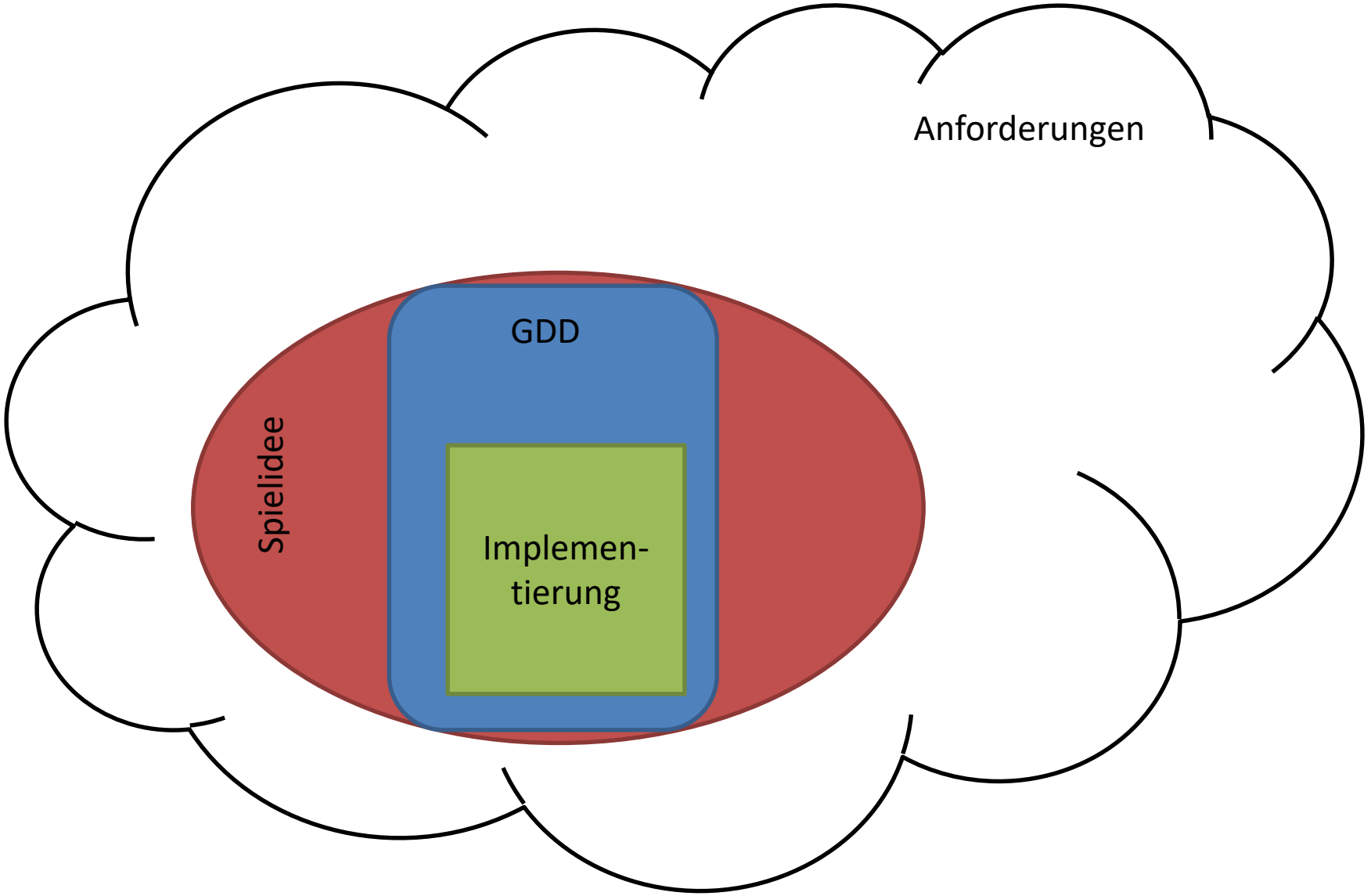
Game Design Document (GDD)

Game Design Document

- Ein GDD soll einen genauen Überblick das Spiel geben.
- Angelehnt an Lastenheft in der Softwareentwicklung.
- Es beinhaltet
  - genaue, widerspruchsfreie Beschreibung des Spielablaufs.
  - Überblick über alle im Spiel enthaltenen Features.
  - Beschreibung der Anwendungsfälle (Use Cases), die aus Anforderungen an das Spiel gewonnen werden.

Game Design Document

- Das GDD kann als Mischung aus Werbeprospekt, Benutzerhandbuch und Lastenheft gesehen werden:
  - Sie wollen dem Kunden Ihr Produkt schmackhaft machen.
  - Sie wollen dem Leser erklären, wie Ihr Spiel funktioniert.
  - Features, die im GDD beschrieben werden, sind bindend, d.h. sie müssen im fertigen Spiel in beschriebener Form vorhanden sein.
- Schreiben Sie das GDD gewissenhaft.
- Die hier vorgestellte Reihenfolge der Abschnitte muss nicht die optimale Reihenfolge sein!
- Hier: Grundlegendes und Beispiele.


Anforderungen
GDD
Spielidee
Implemen-
tierung

GDD

- Deckblatt
- Spielkonzept
  - Zusammenfassung des Spiels
  - Alleinstellungsmerkmal
- Benutzeroberfläche
  - Spieler-Interface
  - Menü-Struktur
- Technische Merkmale
  - Verwendete Technologien
  - Mindestvoraussetzungen
- Spiellogik
  - Optionen & Aktionen
  - Spielobjekte
  - Spielstruktur
  - Statistiken
  - Achievements
- Screenplay
  - Konzeptzeichnungen & Storyboards

GDD
Deckblatt

- Name des Spiels
- Namen der Gruppenmitglieder
- Name des Tutors
- Gruppennummer
- Datum der Erstellung

GDD

- Deckblatt
- Spielkonzept
  - Zusammenfassung des Spiels
  - Alleinstellungsmerkmal
- Benutzeroberfläche
  - Spieler-Interface
  - Menü-Struktur
- Technische Merkmale
  - Verwendete Technologien
  - Mindestvoraussetzungen
- Spiellogik
  - Optionen & Aktionen
  - Spielobjekte
  - Spielstruktur
  - Statistiken
  - Achievements
- Screenplay
  - Konzeptzeichnungen & Storyboards

GDD
Spielkonzept

- Zusammenfassung des Spiels
  - Um was geht es?
  - Was sind die Grundideen im Spiel?
  - Kann auch Storyelemente enthalten.
  - Darf als einziger Abschnitt im GDD auch reißerisch/dramatisch beschrieben sein.
  - Vgl. „Klappentext“ auf Rückseite von Spieleverpackungen

GDD an einem Beispiel

- Name: „Superjumper“
- Platformer (oder „Jump‘n‘Run“)
- Ähnlich zu Super Mario (gleiche Kamera, Steuerung, Verhalten)
- Zusätzlich „Rätsel“ in den Levels
- Rätsel lösbar durch Reihenfolge des Ablaufens bestimmter Wege und durch Einsatz von Fähigkeiten
- (Zauber-)Fähigkeiten, die kombinierbar sind und in Kombination neue Fähigkeiten sind
- Spielziel / Gewinnen: Alle Level meistern (man gelangt pro Level zu einer Markierung)
- Verlieren: In Stacheln/Abgründe/Gegner/usw. fallen/laufen und alle Leben verlieren.

Spielkonzept – Beispiel

Zusammenfassung des Spiels

Superjumper ist ein 2D Platformer. Rette die schöne Prinzessin Qwerty aus den Fängen des bösen Trolls Asdf. Du musst allen Widrigkeiten zum Trotz verschiedene Prüfungen und Rätsel in unterschiedlichen Levels bezwingen und deine Fähigkeiten weise einsetzen, um deine teure Qwerty schließlich wieder in deinen Armen halten zu können. Dabei geht es nicht nur um Geschwindigkeit, sondern vor allem um Geschick und Einfallsreichtum.

Spielkonzept – Beispiel

Zusammenfassung des Spiels

Superjumper ist ein 2D Platformer. Rette die schöne Prinzessin Qwerty aus den Fängen des bösen Trolls Asdf. Du musst allen Widrigkeiten zum Trotz verschiedene Prüfungen und Rätsel in unterschiedlichen Levels bezwingen und deine Fähigkeiten weise einsetzen, um deine teure Qwerty schließlich wieder in deinen Armen halten zu können. Dabei geht es nicht nur um Geschwindigkeit, sondern vor allem um Geschick und Einfallsreichtum.

GDD
Spielkonzept

- Alleinstellungsmerkmal
  - Was hebt das Spiel von der Masse ab?
  - Was ist einzigartig an und in diesem Spiel?
  - Wieso begeistert das Spiel den Spieler?

- Ein Alleinstellungsmerkmal kann sowohl
  - ein Feature, als auch
  - ein ganzes Spielkonzept sein.

Spielkonzept – Beispiel

Alleinstellungsmerkmal

Superjumper enthält viele Elemente eines klassischen Platformers. Was das Spiel jedoch außergewöhnlich macht ist die Tatsache, dass die Hauptfigur, die der Spieler steuert, unterschiedliche Fähigkeiten hat, die auf unterschiedliche Arten miteinander kombiniert werden können. Es kommt darauf an, welche Fähigkeiten miteinander kombiniert werden, um zum Erfolg zu gelangen. Diese Kombinationsmöglichkeiten sorgen dafür, dass Rätsel in unterschiedlichen Levels auf verschiedene Arten lösbar sind und so die Spannung und die Neuheit des Spiels auch bei mehrmaligem Durchspielen erhalten bleibt.

GDD

- Deckblatt
- Spielkonzept
  - Zusammenfassung des Spiels
  - Alleinstellungsmerkmal
- Benutzeroberfläche
  - Spieler-Interface
  - Menü-Struktur
- Technische Merkmale
  - Verwendete Technologien
  - Mindestvoraussetzungen
- Spiellogik
  - Optionen & Aktionen
  - Spielobjekte
  - Spielstruktur
  - Statistiken
  - Achievements
- Screenplay
  - Konzeptzeichnungen & Storyboards

GDD
Benutzeroberfläche

- Beschreibung der Steuerelemente, mit denen der Spieler mit dem Spiel interagiert.
- Beinhaltet
  - Spieler-Interface
    - Beschreibung dessen, was der Spieler sieht.
    - Art der Darstellung, Kameraansichten, sichtbare Elemente (HUD, Minimap, Menüleisten, usw.)
    - Bild (Konzeptzeichnung, Screenshot, Mockup) dessen, wie das Spiel aussehen soll.
    - Beschreibung der Steuerung.
  - Menü-Struktur
    - Struktur des Hauptmenüs und Pause-Menüs mit allen Untermenüs.
- Beinhaltet nicht
  - Beschreibung der Spiellogik.

Benutzeroberfläche – Beispiel

Benutzeroberfläche – Beispiel

Spieler-Interface

Abbildung 1 zeigt das Aussehen von Superjumper. Die Spielwelt orientiert sich an Super Mario Bros. In der oberen linken Ecke in Abbildung 1 befindet sich die **Gesamtanzahl der Punkte**, die in dem aktuellen Level erreicht wurden. Oben in der Mitte ist die **Anzahl der Leben** zu sehen, die der Spieler noch zur Verfügung hat. In der oberen rechten Ecke ist die bereits **verstrichene Zeit** im aktuellen Level zu sehen. Zentriert am unteren Bildschirmrand sieht man die gerade ausgewählten **Fähigkeitenkombination**.

Die zweidimensional gehaltene Spielwelt von Superjumper besteht aus einer Anzahl von **Plattformen**, auf die die Spielfigur springen kann. Außerdem können sich spezielle, einsammelbare **Items** und **Gegner** in der Spielwelt befinden. …

Benutzeroberfläche – Beispiel

Spieler-Interface

Abbildung 1 zeigt das Aussehen von Superjumper. Die Spielwelt orientiert sich an Super Mario Bros. In der oberen linken Ecke in Abbildung 1 befindet sich die **Gesamtanzahl der Punkte**, die in dem aktuellen Level erreicht wurden. Oben in der Mitte ist die **Anzahl der Leben** zu sehen, die der Spieler noch zur Verfügung hat. In der oberen rechten Ecke ist die bereits **verstrichene Zeit** im aktuellen Level zu sehen. Zentriert am unteren Bildschirmrand sieht man die gerade ausgewählten **Fähigkeitenkombination**.

Die zweidimensional gehaltene Spielwelt von Superjumper besteht aus einer Anzahl von **Plattformen**, auf die die Spielfigur springen kann. Außerdem können sich spezielle, einsammelbare **Items** und **Gegner** in der Spielwelt befinden. …

Benutzeroberfläche – Beispiel

Kamera

Die Kamera von Superjumper verhält sich wie in Super Mario Bros. Sie ist im Allgemeinen zentriert auf die Hauptfigur. Bewegt sich die Hauptfigur nach rechts, wird auch die Kamera mitbewegt. Bewegt sich die Spielfigur nach links oder befindet sie sich einer der Ränder der Welt zu nahe an den Grenzen des von der Kamera betrachteten Spielausschnitts, bleibt die Kamera auf ihre letzte Position gerichtet. In diesem Fall kann die Spielfigur an den Rand des sichtbaren Bereiches laufen, jedoch niemals darüber hinaus.

Benutzeroberfläche – Beispiel

Kamera

Die Kamera von Superjumper verhält sich wie in Super Mario Bros. Sie ist im Allgemeinen zentriert auf die Hauptfigur. Bewegt sich die Hauptfigur nach rechts, wird auch die Kamera mitbewegt. Bewegt sich die Spielfigur nach links oder befindet sie sich einer der Ränder der Welt zu nahe an den Grenzen des von der Kamera betrachteten Spielausschnitts, bleibt die Kamera auf ihre letzte Position gerichtet. In diesem Fall kann die Spielfigur an den Rand des sichtbaren Bereiches laufen, jedoch niemals darüber hinaus.

Benutzeroberfläche – Beispiel

Kamera

Die Kamera von Superjumper verhält sich wie in Super Mario Bros. Sie ist im Allgemeinen zentriert auf die Hauptfigur. Bewegt sich die Hauptfigur nach rechts, wird auch die Kamera mitbewegt. Bewegt sich die Spielfigur nach links oder befindet sie sich einer der Ränder der Welt zu nahe an den Grenzen des von der Kamera betrachteten Spielausschnitts, bleibt die Kamera auf ihre letzte Position gerichtet. In diesem Fall kann die Spielfigur an den Rand des sichtbaren Bereiches laufen, jedoch niemals darüber hinaus.

Benutzeroberfläche – Beispiel

Steuerung

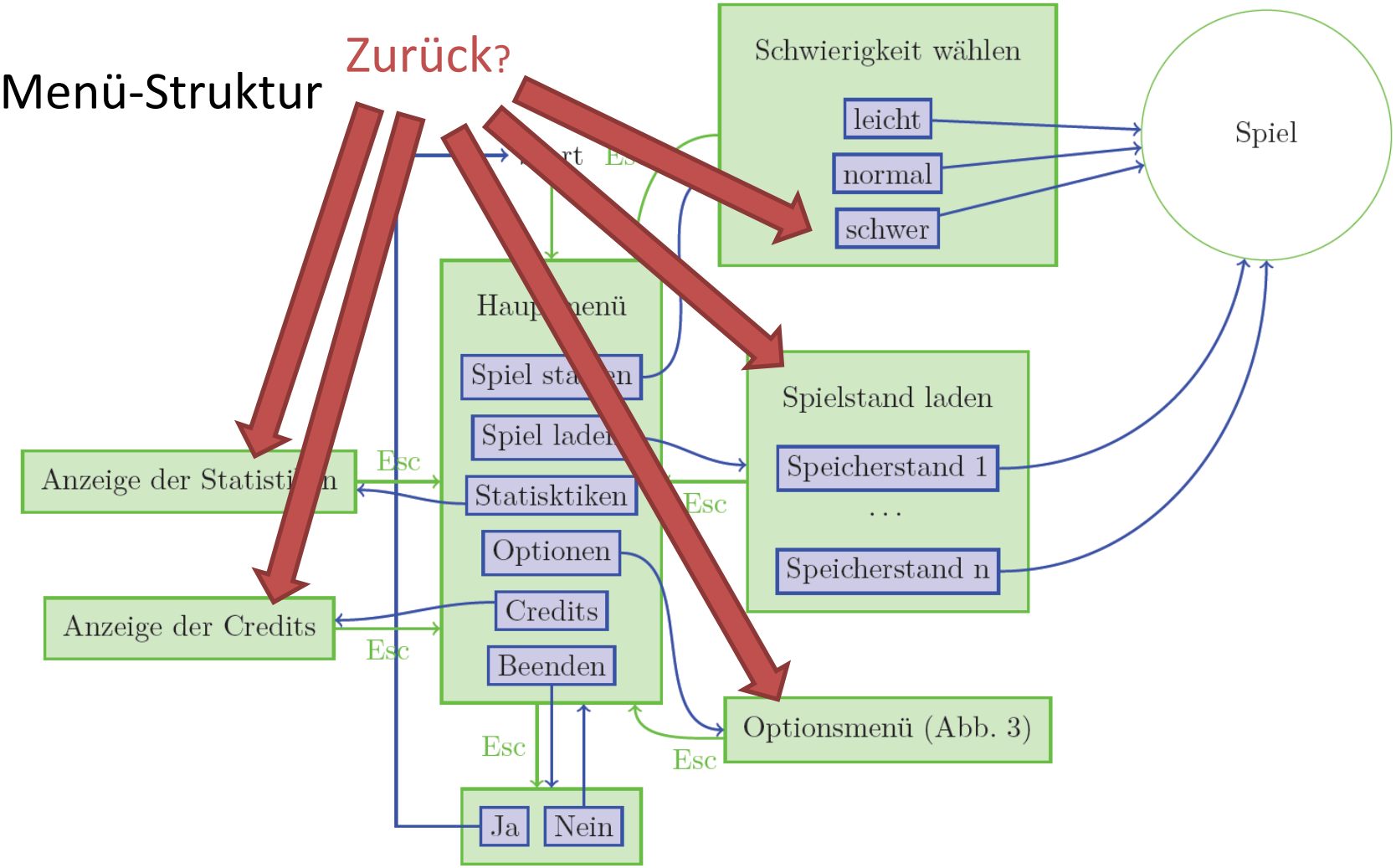
Die Steuerung von Superjumper erfolgt ausschließlich über die Tastatur. Im Menü kann außerdem die Maus zur Navigation verwendet werden. Tabelle 1 zeigt die Tastatursteuerung im Spiel.

Taste	Beschreibung
→	Bewegt die Hauptfigur nach rechts
←	Bewegt die Hauptfigur nach links
↑	Die Hauptfigur springt / schwimmt nach oben
↓	Die Hauptfigur duckt sich / schwimmt nach unten
Ctrl	Die Primärfähigkeit wird benutzt
Alt	Die Sekundärfähigkeit wird benutzt
1 ... 9	Die Primär- und Sekundärfähigkeit wird ausgewählt
Pause	Das Spiel wird pausiert
Esc	Das Pausemenü wird aufgerufen

Tabelle 1: Die Steuerung von Superjumper über die Tastatur.

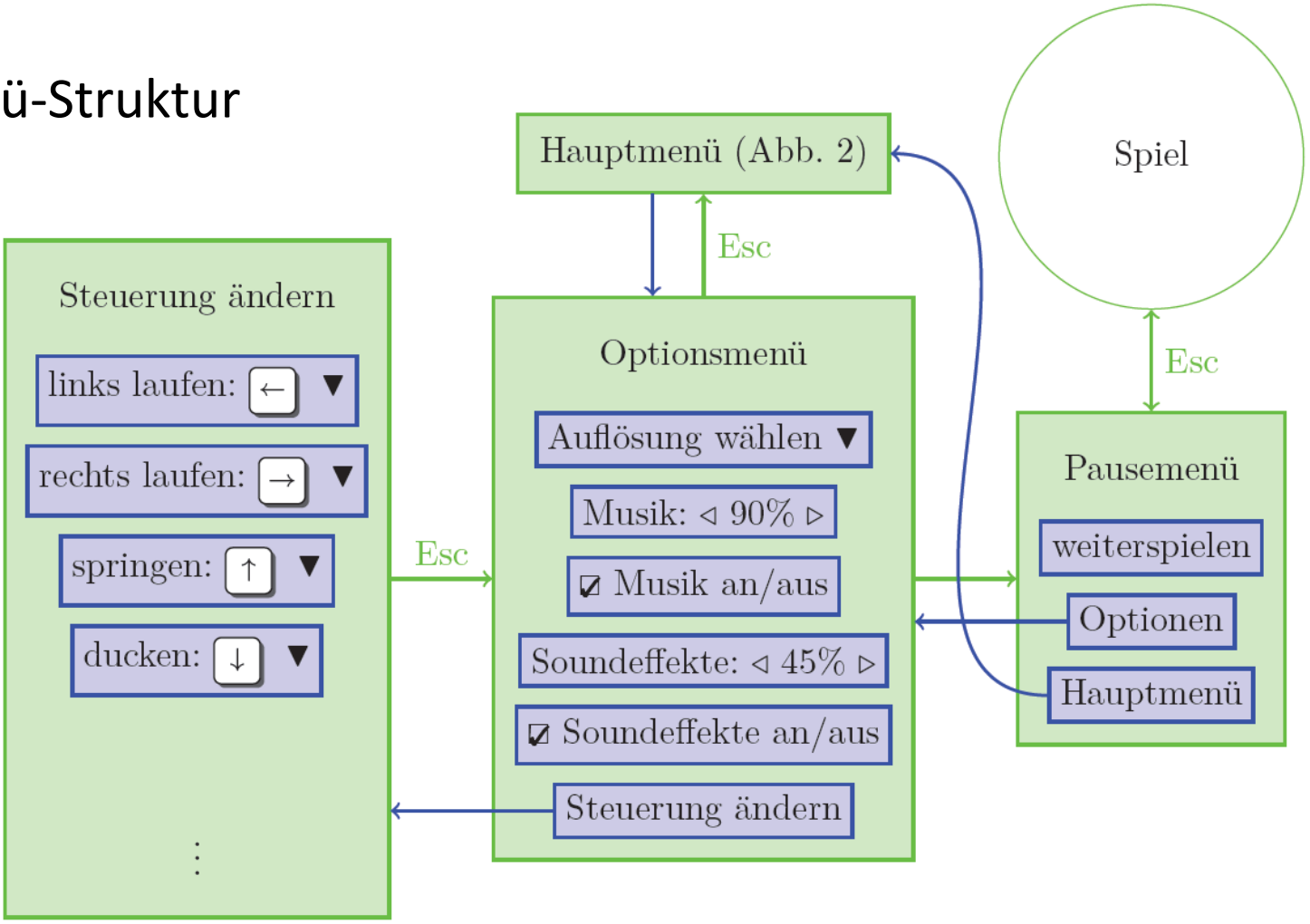
Benutzeroberfläche – Beispiel

Menü-Struktur



Benutzeroberfläche – Beispiel

Menü-Struktur



Benutzeroberfläche – Beispiel

- Graph der Menüstruktur muss auch textuell erklärt werden!

Nach dem Starten des Programms wird zunächst das Hauptmenü angezeigt, welches die folgenden, in Abbildung 2 beschriebenen, Auswahlmöglichkeiten bietet. „Spiel starten“ führt in ein Untermenü, in welchem sich der Schwierigkeitsgrad des Spiels (leicht, normal, oder schwer) auswählen lässt. Wählt der Spieler einen Schwierigkeitsgrad aus, wird ein neues Spiel gestartet.
Über „Spiel laden“ kann über ein Untermenü ein bereits vorhandener Spielstand geladen werden. Der Menüpunkt „Statistiken“ zeigt die gesammelten Statistiken aller Spielsizungen an.

…

GDD

- Deckblatt
- Spielkonzept
  - Zusammenfassung des Spiels
  - Alleinstellungsmerkmal
- Benutzeroberfläche
  - Spieler-Interface
  - Menü-Struktur
- Technische Merkmale
  - Verwendete Technologien
  - Mindestvoraussetzungen
- Spiellogik
  - Optionen & Aktionen
  - Spielobjekte
  - Spielstruktur
  - Statistiken
  - Achievements
- Screenplay
  - Konzeptzeichnungen & Storyboards

GDD
Technische Merkmale

- Verwendete Technologien
  - Was wird/wurde zur Erstellung des Spiels benutzt?
    - Programmiersprache(n)
    - Programme zum Erstellen von Grafiken, Modellen, Sounds, usw.
    - Zusätzliche Programme: z.B. Physik Engines, usw.
- Mindestvoraussetzungen
  - Hardware, externe Bibliotheken, usw. die zum Spielen benötigt werden.
  - Siehe Spieleverpackungen.
  - Achtung: Spiel muss auf Poolrechnern lauffähig sein.

Technische Merkmale – Beispiel

Verwendete Technologien

- Microsoft C#
- MonoGame 3.6
- Visual Studio 2015 mit ReSharper 2017.1
- GIMP 2.6
- Cross-Platform Audio Creation Tool (XACT)
- Cinema4D
- Wavelab
- Tiled Mapeditor
- SomeFancyHackerTool

Technische Merkmale – Beispiel

Mindestvoraussetzungen

- Windows XP SP2
- Monitor mit einer Auflösung von mindestens 1024x768 Bildpunkten
- .NET Framework 4.7
- Microsoft DirectX 9.0c
- Dual-Core Prozessor mit mindestens 2.0 GHz
- 2 GB RAM
- Grafikkarte mit mindestens Shader Model 2.0
- Maus und Tastatur
- Internetverbindung mit mindestens 1 Mbit/s synchroner Übertragungsgeschwindigkeit

GDD

- Deckblatt
- Spielkonzept
  - Zusammenfassung des Spiels
  - Alleinstellungsmerkmal
- Benutzeroberfläche
  - Spieler-Interface
  - Menü-Struktur
- Technische Merkmale
  - Verwendete Technologien
  - Mindestvoraussetzungen
- Spiellogik
  - Optionen & Aktionen
  - Spielobjekte
  - Spielstruktur
  - Statistiken
  - Achievements
- Screenplay
  - Konzeptzeichnungen & Storyboards

GDD
Spiellogik

- Beschreibung der gesamten Spielmechanik und aller Inhalte.
- Nicht vergessen: Gewinnen und Verlieren.

- Abschnitte:
  - Optionen & Aktionen
  - Spielobjekte
  - Spielstruktur
  - Statistiken
  - Achievements

GDD
Spiellogik

- Optionen & Aktionen
  - Welche Möglichkeiten hat der Spieler? Was erlaubt das Spiel?
  - Welche Aktionen können durchgeführt werden?
  - Was sind die Bedingungen für jede Option/Aktion?
  - Wie verändert sich der Zustand des Spiels bei Durchführung von jeder Aktion?

GDD
Spiellogik

- Optionen & Aktionen
  - Tabellarische Auflistung aller Aktionen, die im Spiel vorhanden sind.
  - Auflistung orientiert sich an Anwendungsfällen (Use Cases).
    - Identifikation der Akteure (z.B. Spieler und KI-Player).
    - Der Ereignisfluss (Abfolge von idealerweise max. 9 Ereignissen), der zu einer Aktion gehört, wird aufgelistet.
    - Beschreibung der Anfangs- und Abschlussbedingungen der jeweiligen Aktion.
  - Optionen ergeben sich aus der Beschreibung der Ereignisse.

Spiellogik – Beispiel

Optionen & Aktionen

ID/Name	Akteure	Ereignisfluss	Anfangsbedingungen	Abschlussbedingungen
A01: Laufen	Spieler	1. ← oder → wird gedrückt. 2. Die Hauptfigur läuft in die gewählte Richtung, während die Taste gedrückt ist. 3. Wird die Taste losgelassen, stoppt die Hauptfigur.	In der gewählten Laufrichtung darf sich keine Wand befinden	Die Hauptfigur befindet sich an einem Ort, dessen Position durch die jeweilige Richtungstaste bestimmt wurde.
A02: Springen aus dem Stand	Spieler	1. Es wird die ↑-Taste gedrückt. 2. Die Hauptfigur springt für eine gewisse Zeit nach oben. 3. Die Hauptfigur fällt.	Es befindet sich kein Hindernis direkt oberhalb der Hauptfigur.	Die Hauptfigur befindet sich auf der Ursprungsposition.
A03: Schießen	Spieler	1. Ctrl oder Alt wird gedrückt. 2. Ein Schuss wird ausgelöst, der von der Hauptfigur ausgeht und in ihre Blickrichtung abgefeuert wird. 3. Der Schuss trifft auf ein Hindernis: • Ist das Hindernis eine Plattform oder der Rand der Spielwelt, verschwindet der Schuss und es passiert nichts. • Ist das Hindernis ein gegnerisches Spielobjekt, so wird dieses getötet (siehe Aktion „A06: Gegner töten“).	Als Primär- oder Sekundärfähigkeit muss die Fähigkeit „Schuss“ ausgewählt sein.	Der Schuss hat keine Auswirkungen auf die Spielwelt ODER Der Schuss hat ein gegnerisches Spielobjekt getötet.

Spiellogik – Beispiel

Optionen & Aktionen

ID/Name	Akteure	Ereignisfluss	Anfangsbedingungen	Abschlussbedingungen
A04: Fallen	Spieler	**Ablauf 1** • Der Spieler hat Fähigkeit „Langsamer Fall“ nicht aktiviert: 1. Die Hauptfigur fällt nach unten, wobei sich ihre Fallgeschwindigkeit stetig erhöht. 2. Mögliche Szenarien: 1. Die Hauptfigur trifft auf eine Plattform und der Fall ist gestoppt. 2. Die Hauptfigur fällt in einen Abgrund und stirbt. **Ablauf 2** • Der Spieler hat Fähigkeit „Langsamer Fall“ aktiviert: 1. Die Hauptfigur fällt nach unten, wobei die Fallgeschwindigkeit sich nicht über einen bestimmten Wert erhöht. 2. Mögliche Szenarien: 1. Die Hauptfigur trifft auf eine Plattform und der Fall ist gestoppt. 2. Die Hauptfigur fällt in einen Abgrund und stirbt.	Die Hauptfigur befindet sich in der Luft.	Die Hauptfigur befindet sich auf einer Plattform ODER die Hauptfigur ist gestorben.

Spiellogik – Beispiel

Optionen & Aktionen

ID/Name	Akteure	Ereignisfluss	Anfangsbedingungen	Abschlussbedingungen
A05: Verfolgen	KI	1. Ein verfolgender Gegner beginnt, die Hauptfigur zu verfolgen. 2. Mögliche Szenarien: • Der Gegner erreicht die Hauptfigur und diese stirbt. • Der Gegner wird vom Spieler getötet (siehe Aktion „A06: Gegner töten“).	Die Hauptfigur ist in der Nähe eines Gegners vom Typ „verfolgender Gegner“.	Der Gegner ist getötet ODER die Hauptfigur ist getötet
…	…	…	…	…

GDD
Spiellogik

- Bei Bedarf können auch zusätzliche Unterabschnitte eingefügt werden.

- Unterabschnitte werden benötigt, wenn Optionen & Aktionen nicht ausreichen, um das Spielverhalten genügend zu beschreiben.

Spiellogik – Beispiel

Fähigkeiten

ID/Name	Eigenschaften	Bedingung
F01: Schuss	Aktive Fähigkeit, bei Benutzung wird ein Projektil in Blickrichtung abgefeuert.	Das Item „Schuss“ muss eingesammelt worden sein.
F02: Langsamer Fall	Passive Fähigkeit, verlangsamt die Fallgeschwindigkeit.	Das Item „Langsamer Fall“ muss eingesammelt worden sein.

GDD
Spiellogik

- Abschnitt Spielobjekte:
  - Beschreiben aller im Spiel vorhandenen Objekte
    - z.B. neutrale, gegnerische, eigene, steuerbare, nicht-steuerbare, szenische Objekte usw.
  - Angeben der Eigenschaften jedes Spielobjekts
    - z.B. Hitpoints, Geschwindigkeit, besondere Fähigkeiten (mit Referenz zu Fähigkeiten-Tabelle), usw.
  - Eigenschaften-Werte im GDD müssen noch nicht fest sein, sie sollen aber einen Eindruck über Verhältnisse geben.
  - Angabe der Spielobjekte in tabellarischer Form.

Spiellogik – Beispiel

- Spielobjekte in Superjumper:
  - Einheiten
    - Sind vom Spieler gesteuerte Einheiten, gegnerische Einheiten, oder neutrale Einheiten
  - Items
    - Einsammelbare Gegenstände, zum Beispiel Powerups, Münzen, usw.
  - Plattformen
    - Plattformen mit unterschiedlichen Eigenschaften (z.B. andere Haftreibungskräfte)

Spiellogik – Beispiel

Spielobjekte: Einheiten

Zu Einheiten zählen alle Arten von Gegnern und die vom Spieler gesteuerte Spielfigur. Einheiten sind bewegliche Spielobjekte. Die Eigenschaften aller Einheiten sind in Tabelle 2 aufgelistet.

Eigenschaft	Beschreibung
Max-Geschwindigkeit	Die maximale Geschwindigkeit, mit der sich eine Einheit bewegen kann.
HP	HP (Healthpoints) stellen die Lebenspunkte der Einheit dar. Übertrifft erhaltender Schaden den HP-Wert, so wird die Einheit vernichtet.
Schaden	Der Schadenswert, den die Einheit austeilen kann (siehe Abschnitt 4.2).

Tabelle 2: Eigenschaften von Einheiten

Spiellogik – Beispiel

Spielobjekte: Einheiten

Jumper (Spielfigur)	
	
Beschreibung	Jumper ist der Zustand der Spielfigur, die der Spieler steuert, ohne dass Fähigkeiten aktiv sind. Jumper kann zu Superjumper werden, wenn ein entsprechendes Item gesammelt wird (siehe 4.2).
Max-Geschwindigkeit	8
HP	2
Schaden	1

Spiellogik – Beispiel

Spielobjekte: Einheiten

Superjumper (Spielfigur)	
	
Beschreibung	Superjumper ist ein Zustand der Spielfigur mit verbesserten Eigenschaften, wenn das Item *Super* eingesammelt wurde (siehe Abschnitt 4.2).
Max-Geschwindigkeit	10
HP	2
Schaden	2

Ghumba (Gegner)	
	
Beschreibung	Ein simpler Gegner, der fortwährend in eine Richtung läuft, bis dieser mit einem anderen gegnerischen Spielobjekt oder einer Plattform kollidiert. Die Bewegungsrichtung wird dann umgekehrt. Kollidiert der Gegner mit der Hauptfigur des Spielers, wird der Hauptfigur Schaden zugefügt.
Max-Geschwindigkeit	5
HP	1
Schaden	1

GDD
Spiellogik

- Spielstruktur
  - Beschreibt den Ablauf des Spiels.
  - Was geschieht, wenn der Spieler ein neues Spiel startet?
  - Wie entwickelt sich das Spiel von dort aus?
  - Wann hat der Spieler gewonnen oder verloren?
  - Beschreibung der Spielphasen vom Start des Spiels bis zum Gewinnen/Verlieren.
    - z.B. für (Echtzeit-)Strategiespiele:
      - Early-Game (Eröffnung)
      - Mid-Game (Festlegen der Strategischen Positionen)
      - Late-Game (Gewinnstrategie durchführen)

Spiellogik – Beispiel

Spielstruktur

Das Spiel besteht aus einer Reihe von Levels. Wird ein Level abgeschlossen, so wird das darauf folgende Level geladen. In jedem zehnten Level befindet sich kurz vor dem Zielpunkt ein Bossgegner.

In jedem Level wird immer von links nach rechts gespielt, so dass sich der Startpunkt der Spielfigur am linken Levelrand befindet. Ziel eines jeden Levels ist es den Zielpunkt zu erreichen, der jeweils am rechten Levelrand platziert ist. Im Verlauf des Levels befinden sich verschiedene Hindernisse, die die Spielfigur überwinden muss, um ins Ziel zu gelangen.

Spiellogik – Beispiel

Spielstruktur

Das Spiel besteht aus einer Reihe von Levels. Wird ein Level abgeschlossen, so wird das darauf folgende Level geladen. In jedem zehnten Level befindet sich kurz vor dem Zielpunkt ein Bossgegner.

In jedem Level wird immer von links nach rechts gespielt, so dass sich der Startpunkt der Spielfigur am linken Levelrand befindet. Ziel eines jeden Levels ist es den Zielpunkt zu erreichen, der jeweils am rechten Levelrand platziert ist. Im Verlauf des Levels befinden sich verschiedene Hindernisse, die die Spielfigur überwinden muss, um ins Ziel zu gelangen.

Spiellogik – Beispiel

Spielstruktur

Hindernisse können zum Beispiel Abgründe sein, in die die Spielfigur nicht fallen darf, da diese sonst stirbt und ein Leben verliert. Wenn die Spielfigur keine Leben mehr hat, so hat man das Spiel verloren. Verschiedene Gegner, denen die Spielfigur ausweichen oder die der Spieler – je nach Gegnertyp – besiegen muss, sind weitere Hindernisse auf dem Weg ins Ziel, die der Spielfigur Leben kosten können.
Zudem gibt es in jedem Level mindestens einen Rätselabschnitt, der nur überwunden werden kann, wenn man seine Primär- und Sekundärfähigkeiten richtig einsetzt.

Hat man den Zielpunkt eines Levels erreicht, so hat man dieses abgeschlossen und es wird das nächste Level geladen. Beim Erreichen des Zielpunkts des letzten Levels hat man das Spiel gewonnen. …

Spiellogik – Beispiel

Spielstruktur

Hindernisse können zum Beispiel Abgründe sein, in die die Spielfigur nicht fallen darf, da diese sonst stirbt und ein Leben verliert. Wenn die Spielfigur keine Leben mehr hat, so hat man das Spiel verloren. Verschiedene Gegner, denen die Spielfigur ausweichen oder die der Spieler – je nach Gegnertyp – besiegen muss, sind weitere Hindernisse auf dem Weg ins Ziel, die der Spielfigur Leben kosten können.
Zudem gibt es in jedem Level mindestens einen Rätselabschnitt, der nur überwunden werden kann, wenn man seine Primär- und Sekundärfähigkeiten richtig einsetzt.

Hat man den Zielpunkt eines Levels erreicht, so hat man dieses abgeschlossen und es wird das nächste Level geladen. Beim Erreichen des Zielpunkts des letzten Levels hat man das Spiel gewonnen. …

GDD
Spiellogik

- Statistiken:
  - Statistiken sind gut geeignet, damit sich Spieler gegenseitig messen können und um die Langzeitmotivation am Spiel zu erhöhen.
  - Der Abschnitt beschreibt,
    - welche Statistiken im Spielverlauf gesammelt werden,
    - wie die gesammelten Statistiken Einfluss auf das Spielgeschehen nehmen und
    - wodurch sich die unterschiedlichen gesammelten Werte während des Spielverlaufes ändern.
  - Statistiken können außerdem Highscorelisten sein.
  - Aufschrieb kann tabellarisch erfolgen, um Übersichtlichkeit zu erhöhen.

Spiellogik – Beispiel

Statistiken

Die folgenden Statistiken werden in Superjumper gesammelt und am Ende jedes Levels angezeigt:

- Verbrauchte Zeit
- Getötete Gegner
- Gesammelte Items
- Meistbenutzte Items.

Aus den unterschiedlichen Statistiken errechnet sich eine Gesamtpunktzahl für ein Level nach folgender Formel:

Verbrauchte Zeit in Sekunden * 0.2 + Getötete Gegner * 10 + Gesammelte Items * 3

Außerdem werden die gesammelten Punkte für jedes Level in einer Highscoreliste gespeichert, die vom Hauptmenü aus über den Menüpunkt „Statistiken“ aufgerufen werden kann.

GDD
Spiellogik

- Achievements:
  - Motivation für den Spieler, ähnlich zu Statistiken.
  - Achievements beschreiben besondere Erfolge, die der Spieler erreichen kann.
  - Die Bedingungen, um ein Achievement zu erreichen, sind oft unterschiedlich schwer zu erfüllen.
  - Achievements können oft (ohne Reset) nur einmalig erreicht werden.
  - Achievements bleiben bestehen, auch wenn das Spiel neu gestartet wird.
  - Aufschrieb kann Tabellarisch erfolgen.

Achievements – Beispiel

Achievements

Die erreichbaren Achievements sind in Tabelle 4 aufgelistet.

Titel	**Bedingung**
Du willst der Beste sein!	Alle Münzen im Spiel einsammeln.
Don't touch me	Das Spiel besiegen, ohne einen einzigen regulären Gegner (Bosse ausgenommen) zu töten.
Who's your Daddy?	Den ersten Bossgegner besiegen, ohne auf den Boden zu springen.
Meatgrinder	5000 Gegner töten.

Tabelle 4: Achievements in Superjumper

GDD

- Deckblatt
- Spielkonzept
  - Zusammenfassung des Spiels
  - Alleinstellungsmerkmal
- Benutzeroberfläche
  - Spieler-Interface
  - Menü-Struktur
- Technische Merkmale
  - Verwendete Technologien
  - Mindestvoraussetzungen
- Spiellogik
  - Optionen & Aktionen
  - Spielobjekte
  - Spielstruktur
  - Statistiken
  - Achievements
- Screenplay
  - Konzeptzeichnungen & Storyboards

GDD

- Screenplay
  - Enthält die Spielgeschichte und den Hintergrund des Spiels.
  - Beschreibt die Kampagne (falls vorhanden).
  - Kann zusätzlich Konzeptzeichnungen und/oder Storyboards enthalten.

VORGEHEN

Vorgehen

- Erzählen Sie sich gegenseitig einen exemplarischen Spielablauf.
  - Erzählen Sie sich nicht nur den Anfang, des Spiels, sondern den gesamten Spielablauf („Klick auf EXE“ → „Gewonnen/Verloren“).
- Überlegen Sie, welche Inhalte des Spiels erwähnt werden und formulieren Sie daraus den Abschnitt „Spiellogik“.
- Daraus ergibt sich der Rest des GDDs.

Vorgehen – Beispiel (StarCraft)

- „Der Spieler beginnt in seiner Hauptbasis damit, mit seinen Arbeitern Mineralien abzubauen.“
  → Man braucht Spielobjekte Hauptbasis, Arbeiter, Mineralien.
  → Man braucht Aktion: Mineralien abbauen.

- „Danach beginnt der Spieler, seine Basis aufzubauen.“
  → Man braucht mehrere Basis-Spielobjekte.
  → Man braucht Aktion: Gebäude bauen.

- „Der Spieler beginnt, Einheiten zu bauen.“
  → Man braucht einen Begriff für Einheiten.
  → Man muss Einheiten definieren und deren Eigenschaften festlegen.
  → Man braucht Aktion: Einheiten bauen.

Vermeidbare Verbesserungsarbeit am GDD

- Leere Referenzen vermeiden: Nicht auf etwas verweisen, das (noch) nicht existiert (Abschnitte, Objekte, usw.).
- Keine Vorwärtsreferenzen verwenden.
- Abbildungen und Tabellen benötigen einen Titel und müssen aus dem Text referenziert und erklärt werden.
- Konsistenz: gleiche Dinge müssen gleich benannt werden.
- Vermeiden von Mehrdeutigkeiten.
- Spielbeschreibung muss logisch und nachvollziehbar sein.

VOM GDD ZUM PRODUCT BACKLOG

Vom GDD zum Product Backlog

1. Erstellen von Trac-Requirements aus Spieleigenschaften aus dem GDD.
2. Ableiten von User Stories (aus „Spiellogik“, „Zusammenfassung“, …).
3. Ermitteln Story Points für User Stories.
4. Aufteilen der User Stories in unterschiedliche Tasks, die dann implementiert werden.

Vom GDD zum Product Backlog

- Trac-Requirements haben Business Values.
- Business Values
  - sind ein Schätzwert für die Wichtigkeit eines Trac-Requirements relativ zu anderen Trac-Requirements.
  - geben damit eine zweite Priorisierung der Trac-Requirements im Product Backlog vor.
    - Der Product Owner kann mit Hilfe der Business Values entscheiden, welches Trac-Requirement als nächstes bearbeitet werden soll.

Vom GDD zum Product Backlog

- Business Values sollten über eine der folgenden Kategorien gerechtfertigt werden:
  - New Business: Features, die potentiell neue Kunden gewinnen oder neue Märkte erschließen lassen.
  - Up Sell: Features, die potentiell Geld von bereits existierenden Kunden einbringen und als „Add-On“, „Upgrade“ oder „Plug-In“ verkauft werden können.
  - Retainment: Features, die verhindern, dass Kunden aufhören, das Produkt zu benutzen.
  - Operational Efficiency: Features, die Kosten reduzieren.

[Requirements Engineering Qualifications Board, “Syllabus: REQB® Certified Professional for Requirements Engineering – Agile Practitioner”, Global Association for Software Quality, GASQ, 2015.]

Vom GDD zum Product Backlog

- Story Points
  - geben an, wie schwierig es ist, die User Story fertig zu stellen (Schätzwert).
  - erlauben das Vergleichen mit anderen User Stories.
    - „User Story X ist schwieriger umzusetzen als User Story Y.“
  - Es gibt unterschiedliche Methoden, um Story Points zu ermitteln, z.B. Planning Poker.

Vom GDD zum Product Backlog

- Beispiel:

A03: Schießen	Spieler	1. Ctrl oder Alt wird gedrückt. 2. Ein Schuss wird ausgelöst, der von der Hauptfigur ausgeht und in ihre Blickrichtung abgefeuert wird. 3. Der Schuss trifft auf ein Hindernis: • Ist das Hindernis eine Plattform oder der Rand der Spielwelt, verschwindet der Schuss und es passiert nichts. • Ist das Hindernis ein gegnerisches Spielobjekt, so wird dieses getötet (siehe Aktion „Töten von gegnerischen Spielobjekten“).	Als Primär- oder Sekundärfähigkeit muss die Fähigkeit „Schuss“ ausgewählt sein.	Der Schuss hat keine Auswirkungen auf die Spielwelt ODER Der Schuss hat ein gegnerisches Spielobjekt getötet.

- Trac-Requirement: „‚A03: Schießen‘ implementieren“
- User Story 1: „Wenn ich als Spieler die Fähigkeit Schuss ausgewählt habe, möchte ich mit Alt oder Ctrl schießen können, um Gegner zu töten.“ (Story Points: 100)
- User Story 2: „Wenn ich als Spieler schieße, möchte ich in Blickrichtung schießen, um zielen zu können.“ (Story Points: 50)

Vom GDD zum Product Backlog

- User Story 1: „Wenn ich als Spieler die Fähigkeit Schuss ausgewählt habe, möchte ich mit Alt oder Ctrl schießen können, um Gegner zu töten.“ (Story Points: 100)

- Mögliche Tasks für User Story 1:
  - Task 1: Implementieren der Fähigkeit „Schuss“.
  - Task 2: Implementieren eines Tastendrucks.
  - Task 3: „Sterben“ eines Gegners.
  - Task 4: Implementieren der Kollisionsabfrage, ob der Schuss getroffen hat.
  - Task 5: …

Vom GDD zum Product Backlog

- Je genauer die Beschreibungen in der Spiellogik sind
  - desto einfacher geht das Erstellen von Trac-Requirements, User Stories und Tasks,
  - desto einfacher fällt die Abschätzung der Schwierigkeit jeder einzelnen Aufgabe,
  - desto leichter fällt Ihnen die Abschätzung, ob Sie den Task innerhalb eines Sprints bewältigen können und
  - desto einfacher ist ein Entwurf der Architektur.

Wiederkehrende Aufgaben

- Ab dem zweitem Sprint (Woche 2):
  + Product Owner Tasks:
    - Pflegen und Anpassen der Trac-Requirements und User Stories im Product Backlog.
    - Verfeinern der Trac-Requirements zu User Stories.
    - Trac-Requirements nach Entwicklungsreife ordnen.
    - Gruppentreffen vorbereiten (was ist fertig, wie war die Aufwandsabschätzung).
    - Gemeinsam mit der Gruppe bestimmen, was aufgrund der Entwicklungsreife der Trac-Requirements als nächstes getan werden soll.
  + Qualitätssicherung Tasks:
    - Code auf Clean-Code Richtlinien prüfen.
    - ReSharper-Konformität im ganzen Projekt prüfen und ggf. herstellen.

Wiederkehrende Aufgaben

- Ab dem dritten Sprint (Woche 3):
  + Architektur Tasks:
    - Definieren von Schnittstellen.
    - Pflegen von Architekturbeschreibungen.
    - Einhaltung der Architektur im Projekt sicherstellen und beides ggf. anpassen.
    - Einen Überblick über die Gesamtarchitektur behalten und bei Umsetzung verschiedener Tasks die architektonischen Abhängigkeiten stets berücksichtigen.
  + Coaching (nicht notwendig, aber hilfreich):
    - Code Reviews vorbereiten und durchführen.

Sonstiges

- Wiki-Artikel: „GDD“
- Orientierung an den letzten Jahren (Wiki: „Hall of Fame“, SoSe), nur besser.
- Beta-GDD sollte nahezu vollständig sein.
- Pflicht:
  - Vollständiges Deckblatt.
  - Alle hier vorgestellten Abschnitte müssen sich im GDD wiederfinden.
  - Format der Optionen & Aktionen verwenden.

Sonstiges zum Sopra

- UniCard-Freischaltung für den Pool.
- Quota-Erhöhung falls notwendig.
- Fragen Sie!

Was nun?

- Fragebogen ausfüllen bis heute Abend 23:59 Uhr.
- Pool Account besorgen und sicherstellen, dass er funktioniert.
- Hausaufgabe machen bis Sa., 06.05., 23:59 Uhr.
- Termin für Gruppentreffen ausmachen und treffen.
- Spielidee ausarbeiten und GDD schreiben.
- Definition of Done festlegen.

- Vorlesungen:
  - Nächste Woche: Grundlagen SW-Architektur.
  - Übernächste Woche: Architektur in Computerspielen.

FRAGEN?