

game over

Computerspiel als interaktives Medium
für audio-visuelle Kunst

Schriftlicher Teil der künstlerischen Masterarbeit
im Fach Computermusik

Christof Ressi

Matrikelnummer: 812266

Studienkennzahl: V 066 705

Betreuer:

Univ. Prof. Mag. Ph. D. Marko Ciciliani

Univ. Prov. Mag. art. Dr. phil. Christian Utz

Abstract

Im Zentrum dieser Arbeit steht die Frage, inwiefern das Computerspiel als eigenständiges künstlerisches Medium verstanden werden kann und welche Möglichkeiten es für die Gestaltung interaktiver audiovisueller Kunstwerke bietet. Der Autor geht dabei in erster Linie von seiner eigenen Werkreihe *game over* aus und erläutert anhand konkreter Beispiele die spezifischen Eigenschaften des Mediums, insbesondere die Auswirkungen konstitutiver, operativer und impliziter Spielregeln auf das Spielerlebnis und die Hervorbringung emergenter, offener Formen, aber auch die Entstehung diverser virtueller Raumerfahrungen, die Rolle teleologischer Prozesse in Form von Belohnung und Bestrafung sowie die Konstruktion und Destruktion von Spielwelten als formales und performatives Gestaltungsmittel.

This paper deals with the question to what extent computer games can be seen as a distinct artistic medium and which possibilities they open for the creation of interactive audio-visual artworks. It is mainly based on the author's own work series called *game over*, providing various concrete examples for the specific properties of the medium: first of all the effects of constitutive, operational and implicit game rules on the play experience and the development of emergent, open forms, but also the formation of various virtual room experiences, the role of teleological strategies in terms of reward and punishment, and finally the construction and destruction of game worlds as a performative element and means of formal organization.

Inhaltsverzeichnis

Einführung	S. 3
1 <i>game over</i>	S. 6
1.1 Installative Fassung	S. 6
1.2 Konzertante Fassung	S. 7
1.3 Die Game-Engine	S. 8
1.4 Chronologie	S. 10
2 Regeln	S. 12
2.1 Konstitutive Regeln	S. 12
2.1.1 Physics	S. 14
2.1.2 Graphics	S. 15
2.1.3 Audio	S. 15
2.1.3 Behaviour	S. 15
2.2 Operative Regeln	S. 17
2.2.1 Emergenz	S. 19
2.2.2 Feedback	S. 20
2.2.3 Interaktion	S. 21
2.3 Implizite Regeln	S. 22
2.3.1 Exploits	S. 23
2.3.2 Cheats und Easter Eggs	S. 24
2.3.3 Modding	S. 24

3 Form	S. 26
3.1 Offene Form	S. 26
3.2 Raum	S. 28
3.2.1 Die Welt als Partitur	S. 31
3.3 Teleologie	S. 32
3.3.1 Belohnung und Strafe	S. 34
3.4 Konstruktion/Destruktion	S. 35
Schlusswort	S. 37
Literaturverzeichnis	S. 37

Einführung

Die vorliegende Arbeit geht der Frage nach, inwiefern das Computerspiel als eigenständiges künstlerisches Medium verstanden werden kann und welche Möglichkeiten es für die Gestaltung interaktiver audiovisueller Kunstwerke bietet. Ich gehe dabei in erster Linie von meiner eigenen Werkreihe *game over* aus, wo ich anhand konkreter Beispiele die spezifischen Eigenschaften des Mediums erläutern möchte. Dabei werde ich immer wieder Bezüge zu älteren und neueren Computerspielen herstellen, die mich – direkt oder indirekt – zu bestimmten Designentscheidungen oder auch grundsätzlichen Überlegungen inspiriert haben. Darüber hinaus möchte ich auf diverse Probleme und Defizite hinweisen, alternative Lösungen vorschlagen sowie neue Ideen formulieren, die in zukünftige Versionen des Werkes Einzug finden könnten.

Die Arbeit ist in drei größere Abschnitte gegliedert. Der erste Abschnitt gibt zunächst einen kurzen Überblick über das Projekt *game over* und beschreibt insbesondere das ästhetische Konzept, den Entstehungsprozess sowie die technische Umsetzung.

Der zweite Abschnitt befasst sich mit der Struktur von Computerspielen im Allgemeinen sowie mit den Besonderheiten von *game over*. Hier werden zunächst die Auswirkungen expliziter, operativer und impliziter Regeln auf den Spielverlauf und das Spielerlebnis sowie das Wesen der Interaktion zwischen Spieler/in und Spielwelt untersucht.

Der dritte Abschnitt beschreibt, wie die gewählten inneren Strukturen konkrete äußere Formen im Sinne räumlicher und zeitlicher Abläufe hervorbringen. Es werden dabei u. a. folgende Fragen gestellt: Inwiefern verwirklicht das Medium Computerspiel das Ideal des „offenen Werkes“? Wie werden unterschiedlichen Arten von Raumerfahrungen vermittelt? Welche Rolle spielen explizite Spielziele und Anreizsysteme in der formalen Gestaltung? Wie manifestiert sich die Dialektik von Konstruktion und Destruktion im Spielverlauf? Dabei werden bestimmte als typisch angesehene Eigenschaften von Spielen, wie z. B. Wettkampf mit quantifizierbarem Ausgang, kritisch hinterfragt oder negiert und alternativen, offeneren Definitionen von Spiel gegenübergestellt.

Zunächst möchte ich jedoch auf den Begriff des Mediums eingehen, um mögliche Missverständnisse zu vermeiden. Tatsächlich wurde dieser im Laufe der Geschichte von diversen Autor/innen so unterschiedlich verwendet, dass sich ganze Bücher allein mit der Begriffsgenese beschäftigen.¹ Grundsätzlich üben Medien eine vermittelnde Funktion zwischen einem Sender und

1 Z. B. Stefan Hoffman, *Geschichte des Medienbegriffs*, Hamburg 2002.

einem Empfänger aus. Im klassischen objektorientierten Ordnungsprinzip von Harry Pross werden Medien anhand ihrer Produktions- und Rezeptionsbedingungen klassifiziert:²

- primäre Medien: Vermittlung ohne Einsatz von Technik (z. B. Tanz und Gesang)
- sekundäre Medien: Technikeinsatz bei der Produktion (z. B. Buch, Malerei, Fotografie)
- tertiäre Medien: Technikeinsatz bei der Produktion und Rezeption (z. B. Fernsehen, Film, Radio)

Im Computerspiel treffen verschiedene Einzelmedien aufeinander (Bild, Ton, Text), was jedoch auch auf klassische künstlerische Medien wie das Theater, die Oper oder den Film zutrifft. Man könnte einwenden, dass das eigentliche Medium doch der Computer sei, doch es ist meiner Meinung nach nicht korrekt, das technische Empfangsgerät mit dem Medium gleichzusetzen. Wenn man beispielsweise von Fernsehen oder Film als Medien spricht, meint man nicht das TV-Gerät oder den Kino-Projektor selbst – diese können durch andere Geräte ersetzt werden bzw. werden selbst auch für andere Zwecke eingesetzt –, sondern die spezifische Produktions- und Rezeptionsweise. Der Begriff Medium bezeichnet in dieser Denkweise ein Prinzip der Informationsübertragung und die dafür benötigten technischen Geräte können sich mit der Zeit ändern (was nicht heißen soll, dass sich damit nicht auch das Medium verändert – im Gegenteil!). Das Computerspiel unterscheidet sich deutlich von anderen digitalen Inhalten, die mithilfe eines Computers konsumiert werden können. Eine Besonderheit ist die äußerst stark ausgeprägte interaktive Komponente, welche für mich auch der Hauptanreiz war, mich mit diesem Medium näher zu beschäftigen.

Eine sehr allgemeine Definition von Medien liefert Hans Ulrich Reck: „Medium ist immer ein System von Mitteln für die Produktion, Distribution und Rezeption von Zeichen.“³ Ursula Brandstätter weist darauf hin, dass damit der Medienbegriff in die Nähe der Semiotik gerückt wird. Sie sieht die Besonderheit künstlerischer Medien darin, „dass die Art und Weise der Darstellung und der dargestellte Inhalt so unauflöslich aneinander gekoppelt sind, dass jeder Eingriff in die Darstellungsweise unvermeidbar zu einer Änderung der zu vermittelnden Botschaft führt“.⁴

Das Computerspiel entsteht unter speziellen Produktionsbedingungen und wird auf besondere Weise, nämlich interaktiv, rezipiert; es hat seine eigene Kulturgeschichte und bringt entsprechende ästhetische Voraussetzungen mit, denen sich der/die Künstler/in stellen muss. Dem Computerspiel

2 Harry Pross, *Publizistik: Thesen zu einem Grundcolloquium*, Neuwied 1970, S. 129.

3 Hans Ulrich Reck, *Kunst als Medientheorie*. München 2003, S. 507.

4 Ursula Brandstätter, *Grundfragen der Ästhetik*, Köln 2008, S. 121.

wurde jedoch lange Zeit die Möglichkeit abgesprochen, Kunst hervorzubringen. Roger Ebert beispielsweise schrieb:⁵

To my knowledge, no one in or out of the field has ever been able to cite a game worthy of comparison with the great dramatists, poets, filmmakers, novelists and composers. That a game can aspire to artistic importance as a visual experience, I accept. But for most gamers, video games represent a loss of those precious hours we have available to make ourselves more cultured, civilized and empathetic.

Diese Beobachtung ist in Bezug auf den Großteil der kommerziell erhältlichen Spielen nicht unbedingt falsch, rechtfertigt jedoch nicht Eberts Schlussfolgerung, dass Computerspiele niemals Kunst sein können. Er begründet dies u. a. mit einer sehr engen und naiven Definition von Spiel:⁶

One obvious difference between art and games is that you can win a game. It has rules, points, objectives, and an outcome. Santiago might cite a [sic!] immersive game without points or rules, but I would say then it ceases to be a game and becomes a representation of a story, a novel, a play, dance, a film. Those are things you cannot win; you can only experience them.

Tatsächlich muss der Spielbegriff weiter gefasst sein, um Spiele mit künstlerischen Anspruch einerseits produzieren und andererseits auch rezipieren zu können. Wenn ich persönlich von *game over* als Spiel spreche, folge ich keiner bestimmten strengen Definition des Begriffes, sondern spiele mit den kontextabhängigen Assoziationen und Konnotationen in Bezug auf die technische Ebene des Computerspiels, den spielerischen Akt der spontanen oder geplanten Interaktion sowie den offenen Charakter des Gesamtkonzepts. Ich komme immer wieder auf diese Thematik zurück, indem ich untersuche, inwieweit Definitionsversuche anderer Autoren und Autorinnen sich auf mein Projekt anwenden lassen und welcher Erkenntnisgewinn damit erzielt werden kann.

Schließlich verweise ich noch auf die Unterscheidung zwischen *art games* auf der einen Seite und *video game art* auf der anderen Seite: erstere haben zwar einen künstlerischen Anspruch, sind ansonsten jedoch Computerspiele im üblichen Sinne und werden meist auch vermarktet; zweiteres bezeichnet eine Form von Medienkunst, welche Computerspiele als künstlerisches Material verwenden und dabei meist bereits existierende Spiel modifizierten (siehe 2.3.3). *game over* lässt sich weder hier noch dort recht einordnen; es ist aber auf jeden Fall der Versuch, das Computerspiel als Medium mit spezifischen Eigenschaften zu betrachten, in welchem sich bestimmte künstlerische Vorstellungen am besten umsetzen lassen.

5 Roger Ebert, *Why did the chicken cross the genders?* in: Chicago Sun-Times, Ausgabe vom 27. 11. 2005.

6 Roger Ebert, *Video games can never be art* in: Chicago Sun-Times, Ausgabe vom 16. 4. 2010.

1 *game over*

Mit *game over* bezeichne ich einerseits eine von mir entwickelte 2D-Game-Engine,⁷ andererseits die damit realisierten Konzerte, Performances, Installationen etc. Die begriffliche Unschärfe hat einerseits pragmatische Gründe, andererseits möchte ich damit die konzeptuelle Einheit unterstreichen: Jede Aufführung bzw. Präsentation ist nur eine von vielen möglichen Manifestationen derselben Grundidee und benötigt deshalb keinen eigenständigen Titel. Meist wähle ich eine fortlaufende, aber willkürliche Versionsnummer (z. B. *game over 2.1.7*), um den Charakter des "work in progress" zu unterstreichen. Das Projekt wurde im Rahmen des künstlerischen Forschungsprojekts GAPPP⁸ entwickelt.

In *game over* navigiert ein/e Spieler/in einen Avatar durch eine virtuelle zweidimensionale Welt und erforscht dabei die unterschiedlichen Interaktionsmöglichkeiten sowie die daraus resultierenden Klänge und Bilder. Es gibt keine explizit vorgegebenen Regeln oder Ziele, der/die Spieler/in wird in die Welt „geworfen“ und muss sich selbst darin zurechtfinden. Das Konzept kann, wie bereits angedeutet, auf verschiedene Arten realisiert werden:

1.1 Installative Fassung

Hier sind mehrere „Spiel-Stationen“ im Raum verteilt, mit denen das Publikum nach Belieben interagieren kann. Als Eingabegerät dient dabei meist ein klassischer Game-Controller. In einer bestimmten Version wurden beispielsweise alte TV-Geräte und abgewetzte Sofas verwendet; die Spielstationen wurden zudem von LED-Scheinwerfern beleuchtet, welche anhand der aktuellen Position des jeweiligen Spielers / der jeweiligen Spielerin innerhalb der Spielwelt die Farbe ändern und bei bestimmten Handlungen visuelle Akzente setzen. Die Idee bestand darin, das Publikum in die Inszenierung zu integrieren und es damit quasi zum Teil der Ausstellung zu machen. Jedes Spiel war aus drei verschiedenen Quellen zu hören: erstens über Kopfhörer (welche für den/die Spieler/in selbst bestimmt waren), zweitens aus den Lautsprechern des Fernsehers (lokale Klangquelle) und drittens mithilfe einer Lautsprecherkuppel in den Saal projiziert (siehe 2.1).

⁷ Unter einer Game-Engine versteht man ein Framework (oft inklusive eigener Entwicklungsumgebung), welches die technische Umsetzung von Computerspielen erheblich erleichtert. Bekannte Game-Engines sind u. a. Unreal, Unity, CryEngine, Ogre, Cocos2d und GameMaker.

⁸ „GAPPP: Gamified Audiovisual Performance and Performance Practice.“, PEEK AR 364-G24, www.gappp.net.



Abb. 1: Installative Fassung von *game over*.

1.2 Konzertante Fassung

Hier kann das Spiel einerseits von einem/einer Performer/in auf der Tastatur oder mit einem Game-Controller gespielt werden, andererseits aber auch von einem Instrumentalisten oder einer Instrumentalistin. Letzteres funktioniert so, dass mittels eines Bewegungssensors die Orientierung und Bewegung des Instruments bestimmt sowie der von mehreren Mikrofonen aufgenommene Klang des Instruments auf bestimmte Eigenschaften analysiert (Tonhöhe, Lautstärke, Geräuschhaftigkeit, spektrale Gewichtung, Attack/Release etc.) wird. Die damit gewonnenen Daten werden je nach Spielwelt in unterschiedliche Spielbefehle übersetzt. Der/die Instrumentalist/in hat die Aufgabe, frei auf seinem/ihrem Instrument zu improvisieren, während er/sie gleichzeitig mit der Spielwelt interagiert (die vielfältigen Interaktionsmöglichkeiten werden noch genauer im zweiten Kapitel behandelt). Ich arbeite meist mit dem Klarinetten Szilard Benes zusammen, der in dieser Hinsicht bereits eine beachtliche Virtuosität erreicht hat. Ich versuche, die Spielwelten soweit wie möglich auf ihn „zuzuschneiden“, sodass er problemlos innerhalb der Welt navigieren und seine musikalischen Vorstellungen anhand der von mir zur Verfügung gestellten Mittel überzeugend

ausführen kann. Gleichzeitig baue ich immer wieder Dinge ein, die er noch nie zuvor gesehen hat, um in den Konzerten ein hohes Maß an Spontaneität zu erhalten. Eine solche Zusammenarbeit erfordert ein hohes Maß an gegenseitigem Vertrauen. Während dem Konzerte hab ich die Möglichkeit, quasi von außen in das Spielgeschehen einzugreifen, einerseits um den/die Spieler/in aus schwierigen Lagen zu befreien, andererseits um dem Spiel eine andere Wendung zu geben: so kann ich beispielsweise bestimmte Events auslösen, Eigenschaften ändern oder den/die Spieler/in an einen anderen Ort teleportieren.



Abb. 2: Szilard Benes in der konzertanten Fassung von *game over*. Der Bewegungssensor befindet sich über dem Schalltrichter der Klarinette.

1.3 Die Game-Engine

Die Struktur der Engine hat weitreichende Konsequenzen für die damit entwickelten „Werke“. Grundsätzlich handelt es um eine 2D-Engine, die auf einem gleichmäßigen Raster aus Kacheln (sogenannten *tiles*) basiert. Dieses Design ist typisch für klassische 2D-Spiele unterschiedlicher Genres (Plattformen, RPGs, Simulationen etc.). Die Beschränkung auf zwei Dimensionen hat in erster Linie pragmatische Gründe, eröffnet aber gleichzeitig bestimmte ästhetische Möglichkeiten

(siehe 3.2). Aufgabe der Engine ist die Ausführung der Spiellogik, das regelmäßige Updaten der Spielkomponenten, das Lesen von Spieler-Input, das Senden und Interpretieren von Events, die Darstellung von Grafiken und das Abspielen/Erzeugen von Klängen.

Das Rendering erfolgt mithilfe der Programmbibliothek *openFrameworks*⁹ und basiert auf der Grafik-API *OpenGL*. Für die Klangsynthese verwende ich die grafische Programmierumgebung *Pure Data*,¹⁰ wobei ich Teile der Synthese mithilfe eigener Erweiterungen (sogenannter *externals*) umsetze; darunter befindet sich auch ein Softwareemulator des klassischen D-110-Soundmoduls von Roland, welcher für das Playback von MIDI-Files zuständig ist.

Während die Engine selbst in C++ programmiert ist, wird die Spiellogik in der Skriptingsprache Lua geschrieben, welche traditionell sehr beliebt unter Spieleentwicklern ist. Grund dafür ist die geringe Binary-Größe (knapp 300 kB für Release-Builds ohne Debug-Symbole!) sowie die einfache Einbettung in andere Programme. Ich persönlich verwende die Bibliothek *sol2*¹¹, welche die Arbeit mit Lua innerhalb von C++ erheblich erleichtert.

Für *game over* habe ich zudem einen einfachen grafischen Leveleditor in Python entwickelt, mit dessen Hilfe man sehr schnell neue „Welten“ erstellen kann.

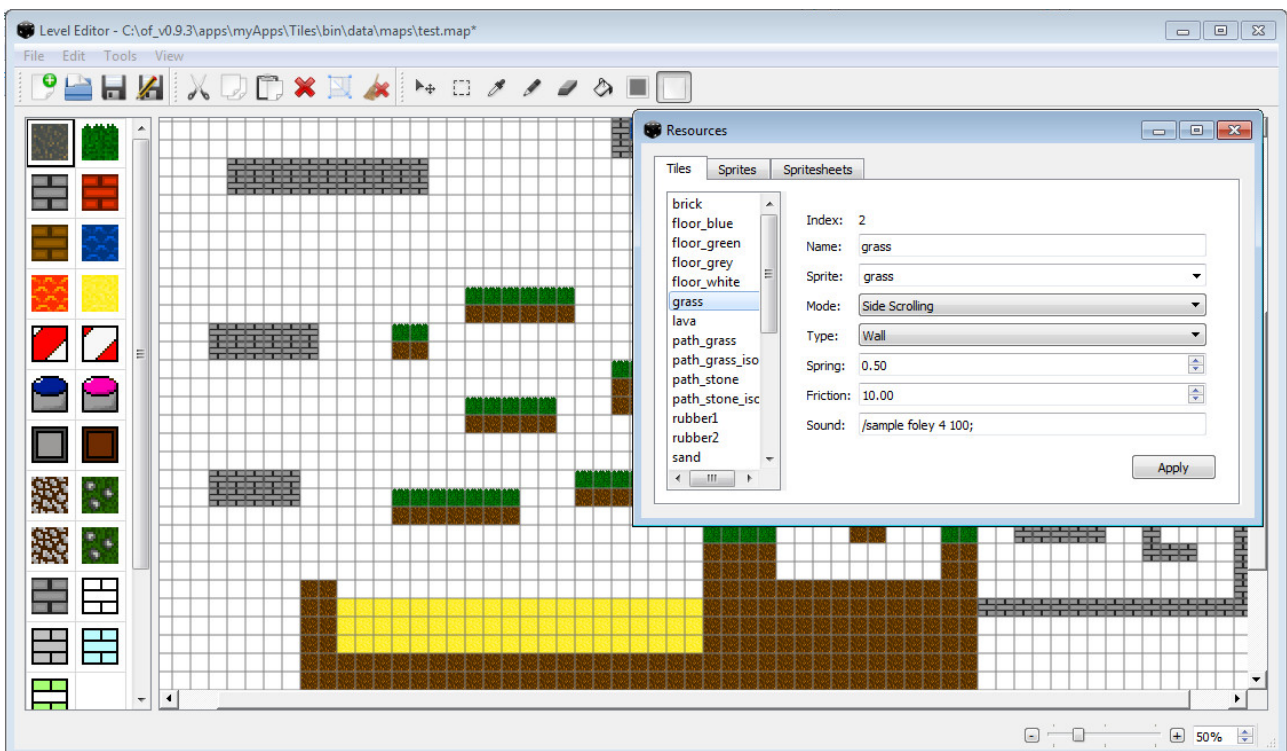


Abb 3: Leveleditor für *game over*.

⁹ Siehe <https://www.openframeworks.cc>

¹⁰ Siehe <https://puredata.info>

¹¹ Siehe <https://github.com/ThePhD/sol2>

1.4 Chronologie

Seit dem ersten Prototypen von *game over* und dem Erscheinen dieser Arbeit sind inzwischen fast zwei Jahre vergangen. Die folgende chronologische Auflistung soll einen Überblick darüber geben, wie sich das Projekt im Laufe der Zeit verändert hat:

November 2016	Erster Prototyp in der Art klassischer Shoot'em Ups wie <i>Space Invaders</i> . Klangerzeugung mithilfe eines original D110-Soundmoduls von Roland.
Dezember 2016	Einführung von JSON-Files für die Konfiguration der Levels.
Januar 2016	Uraufführung der Solo-Version im Rahmen eines Klassenabends im IEM-Cube, Graz.
Februar 2017	Erste GAPPP-Arbeitsperiode. Version für Klarinette mit Szilard Benes.
März 2017	Aufführung im Echo-Raum in Wien.
April-Mai 2017	Arbeit an einem Software-Emulator des D110 aufgrund der Unzuverlässigkeit und Beschränkungen der Hardware.
Juni 2017	Teilnahme an der Konferenz <i>klingt gut! Symposium on Sound</i> in Hamburg. <i>Student Award for Excellence in the Application of New Technologies in Sound Art and Sound Design</i> .
Juli 2017	Aufführung mit Szilard Benes im Rahmen der ASEA-UNINET-Konferenz im Mumuth, Graz.
August-September 2017	Arbeit an einer neuen Engine inkl. eines Prototypen im Platformer-Stil. Präsentation im Rahmen der zweiten GAPPP-Arbeitsperiode gemeinsam mit Matej Bunderla (Sopransaxofon) und Szilard Benes. Verwendung von Lua als Skriptingsprache für <i>game over</i> .
Oktober-Dezember 2017	Loslösung vom reinen Platformer durch Einführung zusätzlicher Tile-Modes (Top-Down und Isometric View). Erstellung mehrere neuer Welten und Implementierung zusätzlicher Synthesemethoden.
Januar 2018	Arbeit an einem Leveleditor. Präsentation der neuen Version mit Szilard Benes in einem Klassenabend im IEM-Cube, Graz.
Februar 2018	Mehrere Verbesserungen (vor allem Fehlerbehandlung). Kurze Präsentation im Rahmen von GAPPP.
März 2018	Einführung von Navigations-Graphen und zahlreiche Verbesserungen.
April 2018	Einführung von Lua-Events (generische Event-Typen, welche in Lua definiert werden können) für bessere Kommunikation zwischen Game-Objekten. Implementierung von Path-Following und Steering-Behaviours. Strikte Trennung zwischen World und View.

Mai 2018	Portierung der Engine auf den Raspberry Pi 3. Zahlreiche Optimierung (vor allem im Bereich Rendering und Kollisionserkennung). Unterschiedliche Update-Raten für die verschiedenen Komponenten.
Juni 2018	Präsentation der installativen Version im Rahmen der Konferenz <i>NIME 2018</i> an der Virginia Tech in Blacksburg, USA.
Juli 2018	Präsentation einer neuen konzertanten und großen installativen Version für das Masterabschlusskonzert im Fach Computermusik im Mumuth, Graz.
August 2018	Beschränkung der Simulation auf einen bestimmten Radius rund um den sichtbaren Bereich, um sehr große Welten ohne CPU-Overhead zu ermöglichen.
September 2018	Präsentation der installativen Version im Rahmen der <i>Game Dev Days Graz 2018</i> .

2 Regeln

Creating a game means designing a structure that will play out in complex and unpredictable ways, a space of possible action that players explore as they take part in your game. [...] You only can indirectly construct the space of possibility, through the rules you design. Game design is an act of faith-in your rules, in your players, in your game itself.¹²

Die einzigartige Struktur eines Spiels wird in starkem Maße von seinen Regeln bestimmt. In ihrem Buch *Rules of Play* unterscheiden Salen und Zimmermann Spielregeln anhand folgender drei Kategorien:¹³

- konstitutiv (die dem Spiel zugrunde liegenden logisch-mathematischen Strukturen)
- operativ (die offiziellen Richtlinien, denen die Spieler/innen folgen müssen)
- implizit (die ungeschriebenen Regeln, z. B. Etiquette, sportliches Verhalten)

Jesse Schell nimmt in *The Art of Game Design* eine ähnliche Einteilung frei nach David Parlett vor:¹⁴ *foundational rules*, *operational rules* und *behavioural rules*. Dazu kommen *written rules* (explizit aufgeschriebene Spielanweisungen, eine Unterkategorie von *operational rules*) und *laws* (explizit formulierte Verhaltensregeln, ein Teil der *behavioural rules*), welche beide in *official rules* (von einer autoritativen Instanz beschlossene Regeln) münden können. Dazu kommen noch *advisory rules* (unverbindliche Vorschläge für strategisch günstiges Spielverhalten) sowie *house rules* (von den Spielern/Spielerinnen selbst beschlossene Änderung bzw. Anpassung einer oder mehrerer vorher genannter Regeln). Für meine Zwecke genügt zunächst die Systematik von Salen und Zimmermann, da es im Falle von *game over* keine schriftlich formulierten, "offiziellen" Regeln existieren, und *advisory rules* und *house rules* als Teil der operativen und impliziten Regeln behandelt werden können.

2.1 Konstitutive Regeln

Diese bestehen in *game over* aus der Spielphysik und dem eigenständigen Verhalten von NPCs (*non player characters*); aus dem Zusammentreffen von Spieler/innen, NPCs und physikalischer Welt entstehen zahlreiche Game-Events, welche wiederum diverse physikalische, verhaltensbezogene, visuelle und klangliche Konsequenzen nach sich ziehen. Diese verschiedenen Ebenen werden

12 Katie Salen und Eric Zimmerman, *Rules of Play*, Cambridge 2004, Kapitel 6, S. 11.

13 Ebenda, Kapitel 12, S. 4 ff.

14 Jesse Schell, *The Art of Game Design. A Book of Lenses*, New York 2008, S. 145 ff.

innerhalb der Game-Engine konzeptuell getrennt, indem jedes Game-Objekt sich aus vier verschiedenen Komponenten zusammensetzt (*behaviour*, *physics*, *graphics*, *audio*), welche sich einerseits durch ihre Eigenschaften, andererseits durch ihr Event-Handling definieren. Der folgende Code zeigt eine minimalistische Definition einer Game-Objekt-Klasse:

```
-- hero
actor {
  name = "hero",
  proto = "default",
  behaviour = {
    walkForce = 30,
    jumpForce = 90,
    jumpTime = 0.4
  },
  physics = {
    mass = 1,
    size = 0.9,
    health = 100,
    lifeTime = 1e007,
    collisionCost = 1,
    friction = 10
  },
  graphics = {
    sprite = "green",
    size = 1,
    -- event handlers
    ActorDied = function (self)
      world:spawnEntity("blood", self.pos, PLAYER_NONE, NONE, nil)
    end
  },
  audio = {
    -- event handlers
    ActorDied = function (self)
      self:play("/sample SID 17 90;")
    end
  }
}
```

Abb. 4: Definition einer Game-Objekt-Klasse.

Das System ist objektorientiert, d. h. Klassen können die Eigenschaften einer anderen Klassen erben (ausgedrückt durch die „proto“-Eigenschaft). Auch sogenannte Mix-Ins sind möglich, indem jede Komponente die Eigenschaften einer anderen Klasse erbt. Es ist also möglich, mit wenigen Zeilen Code eine große Anzahl unterschiedlicher Game-Objekte zu definieren und somit eine hohe Variabilität zu erreichen. Nun folgt eine knappe Beschreibung der oben bereits erwähnten vier Ebenen:

2.1.1 Physics

In den meisten Welten herrschen physikalische Gesetzmäßigkeiten, die u. a. durch folgende Eigenschaften bestimmt werden:

- Stärke und Richtung der Gravitation (falls vorhanden)
- Reibungskoeffizienten der einzelnen Kacheln und Objekte
- Luft- und Wasserwiderstand
- Masse, maximale Beschleunigung und maximale Geschwindigkeit beweglicher Objekte
- Elastizität der Kacheln und Objekte
- Schadenkoeffizienten bei Zusammenstößen
- Kollisionserkennung- und Auflösung

Nicht alle Eigenschaften sind ständig präsent: z. B. herrscht im sogenannten Top-Down-Modus (Vogelperspektive) weder Gravitation noch Luftwiderstand. Manche Eigenschaften werden aus pragmatischen Gründen für bestimmte Objekte außer Kraft gesetzt: Pick-Ups schweben in der Luft, Pistolenkugeln fliegen ungebremst durch die Luft und Geister bleiben von Kollisionen unberührt. Die Wahl der physikalischen Parameter hat einen dramatischen Einfluss auf das gesamte Spielverhalten und muss deshalb sorgfältig getroffen und bei Bedarf angepasst werden. Extreme Einstellungen können jedoch zu interessanten Effekten führen, z. B. wenn plötzlich die Gravitation außer Kraft gesetzt wird oder ein sehr starker Wasserwiderstand das Spiel in Melasse-artiger Langsamkeit versinken lässt.

2.1.2 Graphics

Die Grafikkomponente definiert, wie ein Objekt dargestellt werden soll, wenn es sich in einem bestimmten Zustand befindet; dieser ändert sich meist mit bestimmten Events (z. B. Wechsel in einen anderen Spielmodus, Kollision mit einem anderen Objekt, Änderung der Bewegungsrichtung etc.). Die Darstellung kann sich aber auch selbstständig und nach eigenen Gesetzmäßigkeiten ändern, z. B. periodisch an- und abschwellen, zufällig die Farbe ändern oder mit der Zeit ausfaden. Zudem reagiert die Grafikkomponente auf bestimmte Events, indem sie neue Objekte, wie z. B. Explosionen oder Partikel, erzeugt. Obwohl diese keine direkten Auswirkungen auf die Spielwelt haben, haben sie oft hohe Relevanz für den/die Spieler/in, indem sie den Erfolg (bzw. Misserfolg) von Spielhandlungen anzeigen oder eigentlich „unsichtbare“ Abläufe und Ereignisse visualisieren.

2.1.3 Audio

Analog zur Grafikkomponente, bestimmt die Klangkomponente, wie ein Objekt in gewissen Zuständen klingen soll. Technisch wird das so gelöst, dass OSC-Messages über das lokale Netzwerk an eine Pure-Data-Instanz geschickt werden. Die Messages haben folgenden Aufbau:

1) Adresse, 2) Timestamp, 3) x-Koordinate, 4) y-Koordinate, 5) FUDI-String

Der Timestamp wird verwendet, um manche Audioevents bei Bedarf zu verzögern, die x- und y-Koordinate wird für die Spatialisierung verwendet und der FUDI-String bestimmt, welche Aktionen in Pd ausgeführt werden sollen (`/sample SID 17 90;` beispielsweise spielt das Sample Nr. 17 der Kategorie „SID“ mit einer Lautstärke von 90 ab). OSC-Messages können entweder zu bestimmten Events (Objekt-Kollision, Teleport, Tod etc.) oder selbstständig gesendet werden (z. B. in periodischen Intervallen, abhängig von der Gesundheit oder proportional zur Bewegungsgeschwindigkeit). Der Klangkomponente kommt – besonders in der performativen Variante von *game over* – eine besondere Rolle zu, welche im Abschnitt über Interaktion näher erläutert werden soll.

2.1.4 Behaviour

Die Verhaltenskomponente ist gewissermaßen die komplexeste von allen, da diese weder an einen – mehr oder weniger – realistischen Kontext gebunden ist (wie die Physikkomponente), noch ein bestimmtes Medium bedient (wie die Klang- oder Grafikkomponente). Grundsätzlich ist alles

möglich, jedoch lassen sich grob drei Bereiche definieren, in denen die Verhaltenskomponente operiert:

- Reaktion auf Events (z. B. Erzeugen neuer Objekte bei Kollision)
- selbstständige Aktionen (z. B. regelmäßiges Erzeugen neuer Objekte)
- *Steering-Behaviours* (*Wander*, *Seek*, *Flee*, *Arrive*, *Pursuit*, *Evade* etc.), *Wall-Avoidance* und *Path-Following*

Steering-Behaviours sind besonders gut geeignet, emergentes Verhalten (siehe 2.2.1) zu erzeugen, da mithilfe weniger Parameter recht komplexe Ergebnisse erzielt werden können und bereits geringe Änderungen große Auswirkungen haben können. *Wander* beschreibt einen cleveren Algorithmus, der überzeugende zufällige Bewegungsmuster erzeugt. *Seek* beschreibt das Zusteuern auf ein Ziel (*Arrive* ist eine Variante, die bei kürzer werdender Entfernung abbremst) und *Flee* das Wegbewegen von einem Ziel, *Pursuit* und *Evade* beziehen den aktuellen Geschwindigkeitsvektor des Zielobjekts mit ein, um die zukünftige Position zu schätzen.¹⁵ Handelt es sich bei dem Ziel um den/die Spieler/in, entsteht eine besondere interaktive Dynamik. *Steering-Behaviours* können selbstverständlich auch kombiniert werden oder sich abwechseln; die Parameter können sich während des Spielverlaufs ändern (zu bestimmten Events oder anhand von Objekteigenschaften), sodass mit einfachen Mitteln komplexe Verhaltensmuster entstehen können.¹⁶

Wall-Avoidance ist ein Algorithmus, der Objekte daran hindert, mit Wänden zu kollidieren, indem Ray- bzw. Shape-Casting¹⁷ mit dem *Flee*-Algorithmus kombiniert wird.¹⁸

Path-Following bezeichnet einen Algorithmus, bei dem Objekte einem vorgegebenen Pfad folgen, indem der *Seek*- oder *Arrive*-Algorithmus auf eine Folge von Punkten im Raum angewendet wird.¹⁹

Um das Zusammenspiel der verschiedenen Komponenten zu demonstrieren, möchte ich nun ein konkretes Beispiel aus *game over* geben:

Der/die Spieler/in hat die Möglichkeit, durch absichtliche Kollision mit einer sogenannten Spawnbox vogelartige Akteure in die Welt zu setzen. Diese bewegen sich, ohne Gravitation, mithilfe des *Wander*-Algorithmus zufällig innerhalb der Welt. Befindet sich der/die Spieler/in innerhalb eines bestimmten Radius, flüchten die Akteure vor ihm/ihr. Die Grafikkomponente zeigt

¹⁵ Vgl. Matt Buckland, *Programming Game AI by Example*, Texas 2005, S. 91 ff.

¹⁶ Vgl. ebenda, S. 119 ff.

¹⁷ Siehe https://en.wikipedia.org/wiki/Ray_casting

¹⁸ Vgl. Buckland, *Programming Game AI by Example*, S. 104 ff.

¹⁹ Vgl. ebenda, S. 110 f.

eine Sprite-Animation,²⁰ die abhängig von der Bewegungsrichtung gespiegelt wird. Die Audio-Komponente sendet in zunächst regelmäßigen Abständen Nachrichten, welche ein bestimmtes Sample abspielen. Abspielposition, Länge und Lautstärke sind zufälligen Schwankungen unterworfen. Die Abspielgeschwindigkeit (welche die Tonhöhe bestimmt) wird bei der „Geburt“ zufällig festgelegt und bleibt anschließend immer dieselbe. Die Zeitdifferenz zwischen den einzelnen Samples sinkt proportional zur aktuellen Bewegungsgeschwindigkeit, hängt also von der Physikkomponente ab. Diese sorgt auch dafür, dass dem Akteur bei Kollisionen mit dem/der Spieler/in, abhängig von der Kollisionsstärke, Gesundheitspunkte abgezogen werden. Erreicht die Gesundheit den Wert 0, stirbt das Wesen. Die Gesundheit wiederum hat Einfluss auf die Art der Klangerzeugung: je niedriger sie ist, umso höher ist die Wahrscheinlichkeit, dass die Tonwiederholung durch geräuschhafte Klänge unterbrochen wird. Der Unterschied zwischen tonalen und geräuschhaften Klängen wird durch unterschiedliche Grafikpartikel verdeutlicht, sodass sich der Gesundheitszustand sowohl klanglich als auch visuell ablesen lässt. Zudem hat die Gesundheit Einfluss auf Parameter des Steering-Behaviours: Akteure bewegen sich mit sinkender Gesundheit immer chaotischer.

Dieses im Prinzip recht simple und einleuchtende Verhalten kann vom Spieler / von der Spielerin ausgenutzt werden, um bestimmte musikalische Strukturen zu erzeugen: Wenn man die Vögel „jagt“, bewegen sie sich schneller und erzeugen so dichtere Klänge; allerdings fliehen sie in verschiedene Richtungen, sodass man sie in eine Ecke treiben muss, will man nicht, dass sie sich in alle Himmelsrichtungen zerstreuen. Es ist auch möglich, einzelne unerwünschte Tonhöhen zu entfernen, indem man den entsprechenden Akteur gezielt verjagt. Um mehr klangliche Varianz zu erreichen, kann man gezielt mit Vögeln kollidieren, um ihre Gesundheit zu reduzieren. Dies ist allerdings nicht so einfach und erfordert einiges an Geduld und Geschick. Außerdem muss die Stärke und Anzahl der Kollisionen an den aktuellen Gesundheitsstatus angepasst werden, wenn der Akteur am Leben bleiben soll.

2.2 Operative Regeln

Jesse Schell definiert diese salopp als „what the players do to play the game“,²¹ d. h. die physischen Aktionen, welche Spieler/innen ausführen müssen, um bestimmte Resultate innerhalb des Spiels zu erreichen. Die Anzahl an erlaubten Aktionen ist notwendigerweise begrenzt, sodass der/die

²⁰ Eine rasche Abfolge von Bildern, ähnlich der Zeichentricktechnik.

Siehe [https://de.wikipedia.org/wiki/Sprite_\(Computergrafik\)](https://de.wikipedia.org/wiki/Sprite_(Computergrafik))

²¹ Schell, *The Art of Game Design*, S. 145.

Spieler/in eine Wahl treffen muss. Dabei kann man zwischen einer Mikro-Ebene und einer Makro-Ebene unterscheiden:

The micro level represents the small, moment-to-moment choices a player is confronted with during a game. The macro level of choice represents the way these micro-choices join together like a chain to form a larger trajectory of experience.²²

Salen und Zimmerman betonen, dass der Entscheidungsprozess nicht notwendiger rational sein muss, sondern reflexartig, willkürlich oder intuitiv.²³ Dies trifft besonders auf die Mikroebene zu, z. B. das Drücken des Abzugs in einem First-Person-Shooter, das Treten eines Fußballs oder das Rollen eines Würfels. Bei *game over* werden die meisten Entscheidungen, sowohl auf der Mikro- als auch auf der Makroebene, in erster Linie aus musikalischen Gründen getroffen und sind somit niemals vollständig rational begründet. Selbst banale Aktionen wie das Springen von einer Plattform zur nächsten können durch entsprechende Klänge derart musikalisiert werden, dass das Mittel zum Zweck wird und das Erzeugung interessanter musikalischer Strukturen wichtiger wird als die Fortbewegung von A nach B. Diese Phänomen beschreibt Constantino Oliva anhand seiner persönlichen Erfahrung mit *Super Mario World*:

I remember my younger self playing the game, and being fascinated by the glissando sound of Mario's jumping. [...] Eventually, my fascination with that specific sound would suggest me to just jump on the spot, repeatedly, as fast as possible. While doing that, I would disregard, for a moment, all the challenges of the game.²⁴

Der Möglichkeitsraum physischer Interaktion wird über das physikalische Interface definiert und Parameter müssen mithilfe eines *Mappings* in Game-Events übersetzt werden. Dies ist im Falle eines gewöhnlichen Game-Controllers recht trivial, wirft bei einem Instrument aber einige Fragen auf. In *game over* wird das derzeit so gelöst, dass die Orientierung des Instruments (gemessen von einem Bewegungssensor) konzeptuell einem Steuerkreuz entspricht: befinden sich der Pitch- oder Roll-Winkel über einem gewissen Schwellwert, entspricht dies einer gedrückten Taste des Steuerkreuzes. Da auch der Klang des Instruments analysiert wird, können bestimmte klangliche Features zusätzliche Game-Events erzeugen. Diese können entweder kontinuierlich sein, wie etwa bei Tonhöhe, Lautstärke, *spectral flatness* oder *spectral centroid*, oder aber singulär, wie im Falle der *onset detection*. Die installative Version von *game over* verwendet meist handelsübliche Nachbauten von NES-Controllern. Diese weisen eine sehr begrenzte Anzahl an Steuerungselementen auf: ein Steuerkreuz, zwei Aktions-Tasten (*A* und *B*) sowie zwei Menü-Tasten

22 Salen und Zimmerman, *The Rules of Play*, Kapitel 6, S. 5.

23 Ebenda, S. 6.

24 Constantino Oliva, *On the ontological status of musical actions in digital games*, Krakau 2017, S. 1 f.

(*start* und *select*). Diese beschränkte Konfiguration erweist sich jedoch als großer Vorteil, da sie sehr einfach, übersichtlich und bewährt ist und zudem vielen Menschen bereits vertraut.

2.2.1 Emergenz

Schell beschreibt in *The Lens of Emergence*²⁵ fünf Möglichkeiten, durch Anpassung der Interaktionsmöglichkeiten ein reichhaltigeres Spielverhalten zu erzeugen. Er verwendet dabei den Begriff des Verbs als Metapher für eine Spielaktion.²⁶

- *Add more verbs.*

Mehr operative Aktionen ermöglichen eine größere Anzahl an Resultaten, allerdings kann eine Überzahl an physisch-operativen Möglichkeiten schnell unübersichtlich und in Folge auch frustrierend sein. Gerade in einer installativen Version von *game over* ist es wichtig, dass die Einstiegshürde so gering wie möglich ist und das Publikum das Spiel so weit wie möglich ohne Anleitung bedienen kann.

- *Verbs that act on many objects*

Diese wichtige Designstrategie ermöglicht, mithilfe weniger Verben ein komplexes Spielverhalten zu erzeugen. In der performativen Version von *game over* beispielsweise lösen klangliche Onsets völlige unterschiedliche Mechanismen aus, je nachdem wo sich der/die Spieler/in gerade aufhält bzw. welche Objekte sich in der Nähe befinden.

- *Goals that can be achieved more than one way.*

Auch damit wird erreicht, dass sich der Möglichkeitsraum erweitert, allerdings handelt es sich hierbei eher um Entscheidungen auf der Makroebene. Schell betont, dass die Herausforderung des Designers / der Designerin darin liegt, die Vor- und Nachteile der einzelnen Wege so abzuwiegen, dass diese in Summe gleichwertig erscheinen. In einer alten Version von *game over* hat der/die Spieler/in die Aufgabe (oder besser gesagt: die Möglichkeit), feindliche Raumschiffe im Weltall zu zerstören (wodurch, je nach Art des Treffers, der MIDI-Soundtrack des Spiels auf bestimmte Weise manipuliert wird). Dies kann auf zwei Arten erfolgen: Feuern von Schüssen durch das Spielen kurzer Staccato-Klänge oder absichtliches Rammen der anderen Raumschiffe (wodurch das eigene Raumschiff allerdings Schaden nimmt). Hier geht es weniger um die jeweiligen Nachteile (da es im Spiel ohnehin nicht ums Gewinnen geht), sondern vielmehr um die spielmechanischen (und darum

²⁵ Schell, *The Art of Game Design*, S. 141 f.

²⁶ Wahrscheinlich bezieht er sich dabei auf klassische Text-Adventure-Games von Sierra On-Line (z. B. *King's Quest*), wo der/die Spieler/in mit der Welt interagiert, indem er Textkommandos in die Tastatur tippt. Entscheidend für den Spielfortschritt ist, das passende Verb für die jeweilige Situation zu finden.

auch musikalischen!) Begleiterscheinungen.

- *Many subjects.*

Dies ist eine Technik, die man typischerweise aus Strategiespielen kennt (man denke an Schach und die unterschiedlichen Interaktionsmöglichkeiten der einzelnen Spielsteine). Bis jetzt kontrolliert der/die Spieler/in in *game over* innerhalb einer Welt immer denselben Avatar, allerdings wäre es ein interessantes Experiment, ihn/sie in die Rolle anderer Akteure schlüpfen zu lassen (z. B. bei Kollision).

- *Side effects that change constraints.*

Viele Aktionen haben indirekte Auswirkungen, welche wiederum die Ausführung zukünftiger Aktionen beeinflussen. Durch die Wahl absurder *side effects* lassen sich originelle Effekte erzielen: Bildschirmdrehung bei Wandkollision, Gravitationsänderung bei Bewegung, Größenänderung abhängig von der Lautstärke etc.

2.2.2 Feedback

In *Anatomy of a Choice*²⁷ formulieren Salen und Zimmerman zwei Fragen, welche für das Interface-Design von großer Bedeutung sind:

- *How is the possibility of choice conveyed to the player?*

Die bloße Möglichkeit der Interaktion muss medial vermittelt werden. Dies kann entweder durch explizite Anweisungen oder mithilfe visueller oder akustischer Signale erfolgen. Dabei kann ein sogenanntes *virtuelles Interface*²⁸ helfen, welches traditionellerweise aus GUI-Elementen wie Menüs, Statusleisten oder Statistiken besteht. Natürlich kann es manchmal auch sinnvoll sein, bestimmte Interaktionsmöglichkeiten zu verstecken, sodass diese erst im Verlauf des Spiels entdeckt oder freigeschaltet werden. Das virtuelle Interface kann auch selbst zum Teil der Spielwelt werden: In *game over* gibt es eine Welt, welche von klassischen Desktop-Icons bevölkert ist, die jedoch ihrer angestammten Funktion beraubt sind. Sehr interessant ist in diesem Kontext auch das Computerspiel *Pony Island*,²⁹ wo als Teil des Spiels zunächst einmal das Startmenü „repariert“ werden muss.

- *How is the result of the choice conveyed to the player?*

27 Salen und Zimmerman, *Rules of Play*, Kapitel 6, S. 8.

28 Schell, *The Art of Game Design*, S. 225.

29 <http://www.pony-island.com>

Es ist von großer Wichtigkeit, dass Spieler/innen unmittelbares Feedback für ihre Aktionen erhalten, auch wenn die eigentlichen Resultate indirekt oder verzögert eintreten. Dafür reicht bereits ein kurzes akustisches und/oder visuelles Signal. Dies ist besonders wichtig, wenn die Interaktion mithilfe untypischer Interfaces erfolgt. Falls, wie bei *game over*, der Klang des Instruments bestimmte Game-Events auslöst, sollte der Moment einer signifikanten Parameteränderung idealerweise auch visuell markiert werden, sodass die resultierenden Änderungen im Spielzustand leichter mit der Ursache assoziiert werden. Im Falle der *onset detection* ist weder für den/die Spieler/in, noch für das Publikum allein durch das Hören klar, wann der Schwellwert überschritten wird. Wenn hingegen sofort ein Grafikpartikel aufblitzt, drängt sich der Verdacht auf, dass jene klangliche Aktion womöglich etwas bewirken wird. Wenn tatsächlich kurz darauf ein Objekt eine Reaktion zeigt, entsteht der Eindruck einer Kausalkette. Ist die Art der Reaktion eher subtil oder nicht unmittelbar sichtbar/hörbar, wie z. B. eine graduelle Verhaltensänderung, hilft es, den Beginn der Reaktion, also den Verhaltensimpuls, visuell oder klanglich am Objekt zu markieren. Damit signalisiert man, dass nicht nur der Avatar, sondern auch seine Umgebung auf die Aktion reagiert. Diese Prinzipien ähneln denen der klassischen Konditionierung, welche sich durch Kontiguität (zeitliche und räumliche Nähe) sowie Kontingenz (zuverlässige Vorhersage) auszeichnen.³⁰ In *game over* tritt zudem das Phänomen auf, dass das akustisch-visuelle Signal selbst zum Ziel der Handlung wird, z. B. wenn der/die Spieler/in nur um des Klanges willen in eine Spawnbox hüpfte, ohne die unmittelbare Absicht, ein neues Objekt in die Welt zu setzen.

2.2.3 Interaktion

Das physische und virtuelle Interface vermittelt zwischen dem/der Spieler/in und dem für ihn/sie sicht- und hörbaren Ausschnitt der Welt, den ich *View* nenne möchte. Es gibt digitale Spiele, wie z. B. Tetris, wo die *View* die gesamte mögliche Welt umfasst. Oft ist die Welt jedoch wesentlich umfangreicher als der maximal darstellbare Ausschnitt und dem/der Spieler/in werden somit notwendigerweise Informationen vorenthalten. Es ist zu beachten, dass der hörbare Ausschnitt oft größer ist als der sichtbare. Dies verstärkt den Eindruck, dass die Welt über die Grenzen des Sichtbaren hinaus existiert; wie weit genau, lässt sich allerdings aus der Sicht des Spielers / der Spielerin nicht bestimmen, und jede Aussage darüber fällt streng genommen in den Bereich der Metaphysik. So könnte es zum Beispiel sein, dass aus Gründen der Recheneffizienz nur jene Teile upgedatet werden, die innerhalb eines bestimmten Radius des Spielers / der Spielerin liegen, während die restliche Welt temporär aufhört zu existieren – oder aber dort Dinge passieren, von

³⁰ Richard J. Gerrig und Philip G. Zimbardo, *Psychologie*, 18. Auflage, München 2008, S. 200 ff.

denen wir keine Vorstellung haben. Von außen betrachtet, hat man es mit einer Unschärferelation zu tun: man kann die Welt nicht beobachten, ohne sie damit gleichzeitig zu verändern. Außerdem ist jede View nur eine von unzähligen möglichen sinnlichen Erscheinungsformen jener Entitäten, welche im Grunde nur aus Code und Daten bestehen und vom Spieler / von der Spielerin in ihrer „eigentlichen“ Form nicht wahrgenommen werden können. Es sind sogar Spiele denkbar, welche ein und dieselbe Simulation jedem/jeder Spieler/in auf völlig unterschiedliche Art präsentieren. Die Praktik des Moddings (siehe 2.3.3) demonstriert bereits die Relativität der sinnlichen Erscheinungsform in Computerspielen. Dirk Paesmans und Joan Heemskerk gehen in ihrem *Castle-Wolfenstein*-Mod namens *SOD* sogar so weit, alle figurativen Elemente zu verbannen und die Grafik auf einfache geometrische Formen zu reduzieren.³¹

2.3 Implizite Regeln

Bernard Suits definiert Spiel knapp als „voluntary effort to overcome unnecessary obstacles“ und spezifiziert: „the rules prohibit more efficient in favour of less efficient means [...] such rules are accepted just because they make possible such activity.“³² Er demonstriert dies anhand des Golfspiels:

Suppose I make it my purpose to get a small round object into a hole in the ground as efficiently as possible. Placing it in the hole with my hand would be a natural means to adopt. But surely I would not take a stick with a piece of metal on one end of it, walk three or four hundred yards away from the hole, and then attempt to propel the ball into the hole with the stick. That would not be technically intelligent. But such an undertaking is an extremely popular game, and the foregoing way of describing it evidently shows how games differ from technical activities.³³

Man vergleiche dies mit *game over*: Ein/e Klarinetist/in verrenkt seine/ihre Hände, um eine Computerspielfigur mühsam durch unzählige Hindernisse zu manövrieren, während er/sie gleichzeitig versucht, spontan Musik zu machen. Natürlich könnte er/sie einfach frei improvisieren und jemand anders bedient das Computerspiel (sofern es nicht sogar aufgezeichnet ist). Die freiwillige Akzeptanz ineffizienter und umständlicher Regeln charakterisiert das, was Suits *lusory attitude* nennt. Ein Spiel ist gewissermaßen ein sozialer Vertrag unter Spielern, jene psychische Einstellung über die Länge des Spiels hinweg aufrechtzuerhalten. In gewisser Weise erfordert auch ein gewöhnliches Instrumentalkonzert ein ordentliches Maß an *lusory attitude*, denn natürlich wäre

31 Salen und Zimmerman, *Rules of Play*, Kapitel 32, S. 4.

32 Zitiert nach ebenda, Kapitel 7, S. 7.

33 Zitiert nach ebenda, Kapitel 9, S. 5.

der einfachste Weg, Musik darzubieten, eine Aufnahme abzuspielen. In beiden Fällen ist der persönliche Mehrwert jedoch groß genug, die Anstrengungen des Spiels auf sich zu nehmen.

Nicht alle Regeln sind explizit ausformuliert. In traditionellen Spielen gibt es bestimmte Konventionen, deren Einhaltung Teil von Etikette und Fair-Play sind. Dies gilt auch für sehr freie Systeme, mögen sie noch so sehr den Anschein vollkommener Offenheit erwecken. Auch bei grafisch notierter Musik „darf“ ich nicht alles spielen, die genaue Grenze bleibt jedoch undefiniert, der Kanon des Verbotenen unausgesprochen. Will man seine Prinzipien nicht verraten, muss man seinen Spielern und Spielerinnen vertrauen und sicherstellen, dass man ein ähnliche ästhetische Grundhaltung teilt. In *game over* vermeide ich bewusst, explizite Vorgaben zu machen. Ich verlasse mich dagegen auf die konstitutiven und operativen Regeln und den sich daraus ergebenden Möglichkeitsraum. Ich kann auf Möglichkeiten aufmerksam machen, darf aber nicht erwarten, dass der/die Spieler/in diese auch um jeden Preis ergreifen wird.

2.3.1 Exploits

Ich ermutige die Spieler/innen, kreativ mit den konstitutiven und operativen Regeln umzugehen. Die Game-Engine bzw. einzelne Welten weisen oft Fehler in der Programmierung auf. Solange es zu keinem Absturz kommt, sehe ich darin kein Problem, im Gegenteil: durch das gezielte Ausreizen von Schwächen im System (sogenannte Exploits) entstehen oft hochinteressante, unvorhergesehene Resultate. Um ein konkretes Beispiel zu geben: Wenn der/die Spieler/in von unten in eine Spawnbox hüpfte, wird über ihm ein neues Objekt erzeugt und durch die Kollision wird der/die Spieler/in ein Stück weit zurückgeschleudert. Ich habe jedoch vergessen, einen Schwellwert für die Kollisionsstärke zu setzen, ab welcher die Spawnbox aktiviert werden soll. Der/die Spieler/in hat womöglich herausgefunden, dass wenn er/sie sich im Flugmodus langsam der Box nähert und diese nur leicht berührt, der Rückstoß so schwach ist, dass er/sie mit einiger Beharrlichkeit die Box dauerhaft berühren kann. Ab diesem Moment wird bei jedem neuen Frame ein neues Objekt in die Welt gesetzt (also 30-60 Objekt pro Sekunde). Innerhalb kurzer Zeit ist der Ort überflutet von Akteuren, die alle ihre eigenen Klänge erzeugen. Was zunächst mit komplexen Patterns beginnt, wird schnell zu einer gewaltigen Klangwand, welche die CPU an ihre Grenzen treibt. Meine Reaktion darauf war nun nicht etwa, den Fehler zu beheben; stattdessen habe ich diesen Exploit als mögliche musikalische Entscheidung akzeptiert und eine elegante Exit-Strategie entwickelt (auf Knopfdruck können alle Objekte von mir wieder zerstört werden).

2.3.2 Cheats und Easter Eggs

Als Cheat (englisch für *Betrug*, *Schwindel*) wird die Möglichkeit bezeichnet, in einem Computerspiel selbst oder durch externe Programme das Spiel in einer nicht dem gewöhnlichen Spielverlauf entsprechenden Weise zu beeinflussen.³⁴

Dies ist eine sehr weite Definition von Cheat, welche den Exploit inkludiert (das gezielte Ausnutzen von Fehlern und Schwächen im Spiel). Mich interessieren aber vielmehr zwei spezielle Formen des Cheats: vom Schöpfer / von der Schöpferin bewusst eingebaute Schlupflöcher, welche entweder das Spiel erleichtern, indem Funktionen freigeschaltet werden oder bestimmte Regeln außer Kraft gesetzt werden, sowie sogenannte Easter-Eggs: vom Entwickler / von der Entwicklerin meist gut versteckte Features, welche rein der Unterhaltung dienen und keinen „praktischen“ Zweck erfüllen. In der alten an Weltraum-“Shoot' em ups“ angelehnten Version von *game over* hat der/die Spieler/in eine begrenzte Anzahl an Leben, um für mehr Spannung zu sorgen. Um zu vermeiden, dass das Spiel zu früh endet, habe ich die Möglichkeit eingebaut, über einen externen Befehl zusätzliche Leben zu vergeben. Der/die Spieler/in ist sich dessen bewusst, aber für das Publikum soll die Illusion soweit wie möglich aufrechterhalten werden (lustigerweise wird der Cheat meist gar nicht entdeckt). Die Grenze zwischen Easter-Eggs und bloß absurden Spielelementen ist, besonders bei einem an sich bereits absurden Spiel wie *game over*, schwer zu ziehen. Es ist jedoch explizit mein Ziel, den/die Spieler/in – und damit auch das Publikum – bei jeder Aufführung zu überraschen. In meiner Zusammenarbeit mit Szilard Benes hat sich die Praxis bewährt, einerseits auf bekannte Elemente zurückzugreifen bzw. Neuerungen vorher durchzusprechen oder zu proben, andererseits auch ausreichend Raum für Unbekanntes lassen. Letzteres erfordert ein hohes Maß an gegenseitigem Vertrauen; es hat aber einen ganz besonderen Reiz, wenn Dinge im Konzert zum ersten Mal geschehen und man spürt, dass der/die Spieler/in im wahrsten Sinne des Wortes improvisieren muss.

2.3.3 Modding

Modding bezeichnet die Tätigkeit, vorhandene Spiele zu modifizieren. So können beispielsweise Game-Assets (Texturen, 3D-Modelle, Karten etc.) verändert, ausgetauscht oder neu erzeugt werden oder Teile des Quellcodes umgeschrieben werden. Die so entstehenden Mods reichen von kleinen kosmetischen Änderungen bis hin zu vollkommen transformierten Spielen. Manche Mods werden sogar erfolgreicher als das Original (z. B. Counter Strike). Der sehr einflussreiche Programmierer

³⁴ [https://de.wikipedia.org/wiki/Cheat_\(Computerspiel\)](https://de.wikipedia.org/wiki/Cheat_(Computerspiel))

John Carmack, Schöpfer von *Doom*, hat sich schon früh dafür eingesetzt, den Quellcode von Computerspielen offenzulegen und die Modding-Kultur bewusst zu fördern, da Spiele so über Jahre hinweg aktuell bleiben. Das von seiner Firma *id Software* herausgebrachte Spiel *Quake* (1996) verwendet für die Spiellogik eine eigens entwickelte Skriptsprache namens QuakeC. Diese wurde sofort veröffentlicht, sodass technisch versierte Spieler/innen das Spiel in bisher unbekanntem Maße verändern konnten.

game over wurde mit dem Ziel erstellt, Spielwelten möglichst einfach und schnell erstellen und manipulieren zu können: Karten werden in einem grafischen Level-Editor gezeichnet, Grafiken und Klänge können problemlos ausgetauscht werden, auch die in der Skriptingsprache Lua geschriebene Spiellogik lässt sich leicht verändern. Eine Besonderheit von *game over* ist, dass das Spiel auch in Echtzeit angepasst und umgeschrieben werden kann. Dabei liegt der Fokus auf Sicherheit (das Spiel kann praktisch nicht abstürzen), sodass es sich auch für Live-Coding eignet. Theoretisch lässt sich *game over* auch vollkommen ohne Avatar „spielen“, in dem man mithilfe von Textbefehlen Objekte erzeugt, Events sendet und Eigenschaften bzw. Funktionen überschreibt.

3 Form

Dieser Abschnitt beschäftigt sich nun mit den Konsequenzen all jener expliziten und impliziten Regeln auf die formale Struktur des Spiels. Ich unterscheide grob gesprochen zwischen zwei Arten von Form: Die *konkrete* Form bezeichnet den zeitlichen Verlauf eines Werkes innerhalb einer bestimmten singulären Aufführungssituation (dazu zählt auch das Abspielen einer Aufnahme); die subjektive Erfahrung ist naturgemäß stark von der physischen und psychischen Disposition des Rezipienten / der Rezipientin abhängig, sie ist aber in jedem Fall einzigartig. Unter *konzeptuelle* Form hingegen verstehe ich den gesamten Möglichkeitsraum eines Werkes; der/die Künstler/in kann diesen nach Belieben enger oder weiter gestalten, jedoch nie zur Gänze kontrollieren. Diese grundsätzliche Unterscheidung kann bei der Klärung mancher Missverständnisse oder scheinbarer Widersprüche hilfreich sein.

3.1 Offene Form

Die Idee der offenen Form kommt aus der Literatur und Musik und ist u. a. von Umberto Eco's Schrift *Das offene Kunstwerk* beeinflusst. Während Eco in erster Linie die Offenheit eines Werkes in Bezug auf mögliche Interpretationen und Rezeptionshaltungen untersucht, begannen Künstler/innen damit, den linearen Verlauf ihrer Werke aufzubrechen und durch ein Netz möglicher Abläufe oder Kombinationsmöglichkeiten zu ersetzen. Konkrete Vorbilder sind u. a. das Konzept des Mobiles von Alexander Calder, auf welches sich Komponisten wie Earle Brown oder Roman Haubenstock-Ramati beriefen,³⁵ oder das unvollendete *Livre* von Stephan Mallarmé. Auch hier muss man eine wichtige Unterscheidung treffen, nämlich die zwischen Herstellung, Aufführung und Rezeption. Ich möchte nun Beispiele für offene Formen geben, damit die Relevanz dieser Unterscheidung sichtbar wird.

a) Ein/e Künstler/in verfolgt eine *konzeptuelle* offene Form, welche sich durch ein bestimmtes Regelwerk definiert, sodass das konkrete Werk nur eine von vielen möglichen Ausformungen ein und derselben Idee ist (oder womöglich gar keiner Materialisierung bzw. klanglichen Realisierung bedarf). Dies ist z. B. für das Schaffen von Josef Matthias Hauer charakteristisch (*Zwölftonspiel*), aber auch für viele Arten algorithmischer Komposition. Das einzelne Werk selbst ist fixiert und somit für Aufführende wie auch das Publikum keine offene Form im Sinne der konkreten Form; man muss den Kontext des Schaffensprozesses kennen, um die konzeptuelle offene Form würdigen

³⁵ Explizit in *Calder Piece* (1966) von Earle Brown oder in *Mobile for Shakespeare* (1958) von Haubenstock-Ramati.

zu können – und dies wahrscheinlich mehr auf einer intellektuellen als sinnlichen Ebene. In gewisser Hinsicht trifft dies auch auf *game over* zu, da die individuellen Spiele nur temporäre Manifestationen einer übergeordneten Idee darstellen. Dieser Aspekt lässt sich im Grunde nur überzeugend vermitteln, wenn das Publikum mehrere Versionen des Werkes sieht, am besten in längeren zeitlichen Abständen, damit die Unterschiede deutlicher zutage treten.

b) Ein/e Künstler/in schafft Werke, welche von dem/der Ausführenden anhand bestimmter Regeln gelesen oder assembliert werden müssen. Dies trifft z. B. auf die *Variations* von John Cage, das *Klavierstück XI* von Karlheinz Stockhausen oder viele grafische Partituren zu. Die Erfahrung einer konkreten offenen Form ist für den/die Ausführende/n unmittelbar erlebbar – und zwar umso stärker, je mehr die Entscheidungen in Echtzeit getroffen werden können/müssen. Für das Publikum gilt dies nur in eingeschränktem Maße, denn es hat auf die konkrete formale Entwicklung keinen (bewussten) Einfluss. Wenn man ein solches Werk mehrmals hört, kann man zumindest die Abweichungen erkennen und somit womöglich die konzeptuelle offene Form würdigen. Je nach dem, wie stark sich die spezifische Situation des/der Ausführenden auf das Publikum überträgt, ergibt sich aber auch die Möglichkeit einer sinnlichen Erfahrung der offenen Form: man spürt, wie das Kunstwerk im Moment entsteht und somit nicht wiederholbar ist. Zu jener Aura der Singularität gesellt sich das Erlebnis der körperlichen und geistigen Anstrengungen des Interpreten / der Interpretin im Kampf um das gelungene Fortführen der Form. Dies gilt besonders für viele Arten der improvisierten Musik, welche den Endpunkt offener musikalischer Form markieren. Eine konzertante Aufführung von *game over* lässt sich unter diesen Punkt einordnen, wobei hier durch die visuelle Repräsentation des Spiels der Entscheidungsspielraum des Interpreten noch verdeutlicht werden soll, sodass sich die Idee und Erfahrung der offenen Form so weit wie möglich auch auf das Publikum überträgt.

c) Das Werk ist ein variabler Text, dessen Struktur vom Rezipienten / von der Rezipientin selbst bestimmt werden muss. Neben manchen Formen der Literatur (*game books*) oder des Computerspiels (*adventure games*) trifft dies auch auf gewisse musikalische Situationen zu, z. B. wenn sich das klangliche Geschehen über einen größeren Raum erstreckt, der vom Publikum frei durchwandert wird. In diese Kategorie fällt die installative Version von *game over*. Dort bestimmt das Publikum auf mehreren Ebenen die formale Struktur der eigenen und teilweise auch kollektiven ästhetischen Erfahrung: Ist man im Besitz des Game-Controllers, lässt sich das individuelle Erlebnis der Spielwelt zum größten Teil frei bestimmen; dadurch, dass andere Menschen einem womöglich über die Schulter schauen oder die Klänge aus der Nähe hören, wird auch deren ästhetische Erfahrung beeinflusst. In der bisher größten Variante waren sechs Spielstationen (Sofa,

Röhrenfernseher, Einplatinen-Computer, Kopfhörer und Game-Controller) an den Rändern eines großen Saals platziert. Der Klang wurde gleichzeitig über Kopfhörer und die integrierten Fernhsehlautsprecher abgespielt sowie zeitlich verzögert und verfremdet über eine ambisonische Lautsprecherkuppel in den Saal projiziert, sodass auch ohne aktive Teilnahme am Spielgeschehen eine in gewissem Rahmen offene ästhetische Erfahrung möglich war.

3.2 Raum

Die konkrete Form von *game over* bestimmt sich jedoch nicht nur über den zeitlichen Verlauf, sondern auch wesentlich über die virtuelle Raumgestaltung, sowohl in visueller als auch akustischer Hinsicht. Eine der zentralen Ausgangsideen für die Entwicklung von *game over* war der Wunsch, unterschiedliche räumliche Repräsentationen, inklusive der dazugehörenden Spiel-Modi, innerhalb einer Welt zu versammeln. Die Voraussetzung dafür ist die bewusste Beschränkung der Spiel-Physik auf zwei Dimensionen, deren offensichtlicher Vorteil natürlich die im Vergleich zu einer 3D-Engine ungemein einfachere und schnellere Programmierung von Welten bzw. Erstellung von Game-Assets ist; weniger offensichtlich ist jedoch, dass auf aufgrund der räumlichen bzw. perspektivischen Einschränkung unterschiedliche Modi erst ermöglicht werden. In einer 3D-Engine hat man in der Regel eine Kamera, welche verschiedene Positionen und Perspektiven einnehmen kann. Dabei wird oft zwischen First-Person- und Third-Person-Einstellungen unterschieden. Diese Trennung ist jedoch nicht so fundamental, wie man zunächst glauben möchte, denn die Art der Projektion bleibt dieselbe; viele Leute behaupten (ob zu Recht oder nicht, sei dahingestellt), dass die First-Person-Einstellung immersiver sei; aber sie ändert nicht grundsätzlich, wie ich die Welt wahrnehme, denn ich habe genau dieselben Freiheitsgrade, mehr oder weniger dasselbe Blickfeld und in jedem Fall dieselbe Art der perspektivischen Projektion. In der Welt der 2D- und Pseudo-3D-Grafik hingegen gibt es zahlreiche Modi, die sich jeweils über die charakteristischen Einschränkungen der Freiheitsgrade, das sichtbare Blickfeldes sowie die Art der Projektion definieren. Die folgende Grafik stellt den Versuch einer Systematik dar, welche die verschiedenen Grafik- und Spiel-Modi miteinander in Beziehung setzt. Die Modi selbst sind persönliche Abstraktionen, denen sich oft mehrere, teils sehr unterschiedliche, Spielgenres zuordnen lassen.

Modus	Beispiele	Perspektive	Projektion	Rotation	Skalierung	z-Achse	Steuerung	Gravitation
Platformer	Super Mario, Mega Man, Sonic	seitlich	orthographisch	nein	nein	nein	lateral + springen	ja
Beat 'em up/ Shoot 'em up	Final Fight, Streets of Rage, Double Dragon	seitlich/von oben	orthographisch	nein	nein	ja	lateral + vorne/hinten (z-Achse) + springen	ja
Adventure/ RPG	Zelda, Diablo, Pokemon, Ultima	seitlich/von oben teils 45° gedreht für isometrische Projektion	axononmetrisch bzw. isometrisch	nein	nein	selten	direktional, [+ springen in z-Achse]	nein bzw. in Richtung z-Achse
Top Down	Pacman, GTA 1 & 2, Death Rally	von oben	orthographisch	eventuell	eventuell	eventuell	direktional oder rotieren + thrust, [+ springen in z-Achse]	nein bzw. in Richtung z-Achse
Mode 7	Mario Kart, F-Zero	hinten	perspektivisch	ja	ja	selten	rotieren + thrust, [+ springen in z-Achse]	nein bzw. in Richtung z-Achse
Space Shooter	Space Invaders, Asteroids	undefiniert	orthographisch	nein	nein	nein	lateral oder direktional oder rotieren + thrust	nein

Abbildung 5: Systematische Darstellung der verschiedenen Spielmodi

Diese Tabelle ist die Grundlage für die Gestaltung der Spielwelten in *game over*. Durch sorgfältige Auswahl der Spielmodi lassen sich entweder sehr logisch und flüssige Übergänge oder verwirrende Collagen erstellen. Durch die Simultanität sich widersprechender Perspektiven wird das Konzept einer allgemein gültigen Realitätsgrundlage in Frage gestellt – die Welt wird zur Ansichtssache.

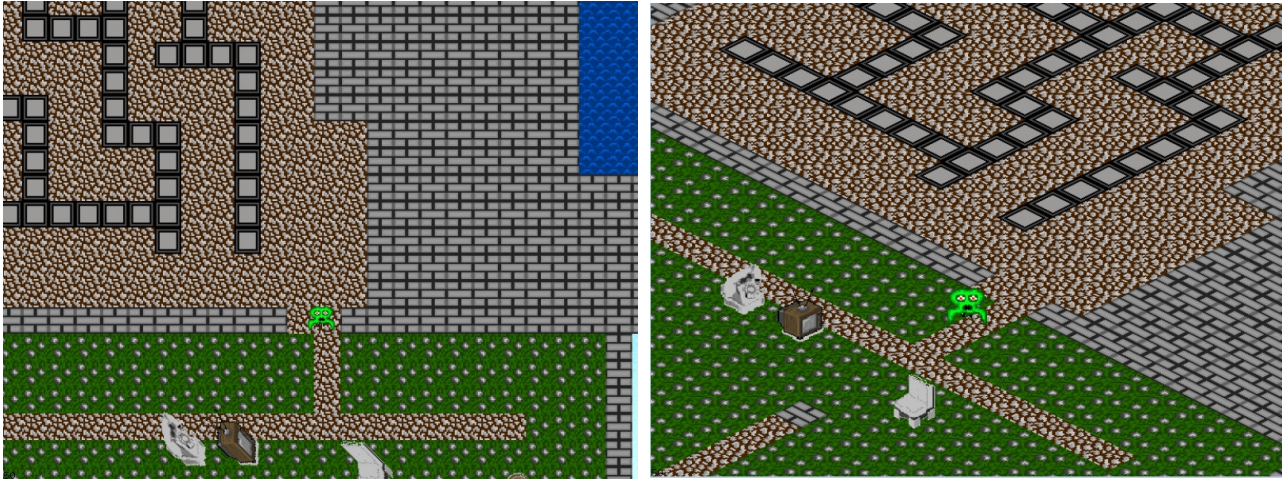


Abb. 6: Simultanität der Spielmodi (hier: Vogelperspektive vs. isometrische Ansicht).

Der nächste Schritt besteht darin, über die Navigation innerhalb einer Welt bzw. zwischen mehreren Welten nachzudenken. Diese hat nämlich nicht nur Einfluss auf die räumlich-visuelle Erfahrung, sondern auch auf die klangliche Struktur. Es gibt zwei grundsätzliche Techniken, Räume miteinander zu verbinden: einerseits können sie einfach nebeneinander montiert werden, also innerhalb einer großen Welt untergebracht werden, andererseits können die unterschiedlichen Räume je ihre eigene Welt ausfüllen. Im ersten Fall erfolgt die Navigation kontinuierlich durch eine Bewegung von A nach B, im zweiten Fall jedoch diskontinuierlich, möglicherweise durch das Betreten von Portalen oder automatisch nach gewisser Zeit. Allerdings kann auch die Navigation innerhalb einer großen Karte abrupt verlaufen, beispielsweise über Teleportation, und wenn die einzelnen Abschnitt ausreichend verschieden in ihrem Erscheinungsbild sind, lässt sich oft von außen gar nicht unterscheiden, ob man sich überhaupt noch innerhalb derselben Welt befindet. Werden separate Welten miteinander über Portale o. ä. verknüpft, bilden sich verschiedene Arten von Graphen. Schell unterscheidet fünf verschiedene Arten, den Raum zu organisieren: *linear*, *grid*, *web*, *points in space* und *divided space*.³⁶ Solche Topologien lassen sich auf alle Ebenen des Spiels anwenden: von der Montage weniger Kacheln bis hin zum Zusammenschluss ganzer Welten.

³⁶ Vgl. Schell, *The Art of Game Design*, S. 331-333

3.2.1 Die Welt als Partitur

Die Art und Weise, wie man sich durch die Welten bewegt, hat konkrete musikalische Auswirkungen. Lautstärke und Spatialisierung der einzelnen Klangerzeuger hängen von der aktuellen Avatar-Position ab: bewegt sich dieser, ändern sich entsprechend auch die klangliche Struktur. Klangerzeuger, denen eine physikalische Position und grafische Repräsentation innerhalb der Welt zugeordnet sind, nenne ich *diegetisch*, wobei man wiederum zwischen *on-screen-diegetisch* (die virtuelle Klangquelle ist sichtbar) und *off-screen-diegetisch* (die Klangquelle befindet sich außerhalb des sichtbaren Bereichs) unterscheiden kann. Diese können einerseits selbstständig Rhythmen, Sequenzen und Patterns produzieren, andererseits auch auf die Aktionen des Spielers / der Spielerin reagieren. Die Wesen können eine statische Position einnehmen, sodass der/die Spieler/in sich aktiv durch die Welt bewegen muss, um bestimmte Klänge hervorzuheben bzw. verschwinden zu lassen; oder sie können sich zufällig oder entlang festgelegter Pfade bewegen, sodass sich das Klangbild kontinuierlich ändert, auch wenn die View selbst statisch bleibt. Das Gestalten solcher Pfade ist an sich schon eine kompositorische Tätigkeit: Wesen können sich in einer Schleife bewegen, verschiedene Abzweigungen nehmen oder bei bestimmten Knoten klangliche oder rhythmische Eigenschaften ändern. Durch die Wahl der Position der Knotenpunkte (insbesondere deren Entfernung) und deren Verbindung lassen sich gezielt bestimmte Abläufe herstellen. Bewegen sich mehrere Wesen auf einem solchen Graphen, entstehen sehr rasch komplexe zeitliche Strukturen.

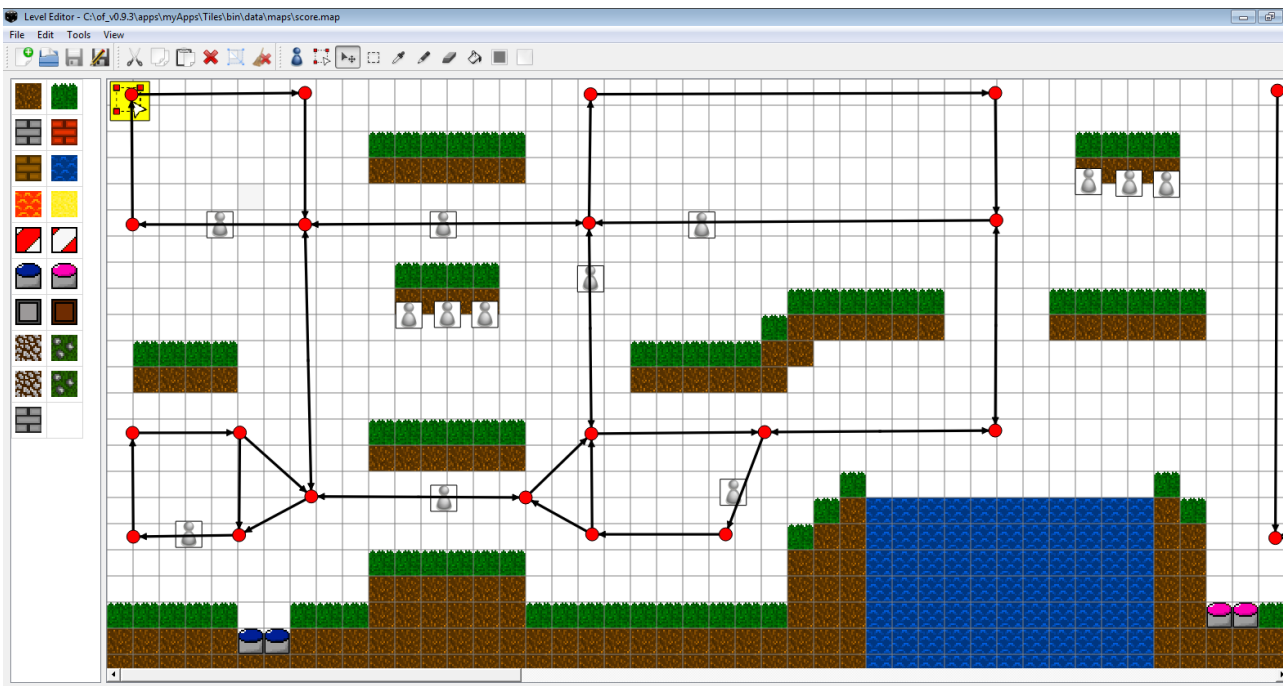


Abb. 7: Navigationsgraph im Leveleditor.

Dazu kommen sogenannte Regionen, die zwar eine physikalische Ausdehnung, aber keine grafische Repräsentation besitzen. Bewegt man sich innerhalb einer solchen Region, werden etwa bestimmte Prozesse ausgelöst oder globale klangliche Parameter verändert. Zum Beispiel gibt es Regionen, in denen die relative Position des Avatars die Position des Lesekopfes in einem Soundbuffer ändert, welcher etwa mithilfe von Granularsynthese oder eines Phasenvocoders abgespielt wird, wodurch die Bewegung des Spielers / der Spielerin direkt in Klang übersetzt wird. Durch das Einsammeln bestimmter Pickups kann der/die Spieler/in den Klang in eine von ihm gewünschte Richtung lenken. Dazu kommt noch der sporadische Einsatz von sogenannter *extra-diegetischer* Musik, welche ihren Ursprung nicht in der Spielwelt hat, also an keine bestimmte Entität gebunden ist, sondern unabhängig von der Avatar-Position erklingt und damit eher den Charakter eines Soundtracks hat. Allerdings können Aktionen des Spielers / der Spielerin durchaus Auswirkungen auf die extra-diegetische Musik ausüben, wodurch jene ontologische Trennung wieder in Frage gestellt wird.

Nimmt man die Idee des Computerspiels als komponierte Partitur ernst, sind auch Situationen denkbar, wo es gar nicht mehr eines menschlichen Spielers bedarf. So könnte sich die View auch automatisch entlang eines vorgegeben Pfades bewegen und damit die Komposition zur Aufführung gebracht werden. Andererseits kann das Konzept der Partitur, besonders in der konzertanten Version von *game over*, auch für den/die Spieler/in relevant sein, indem es als Handlungsanleitung oder Wegweiser für die Performance dient.

3.3 Teleologie

Viele Versuche einer konzisen Definition des Begriffs Spiel verweisen in der ein oder anderen Form auf ein bestimmtes Ziel oder Resultat, welches vom Spieler / von der Spielerin verfolgt wird. Hier sind zwei charakteristische Beispiele aus der Literatur:

- Q1. Games are entered willfully.
- Q2. Games have goals.
- Q3. Games have conflict.
- Q4. Games have rules.
- Q5. Games can be won and lost.
- Q6. Games are interactive.
- Q7. Games have challenge.
- Q8. Games can create their own internal value.

Q9. Games engage players.

Q10. Games are closed, formal systems.³⁷

A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.³⁸

Fairerweise muss ich darauf hinweisen, dass die Autoren/Autorinnen hier nicht beabsichtigen, eine allgemein gültige, absolute Definition zu liefern, sondern anerkennen, dass es auch Spiele gibt, die nicht gewonnen oder verloren werden können bzw. welche nicht in einem quantifizierbaren Resultat enden. Dies gilt z. B. für viele Simulationsspiele, welche zwar meist eine Idee von Fortschritt vermitteln, jedoch kein klares Ende haben. Andererseits muss ein von den Entwicklern/Entwicklerinnen beabsichtigtes Spielziel nicht unbedingt mit der persönlichen Absicht des Spielers / der Spielerin übereinstimmen. Dies trifft besonders auf Open-World-Spiele zu, z. B. wenn jemand in *Grand Theft Auto* lieber Amok läuft als die nächste Mission zu erfüllen oder wenn die Teilnehmer in *DayZ* (ein beliebter *ARMA*-Mod) die Zombie-Gegner ignorieren und sich lieber gegenseitig jagen. Erweitert man also den Begriff des Spielziels auf jene vom Spieler / von der Spielerin selbst gesteckten Ziele oder Strategien, erhalten folgende Beobachtungen eine tiefere Bedeutung:

A game's goal is often the largest single element that drives the pleasure of a player. The goal is the ostensible reason for playing, but the goal is never easily attained; rather, it is the obscure object of desire, the carrot held just out of reach, pulling players forward through the varied pleasures of game play. The goal helps move players through the space of possibility, a space stretched between the starting state of the game and its outcome like a billowing cloth staked to the ground. The goal acts to guide the players along the axis defined by the beginning and the end, letting them know if they are advancing or falling behind.³⁹

Was von Salen und Zimmerman als Charakterisierung übergeordneter, vom Entwickler / von der Entwicklerin definierter Spielziele gedacht war, lässt sich somit auch als Anleitung zur Selbstdisziplin lesen. Ich verzichte in meiner Spielanordnung grundsätzlich auf Ersteres und vertraue auf Letzteres. Das Ziel ist, sich seine eigenen Ziele zu setzen. Hier ein Zitat aus Mihaly Csikszentmihalyis Definition seines Flow-Konzepts: „A final characteristic is its 'autotelic' nature ... It needs no goals or rewards external to itself.“⁴⁰

37 Schell, *The Art of Game Design*, S. 34.

38 Salen und Zimmerman, *Rules of Play*, Kapitel 7, S. 11.

39 Ebenda, Kapitel 24, S. 14.

40 Zitiert nach Brian Sutton-Smith, *The Ambiguity of Play*, Harvard 1997, S. 185.

3.3.1 Belohnung und Bestrafung

It's surprising how many developers forget that it's the victories and the treasures – not the obstacles – that make people interested in playing in the first place. If you stop giving out the carrots that will keep players excited, or even worse, if you start punishing them for their curiosity, you're only going to drive away the very people who want to enjoy your game.⁴¹

Flexible und wohl durchdachte Anreizsysteme ermöglichen ein sinnvolles und befriedigendes Spielerlebnis, ohne dass es eines konkreten Spielziels bedarf. Salen und Zimmerman warnen zurecht vor einer Überzahl an Hindernissen und Bestrafungen, allerdings sind diese notwendig, um die Balance innerhalb des Spiels zu bewahren. An 2.2 anknüpfend, kann man sich den Möglichkeitsraum eines Spiels als gewichteten Graphen vorstellen: diejenigen Wege, welche zu einer außerordentlichen Belohnung führen, müssen mit einem entsprechenden Kostenfaktor (in Gestalt von Hindernissen und Strafen) belegt werden, um die Wahl riskanter und deshalb unwahrscheinlicher zu machen. Erst dadurch wird es für den/die Spieler/in sinnvoll, auch andere Wege einzuschlagen, welche zu geringeren Belohnungen führen. Durch die bewusste Platzierung solcher Belohnungen und Hindernisse innerhalb der Welt gestalte ich die möglichen räumlichen, zeitlichen und somit auch musikalischen formalen Abläufe. Salen und Zimmermann unterschieden Belohnungen anhand von vier Kategorien⁴²:

- *Rewards of Glory*: abstrakte Belohnungen, z. B. in Form von High-Scores, welche keinerlei Einfluss auf das Spielverhalten haben. Spiele-Designer/innen warnen zwar vor einem übermäßigen Einsatz, allerdings können sie ironisch, sarkastisch oder parodistisch eingesetzt sehr effektiv sein.
- *Rewards of Sustenance*: die Belohnung besteht darin, dass man etwas behält, im einfachsten Fall, dass man schlichtweg am Leben bleibt. Sie zeichnet sich also absurderweise nicht dadurch aus, dass man etwas bekommt, sondern dass man etwas nicht verliert. Man könnte in diese Kategorie eventuell auch jene Belohnung einordnen, welche darin besteht, dass die Spuren, welche man in der Welt hinterlassen hat, über das eigene Spiel hinweg bestehen bleiben. Diese Motivation ist sehr real und z. B. in der installativen Version von *game over* öfter zu beobachten.
- *Rewards of Access*: man erhält Zugang zu einem unbekannten oder besonderen Teil der Spielwelt, welcher den Möglichkeitsraum erweitert oder auch bloß für Überraschung sorgt

41 Salen und Zimmerman, *Rules of Play*, Kapitel 24, S. 18.

42 Ebenda.

(wie bei den schon besprochenen Easter-Eggs).

- *Rewards of Facility*: man wird mit dem Erhalt zusätzlicher Fähigkeiten belohnt, durch welche die bereits bekannten Bereiche der Welt in einem anderen Licht erscheinen.

Sämtliche Belohnungen, Hindernisse und Bestrafungen können teils oder zur Gänze musikalischer Natur sein. Allerdings wird gerade hier deutlich, dass die Einteilung in „erwünschte“ und „unerwünschte“ Ereignisse letztlich sehr subjektiv ist. In *game over* gibt es mehrere Pickups, die wie Medikamente aussehen und beim Einsammeln mehr oder weniger starke grafische und akustische Verfremdungen bewirken. Ich konnte selbst beobachten, wie manche Spieler/innen es sich zum Ziel gemacht haben, möglichst viele dieser Objekte zu sammeln (zum Teil bis das Bild nur mehr aus groben Pixeln und der Klang nur mehr aus verzerrtem Rauschen besteht), während andere diese ob ihrer krassen und für sie persönlich unangenehmen Wirkung bewusst vermieden haben. Das ein und dasselbe Objekt kann also sowohl Belohnung als auch Hindernis sein! Der entscheidende Kontext ist einerseits die generelle Disposition des Spielers / der Spielerin, andererseits der jeweils konkrete Spielzustand im Sinne von Vergangenheit, Gegenwart und Zukunft: wurde ein gewisser Anreiz bereits öfter genutzt, will man ihn womöglich in Zukunft zugunsten größerer Abwechslung vermeiden, vor allem wenn er anderen möglichen Anreizen entgegenwirkt.

3.5 Konstruktion/Destruktion

Die API der Game-Engine bietet vielfältige Möglichkeiten, während des Spiels Entitäten zu erzeugen, zu manipulieren und wieder zu zerstören. Von einem elementaren Standpunkt betrachtet mag das trivial erscheinen (die Welt würde ja ansonsten leer sein) und viele Strukturen, wie z. B. ein musikalisches Pattern aus verschiedenen Dauern und Tonhöhen, entstehen erst aus der Konstruktion bzw. Destruktion ihrer Einzelelemente. Interessant wird es jedoch, wenn man dieses Prinzip auf einer höheren Ebene anwendet, also in größeren räumlichen und/oder zeitlichen Dimensionen. Die Idee der ursprünglichen Version von *game over* war, ein anfangs intaktes MIDI-File mittels diverser Spielhandlung über einen Zeitraum von 10-15 Minuten langsam zu verändern bzw. zu zerstören. Die Destruktion in der Kunst ist zugleich auch Konstruktion, denn alles, was zerstört wird, öffnet den Blick auf das Andere. Nachdem ich das meiner Meinung nach zu geradlinige Spielprinzip von *game over* verlassen habe, habe ich versucht, neue Wege der kreativen Zerstörung zu finden. Eine Möglichkeit ist paradoxerweise die exzessive Konstruktion, z. B. durch

das Erzeugen tausender Wesen oder das Zumauern der Umgebung mit verschiedenen Kacheln. Eine andere Strategie ist das Verschieben bzw. Austauschen von bestimmten Bereichen, sodass der ursprüngliche Zusammenhang zerstört wird, aber gleichzeitig neue Verknüpfungen entstehen. Dinge, die vor kurzem noch weit voneinander getrennt waren, sind nun plötzlich benachbart. Schließlich kann die Destruktion auch rein auf der Ebene der View stattfinden: durch den Einsatz extreme Blickwinkel, das Verändern von Grafik-Parametern oder die Anwendung globaler Audio-Effekte lässt sich die ästhetische Erfahrung vollkommen transformieren, ohne die physikalisch-logische Welt zu beeinflussen.



Abb. 8: Destruktion der Spielwelt.

Schlusswort

Das Computerspiel wird in weiten Teilen der Gesellschaft noch immer nicht als eigenständiges Medium wahrgenommen, geschweige denn als legitimes künstlerisches Ausdrucksmittel. Schuld daran ist nicht zuletzt die Spiele-Industrie selbst, die extrem gewinnorientiert arbeitet und in vielen Fällen keinerlei künstlerische Absichten erkennen lässt. Allerdings gab es immer schon unabhängige Entwickler/innen bzw. kleine Studios mit starken künstlerischen Visionen sowie Künstler/innen, die das Potential des Mediums erkannt haben und auf kreative Weise für ihre Zwecke eingesetzt haben. Man muss sich vor Augen halten, dass auch der Film einige Zeit gebraucht hat, um sich von einer reinen Jahrmarkt-Attraktion zu einer ernstzunehmenden Kunstform zu wandeln.

Es ist notwendig, dass sich Künstler/innen mit den spezifischen Eigenschaften ihres Mediums auseinandersetzen; dazu zählt im Falle des Computerspiels vor allem die interaktive Komponente, welche es geradezu erfordert, traditionelle Denkweisen über Form und Autorschaft über Bord zu werfen. Ich hoffe, dass diese Arbeit dazu dienen kann, das Computerspiel als Medium in seinem ästhetischen und künstlerischen Potential ernst zu nehmen und Impulse für mögliche zukünftige Arbeiten zu geben.

Literaturverzeichnis

Brandstätter, Ursula: *Grundfragen der Ästhetik*, Köln 2008.

Buckland, Matt: *Programming Game AI by Example*, Texas 2005.

Ebert, Roger: *Why did the chicken cross the genders?* in: Chicago Sun-Times, Ausgabe vom 27. 11. 2005.

Ebert, Roger: *Video games can never be art* in: Chicago Sun-Times, Ausgabe vom 16. 4. 2010.

Oliva, Constantino: *On the onotological status of musical actions in digital games*, Krakau 2017.

Pross, Harry: *Publizistik: Thesen zu einem Grundcolloquium*, Neuwied 1970.

Reck, Hans Ulrich: *Kunst als Medientheorie*, München 2003.

Schell, Jesse: *The Art of Game Design. A Book of Lenses*, New York 2008.

Sutton-Smith, Brian: *The Ambiguity of Play*, Harvard 1997.

Zimbardo, Philip G.; Gerrig, Richard J.: *Psychologie*, 18. Auflage, München 2008.

Zimmerman, Eric; Salen, Katie: *Rules of Play*, Cambridge 2004.

Internet:

<https://www.openframeworks.cc> (zuletzt abgerufen am 30. 9. 2018)

<https://puredata.info> (zuletzt abgerufen am 30. 9. 2018)

<https://github.com/ThePhd/sol2> (zuletzt abgerufen am 30. 9. 2018)

https://en.wikipedia.org/wiki/Ray_casting (zuletzt abgerufen am 30. 9. 2018)

[https://de.wikipedia.org/wiki/Sprite_\(Computergrafik\)](https://de.wikipedia.org/wiki/Sprite_(Computergrafik)) (zuletzt abgerufen am 30. 9. 2018)

[https://de.wikipedia.org/wiki/Cheat_\(Computerspiel\)](https://de.wikipedia.org/wiki/Cheat_(Computerspiel)) (zuletzt abgerufen am 30. 9. 2018)

<https://www.pony-island.com> (zuletzt abgerufen am 30. 9. 2018)

Abbildungen:

1 Installative Fassung von *game over* (Foto © reithofermedia), S. 7.

2 Szilard Benes in der konzertanten Fassung von *game over* (Foto © reithofermedia), S. 8.

3 Leveleditor für *game over*, S. 9.

4 Definition einer Game-Objekt-Klasse, S. 13.

5 Systematische Darstellung der verschiedenen Spielmodi, S. 29.

6 Simultaneität der Spielmodi, S. 30.

7 Navigationsgraph im Leveleditor, S. 31.

8 Destruktion der Spielwelt (Foto © reithofermedia), S. 36.