



# Game Design Document (GDD)



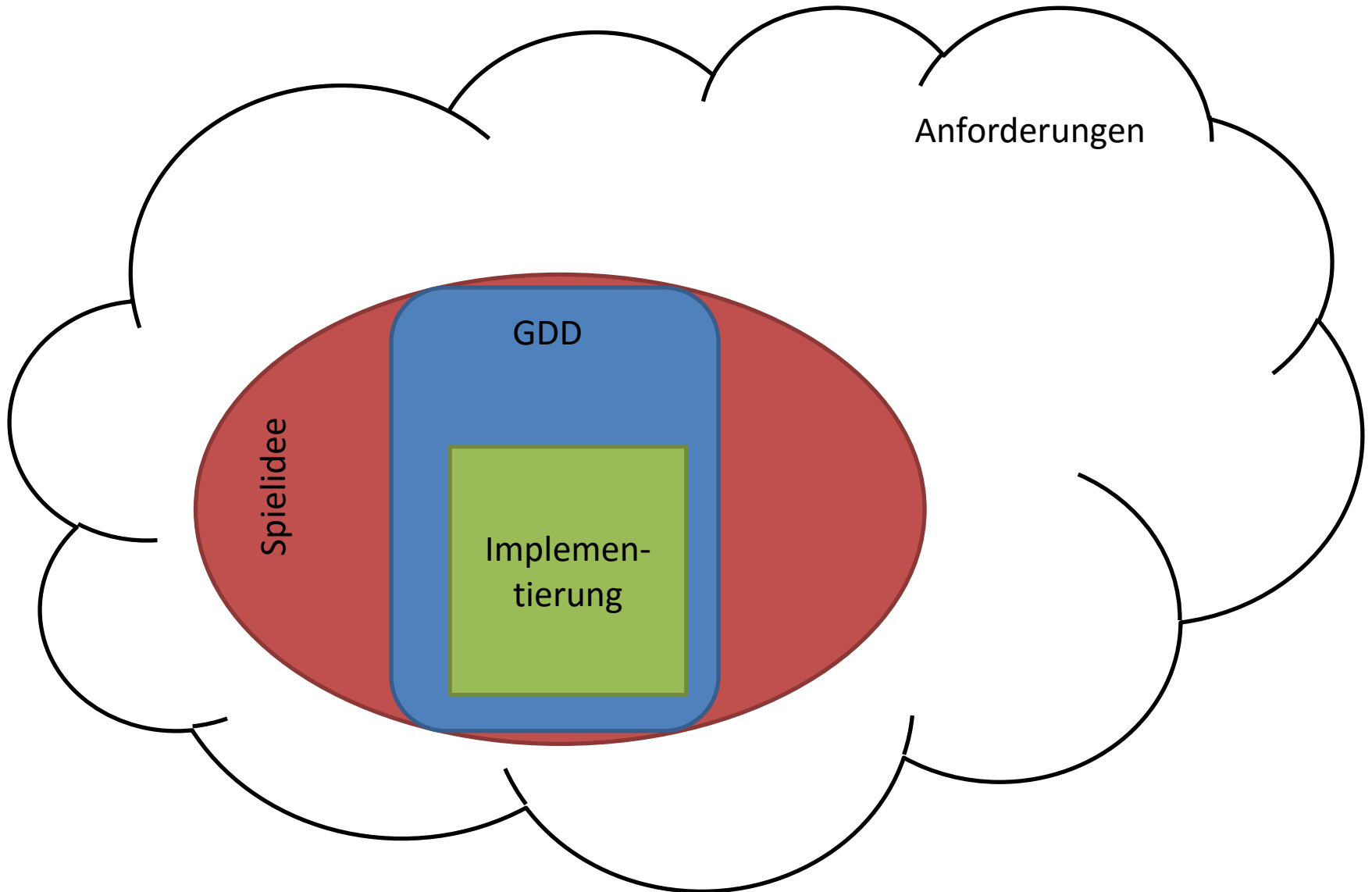
# Game Design Document

- Ein GDD soll einen **genauen Überblick** das Spiel geben.
- Angelehnt an **Lastenheft** in der Softwareentwicklung.
- Das heißt, 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.
- 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.





# 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 allgemeinen 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, Einfallsreichtum und Reaktionsvermögen.



# 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 **Fähigkeitskombination**.

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 Hauptfigur 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 Hauptfigur 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.










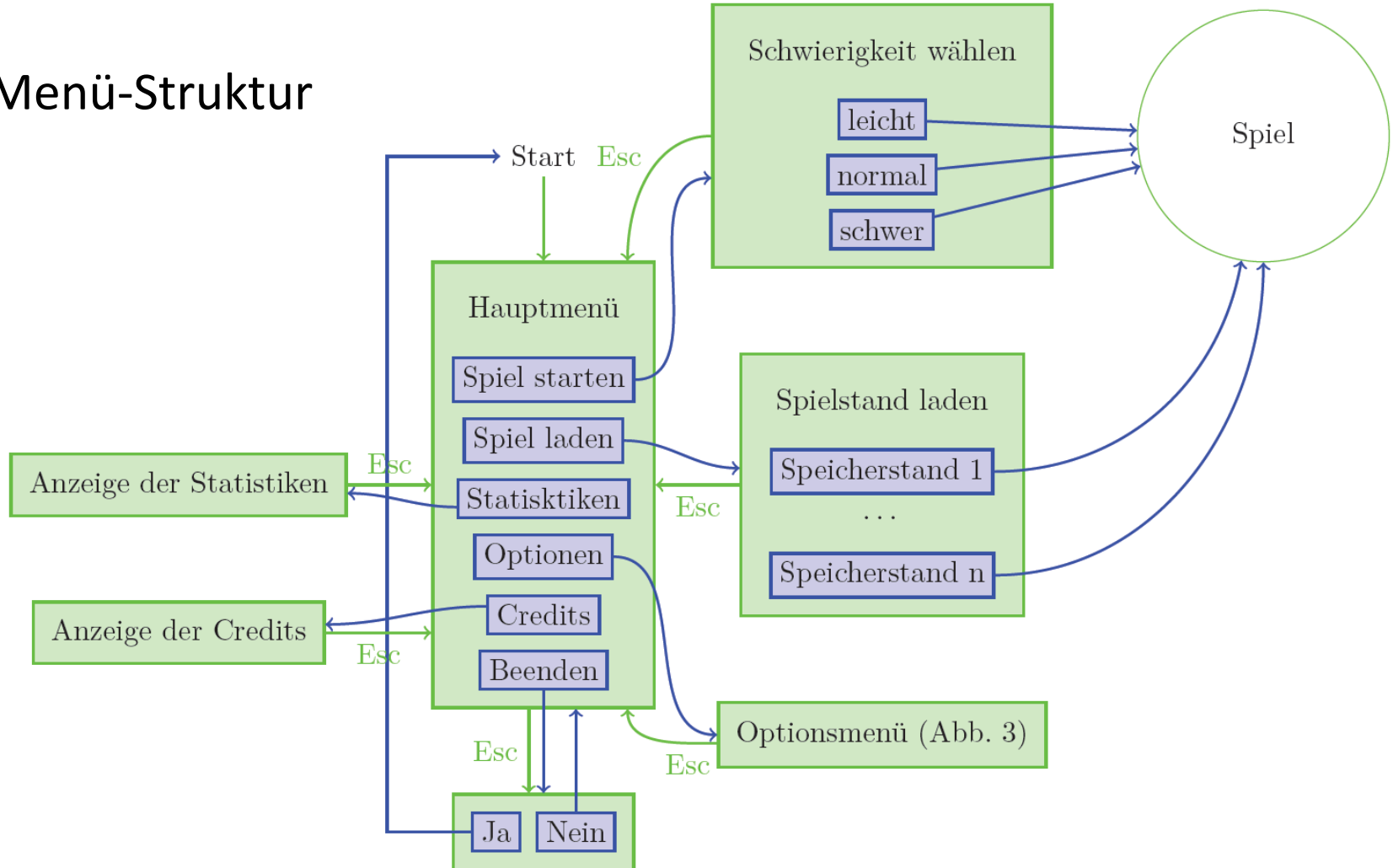
Taste	Beschreibung
	Bewegt die Hauptfigur nach rechts
	Bewegt die Hauptfigur nach links
	Die Hauptfigur springt / schwimmt nach oben
	Die Hauptfigur duckt sich / schwimmt nach unten
	Die Primärfähigkeit wird benutzt
	Die Sekundärfähigkeit wird benutzt
	Die Primär- und Sekundärfähigkeit wird ausgewählt
	Das Spiel wird pausiert
	Das Pausemenü wird aufgerufen

Tabelle 1: Die Steuerung von Superjumper über die Tastatur.

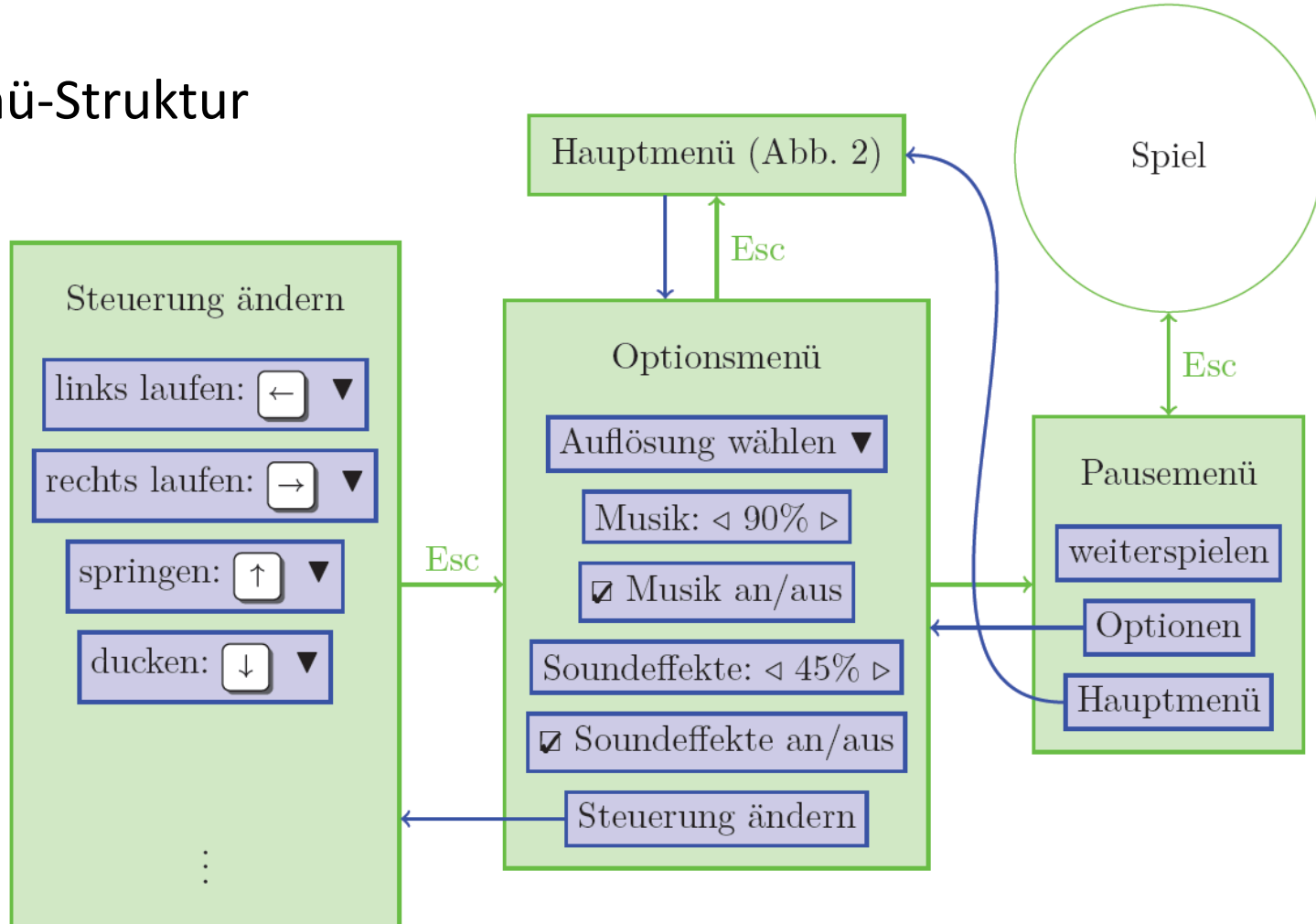
# Benutzeroberfläche – Beispiel

## Menü-Struktur



# Benutzeroberfläche – Beispiel

## Menü-Struktur





# Benutzeroberfläche – Beispiel

- Graph der Menüstruktur muss auch textuell erklärt werden!



# 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

- Übersicht über **Technologien**, die im Spiel verwendet werden.
- 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 3.5
- Visual Studio 2015 mit ReSharper 2016.1
- GIMP 2.6
- Cross-Platform Audio Creation Tool (XACT)
- Cinema4D
- Wavelab
- Tiled Mapeditor





# Technische Merkmale – Beispiel

## Mindestvoraussetzungen

- Windows XP SP2
- Monitor mit einer Auflösung von mindestens 1024x768 Bildpunkten
- .NET Framework 4.6.1
- 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.
- Oft vergessen: **Gewinnen** und **Verlieren**.
- Abschnitte:
  - Optionen & Aktionen
  - Spielobjekte
  - Spielstruktur
  - Statistiken
  - Achievements



# GDD

## Spiellogik

- Optionen & Aktionen
  - Welche **Möglichkeiten** hat der Spieler?
  - 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	<ol style="list-style-type: none"> <li>1.  oder  wird gedrückt.</li> <li>2. Die Hauptfigur läuft in die gewählte Richtung, während die Taste gedrückt ist.</li> <li>3. Wird die Taste losgelassen, stoppt die Hauptfigur.</li> </ol>	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	<ol style="list-style-type: none"> <li>1. Es wird die -Taste gedrückt.</li> <li>2. Die Hauptfigur springt für eine gewisse Zeit nach oben.</li> <li>3. Die Hauptfigur fällt.</li> </ol>	Es befindet sich kein Hindernis direkt oberhalb der Hauptfigur.	Die Hauptfigur befindet sich auf der Ursprungposition.
A03: Schießen	Spieler	<ol style="list-style-type: none"> <li>1.  oder  wird gedrückt.</li> <li>2. Ein Schuss wird ausgelöst, der von der Hauptfigur ausgeht und in ihre Blickrichtung abgefeuert wird.</li> <li>3. Der Schuss trifft auf ein Hindernis: <ul style="list-style-type: none"> <li>• Ist das Hindernis eine Plattform oder der Rand der Spielwelt, verschwindet der Schuss und es passiert nichts.</li> <li>• Ist das Hindernis ein gegnerisches Spielobjekt, so wird dieses getötet (siehe Aktion „A06: Gegner töten“).</li> </ul> </li> </ol>	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	<p><b>Ablauf 1</b></p> <ul style="list-style-type: none"><li>Der Spieler hat Fähigkeit „Langsamer Fall“ nicht aktiviert:<ol style="list-style-type: none"><li>Die Hauptfigur fällt nach unten, wobei sich ihre Fallgeschwindigkeit stetig erhöht.</li><li>Mögliche Szenarien:<ol style="list-style-type: none"><li>Die Hauptfigur trifft auf eine Plattform und der Fall ist gestoppt.</li><li>Die Hauptfigur fällt in einen Abgrund und stirbt.</li></ol></li></ol></li></ul> <p><b>Ablauf 2</b></p> <ul style="list-style-type: none"><li>Der Spieler hat Fähigkeit „Langsamer Fall“ aktiviert:<ol style="list-style-type: none"><li>Die Hauptfigur fällt nach unten, wobei die Fallgeschwindigkeit sich nicht über einen bestimmten Wert erhöht.</li><li>Mögliche Szenarien:<ol style="list-style-type: none"><li>Die Hauptfigur trifft auf eine Plattform und der Fall ist gestoppt.</li><li>Die Hauptfigur fällt in einen Abgrund und stirbt.</li></ol></li></ol></li></ul>	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	<ol style="list-style-type: none"><li>Ein verfolgender Gegner beginnt, die Hauptfigur zu verfolgen.</li><li>Mögliche Szenarien:<ul style="list-style-type: none"><li>Der Gegner erreicht die Hauptfigur und diese stirbt.</li><li>Der Gegner wird vom Spieler getötet (siehe Aktion „A06: Gegner töten“).</li></ul></li></ol>	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 aussteilen 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 <i>Super</i> 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

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 Punkt „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

- **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

- **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.“



# Vom GDD zum Product Backlog

- Beispiel:

A03: Schießen	Spieler	<ol style="list-style-type: none"> <li>1.  oder  wird gedrückt.</li> <li>2. Ein Schuss wird ausgelöst, der von der Hauptfigur ausgeht und in ihre Blickrichtung abgefeuert wird.</li> <li>3. Der Schuss trifft auf ein Hindernis: <ul style="list-style-type: none"> <li>• Ist das Hindernis eine Plattform oder der Rand der Spielwelt, verschwindet der Schuss und es passiert nichts.</li> <li>• Ist das Hindernis ein gegnerisches Spielobjekt, so wird dieses getötet (siehe Aktion „Töten von gegnerischen Spielobjekten“).</li> </ul> </li> </ol>	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

- 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 und
  - desto einfacher ist ein **Entwurf der Architektur**.



# Recurring Task

- Zum Beispiel ab 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).



# Sonstiges

- Wiki-Artikel: „GDD“
- Orientierung an den letzten Jahren (Wiki: „Hall of Fame“),  
nur besser.
- Beta-GDD sollte nahezu **vollständig** sein.
- **Pflicht:**
  - Vollständiges **Deckblatt**.
  - **Alle hier vorgestellten Abschnitte** müssen im GDD vorhanden sein.
  - **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., 30.4., 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?**