag.
- Ducheneaut, N., Wen, M.-H., Yee, N. & Wadley, G. (2009). Body and mind: a study of avatar personalization in three virtual worlds. In: *Proceedings of the SIGCHI*

- Conference on Human Factors in Computing Systems (CHI '09)*. 1151-1160. New York, NY: ACM.
- Harvey, B. & Mönig, J. (o.D.). *BYOB Reference Manual*. URL:  
<http://byob.berkeley.edu/BYOBManual.pdf> [zuletzt abgerufen am 08. November 2012].
- Helmke, A. (2010). *Unterrichtsqualität und Lehrerprofessionalität. Diagnose, Evaluation und Verbesserung des Unterrichts*. Seelze: Kallmeyer Verlag.
- Herkner, W. (2001). *Lehrbuch Sozialpsychologie*. Bern: Huber Verlag.
- Himmelbauer, K. (2011). *Der Stellenwert von Kommunikation im MMORPG World of Warcraft*. Magisterarbeit. Universität Wien. URL:  
[http://othes.univie.ac.at/16136/1/2011-06-20\\_0008405.pdf](http://othes.univie.ac.at/16136/1/2011-06-20_0008405.pdf) [zuletzt abgerufen am 12. Februar 2013]
- Keller, L., Scheuner, B., Serafini, G. & Steffen, B. (2011). A Short Introduction to Classical Cryptology as a Way to Motivate High School Students for Informatics. In: *Proceedings of the 5th international conference on Informatics in Schools: Situation, Evolution and Perspectives (ISSEP'11)*. LNCS 7013, 189-200. Berlin: Springer Verlag.
- Klaß, C. (2005). *Tödliche Erschöpfung durch Dauerspielen*. URL:  
<http://www.golem.de/0511/41466.html> [letzter Zugriff: 08. November 2012].
- König, N. (2011). *The Play Experience - A Constructivist Anthropology on Computer Games*. Dissertation. Universität Wien. URL:  
[http://othes.univie.ac.at/18623/1/2011-08-31\\_9304501.pdf](http://othes.univie.ac.at/18623/1/2011-08-31_9304501.pdf) [zuletzt abgerufen am 12. Februar 2013]
- Lang, H. W. (2006). *Algorithmen in Java*. München: Oldenbourg Wissenschaftsverlag.

- Laster, J. (2010). *At Indiana U., a Class on Game Design Has Students Playing to Win*. URL: <http://chronicle.com/blogs/wiredcampus/at-indiana-u-a-class-on-game-design-has-students-playing-to-win/21981> [zuletzt abgerufen am 08. November 2012].
- Lehner, M. (2012). *Didaktische Reduktion*. Berne: Haupt Verlag.
- Lonati, V., Monga, M., Morpurgo, A. & Torelli, M. (2011). What's the Fun in Informatics? Working to Capture Children and Teachers into the Pleasure of Computing. In: *Proceedings of the 5th international conference on Informatics in Schools: Situation, Evolution and Perspectives (ISSEP'11)*. LNCS 7013, 213-224. Berlin: Springer Verlag.
- Papert, S (1998). *Does Easy Do It? Children, Games, and Learning*. Game Developer magazine. URL: <http://www.papert.org/articles/Doeseasydoit.html> [letzter Zugriff: 22. Oktober 2012].
- Prensky, M. (2001). *Digital Game-Based Learning*. St. Paul, MN: Paragon House.
- Reich, K. (2002). *Grundfehler des Konstruktivismus – Eine Einführung in das konstruktivistische Denken unter Aufnahme von 10 häufig gehörten kritischen Einwänden*. URL: [http://www.uni-koeln.de/hf/konstrukt/reich\\_works/aufsatz/reich\\_39.pdf](http://www.uni-koeln.de/hf/konstrukt/reich_works/aufsatz/reich_39.pdf) [Zuletzt abgerufen am 08. November 2012].
- Reich, K. (2008). *Konstruktivistische Didaktik*. Weinheim und Basel: Beltz Verlag.
- Reich, K. (2010). *Systemisch-konstruktivistische Pädagogik: Einführung in die Grundlagen einer interaktionistisch-konstruktivistischen Pädagogik*. Weinheim und Basel: Beltz Verlag.
- Schmidt, J., Dreyer, S. & Lampert, C. (2008). *Spielen im Netz. Zur Systematisierung des Phänomens "Online-Games"*. Arbeitspapier. Hamburg: Verlag Hans-Bredow-Institut.

- Schüßler, I. (2004). *Lernwirkungen neuer Lernformen*. Studie der Arbeitsgemeinschaft Betriebliche Weiterbildungsforschung e. V.:Berlin. URL: <http://www.abwf.de/content/main/publik/materialien/materialien55.pdf> [letzter Zugriff: 22. Oktober 2012].
- Schüßler, I. (2005). *Paradoxien einer konstruktivistischen Didaktik: Zur Problematik der Übertragung konstruktivistischer Erkenntnisse in didaktische Handlungsmodelle – theoretische und praktische Reflexionen*. In: Report: Zeitschrift für Weiterbildungsforschung. 28(1), 88-94.
- Shaffer, D. W., Squire, K. R., Halverson, R. & Gee, J.P. (2004). *Video games and the future of learning*. Arbeitspapier. Madison, WI: University of Wisconsin-Madison. URL: <http://www.academiccolab.org/resources/gappspaper1.pdf> [letzter Zugriff: 04. November 2012].
- Sheldon, L. (2011). *The Multiplayer Classroom: Designing Coursework as a Game*. Boston, MA: Cengage Learning Verlag.
- Tulodziecki, G., Herzig, B. & Blömeke, S. (2009). *Gestaltung von Unterricht. Eine Einführung in die Didaktik*. Bad Heilbrunn: Julius Klinkhardt Verlag.
- Wallner, F. (1992). *Konstruktion der Realität. Von Wittgenstein zum Konstruktiven Realismus*. Wien: WUV Universitätsverlag.
- Yee, N. (2006). *The Labor of Fun. How Games Blur the Boundaries of Work and Play*. In: Games and Culture. 1(1), 68-71.

## **10 LEBENSLAUF**

### **.:: Persönliche Daten**

|                     |               |
|---------------------|---------------|
| Name                | Andreas Braun |
| Geburtsdatum        | 03. Juli 1983 |
| Staatsangehörigkeit | Österreich    |
| Kontakt             | ab023@gmx.at  |

### **.:: Ausbildung**

|                  |                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>seit 2009</b> | Fortsetzung des Lehramtsstudiums UF Psychologie und Philosophie, UF Informatik und Informatikmanagement                                                                      |
| <b>2008-2009</b> | Beurlaubung vom Studium aufgrund des Zivildiensts im Laura Gatner Haus für unbegleitete minderjährige Flüchtlinge in Hirtenberg (NÖ) inkl. Leitung des Pilotprojektes „ECDL“ |
| <b>2008</b>      | Absolvierung des „Ausbildungskurses zum Begleitlehrer einer Wintersportwoche“ am USI Wien (A2)                                                                               |
| <b>2001-2008</b> | Lehramtsstudium LA Philosophie, Pädagogik, Psychologie UF Informatik und Informatikmanagement an der Universität Wien                                                        |
| <b>1993-2001</b> | Bundesgymnasium und Bundesrealgymnasium Berndorf                                                                                                                             |
| <b>1989-1993</b> | Volksschule St. Veit                                                                                                                                                         |