

Softwarepraktikum

Daniel Dietsch, *Vincent Langenfeld*, Frank Schüssele

October 21, 2021

[Donnerstag Nachmittag, Büro von Mitarbeiter S.]

Chef:

S., die WiKe AG (Wichtiger Kunde) möchte, dass wir ein Computerspiel für sie produzieren.

S.:

Ein Computerspiel? Das ist etwas Neues, haben wir da Leute für?

Chef:

Nein, aber man soll sich ja nie etwas Neuem gegenüber verschließen.
S., Sie und Ihr Team, Sie bekommen das hin!

[Donnerstag Nachmittag, Büro von Mitarbeiter S.]

Chef:

S., die WiKe AG (Wichtiger Kunde) möchte, dass wir ein Computerspiel für sie produzieren.

S.:

Ein Computerspiel? Das ist etwas Neues, haben wir da Leute für?

Chef:

Nein, aber man soll sich ja nie etwas Neuem gegenüber verschließen.
S., Sie und Ihr Team, Sie bekommen das hin!

- ▶ Im Softwarepraktikum sind Sie Mitarbeiter **S.**
- ▶ Im Softwarepraktikum sind Sie Dozenten die Vertreter der WiKe AG (Wichtiger Kunde AG)
- ▶ Im Softwarepraktikum simulieren wir Softwareentwicklung

Mitarbeiter S. beginnt mit der Arbeit am Projekt.
Die ersten Schritte sind ...

Mitarbeiter S. beginnt mit der Arbeit am Projekt.
Die ersten Schritte sind ...

- ▶ die *Domäne* analysieren und verstehen

Mitarbeiter S. beginnt mit der Arbeit am Projekt.
Die ersten Schritte sind ...

- ▶ die *Domäne* analysieren und verstehen
- ▶ die *Anforderungen* erheben und verstehen

Mitarbeiter S. beginnt mit der Arbeit am Projekt.
Die ersten Schritte sind ...

- ▶ die *Domäne* analysieren und verstehen
- ▶ die *Anforderungen* erheben und verstehen
- ▶ die *Umsetzung* konzipieren

- ▶ Organisatorisches
- ▶ Vorgehen
- ▶ Gesamtablauf
- ▶ Vorgehensmodell: Scrum
- ▶ Scrum im Sopra

Organisatorisches

- ▶ Selbständiges Einarbeiten in ein Gebiet
- ▶ Arbeiten im Team
- ▶ Umgang mit Komplexität
- ▶ Anwendung softwaretechnischer Prinzipien

- ▶ Selbständiges Einarbeiten in ein Gebiet
- ▶ Arbeiten im Team
- ▶ Umgang mit Komplexität
- ▶ Anwendung softwaretechnischer Prinzipien

Wir setzen voraus,
dass Sie bereits
programmieren können!

► *Tutoren*

Leon Holtmeier, Ming Hu, Robin Steiger,
Gerrit Freiwald, Tobias Kolzer, Robin Wu,
Florian Probst, Jakob Sailer, Marvin Sautter

► *Dozenten*

Daniel Dietsch, Frank Schüssele, Vincent Langenfeld

► *Verantwortlich*

Prof. Dr. A. Podelski

- ▶ Gruppen mit 5-7 Studenten
- ▶ 6 ECTS (180h) über 15 Wochen
 - ▶ 14 Arbeitsabschnitte (Sprints), je eine Woche
- ▶ 4 Vorlesungen
- ▶ Termine
 - ▶ 3 Abgaben
 - ▶ 3 Präsentationen vor einem Teil des Kurses (Do. 14-18 Uhr)
 - ▶ 1 Vortrag mit den Dozenten
 - ▶ Wöchentliches Gruppentreffen

- ▶ *Wiki*
- ▶ *Fragen und Informationen*
Gruppenmitglieder, Tutoren, Wiki, Discourse, Sprechstunde und Mattermost
- ▶ *Primärdienste*
Gitea, Git, Jenkins, Sonar, Dashboard, Mailinglisten (sopra-crew@... , sopra-XX@...)
- ▶ *Sekundärdienste*
Mattermost, Discourse
- ▶ *Werkzeuge*
C#, .NET Core 3.1, Monogame 3.8, Visual Studio Community, Resharper



Für das virtuelle Semester

- ▶ *Gruppentreffen*
 - ▶ *Big Blue Button* in Ilias (je Gruppe ein Raum)
- ▶ *Vorlesungen*
 - ▶ *Vorlesungen* je Montags als Aufzeichnung
 - ▶ *Fragesession* am entsprechenden Slot
 - ▶ Links **immer** auf dem Wiki
- ▶ *Sprechstunden*
 - ▶ Nach Absprache (Mention in Gitea, Email, Mattermost)

Mitarbeit

- ▶ Kontinuierliche Mitarbeit während der Sprints:
 - ▶ *Commits* im Git-Repository und
 - ▶ *Aktivität* (Tickets, Kommentare, etc.) in Gitea
 - ▶ Es darf max. 2 Sprints nicht mitgearbeitet werden
- ▶ Im Durchschnitt Aufgaben mit min. *7h geschätzter Arbeitszeit* pro Sprint erledigt

Mitarbeit

- ▶ Kontinuierliche Mitarbeit während der Sprints:
 - ▶ *Commits* im Git-Repository und
 - ▶ *Aktivität* (Tickets, Kommentare, etc.) in Gitea
 - ▶ Es darf max. 2 Sprints nicht mitgearbeitet werden
- ▶ Im Durchschnitt Aufgaben mit min. *7h geschätzter Arbeitszeit* pro Sprint erledigt

ECTS:

$$\frac{6 * 30h}{180h}$$

Termine:

$$\begin{aligned} 4 * 2h &= 8h \\ 3 * 4h &= 12h \\ 14 * 2h &= 26h \\ \hline &48h \end{aligned}$$

Arbeitszeit:

$$\frac{(180h - 48)}{9,43h}$$

Mitarbeit

- ▶ Kontinuierliche Mitarbeit während der Sprints:
 - ▶ *Commits* im Git-Repository und
 - ▶ *Aktivität* (Tickets, Kommentare, etc.) in Gitea
 - ▶ Es darf max. 2 Sprints nicht mitgearbeitet werden
- ▶ Im Durchschnitt Aufgaben mit min. *7h geschätzter Arbeitszeit* pro Sprint erledigt

Gruppentreffen

- ▶ Anwesenheitspflicht
 - ▶ Es darf max. 1x gefehlt werden

ECTS:

$$\frac{6 * 30h}{180h}$$

Termine:

$$4 * 2h = 8h$$

$$3 * 4h = 12h$$

$$\frac{14 * 2h = 26h}{48h}$$

Arbeitszeit:

$$\frac{(180h - 48)}{9,43h}$$

Mitarbeit

- ▶ Kontinuierliche Mitarbeit während der Sprints:
 - ▶ *Commits* im Git-Repository und
 - ▶ *Aktivität* (Tickets, Kommentare, etc.) in Gitea
 - ▶ Es darf max. 2 Sprints nicht mitgearbeitet werden
- ▶ Im Durchschnitt Aufgaben mit min. *7h geschätzter Arbeitszeit* pro Sprint erledigt

Gruppentreffen

- ▶ Anwesenheitspflicht
 - ▶ Es darf max. 1x gefehlt werden

Präsentationen

- ▶ Anwesenheitspflicht

ECTS:

$$\frac{6 * 30h}{180h}$$

Termine:

$$4 * 2h = 8h$$

$$3 * 4h = 12h$$

$$\frac{14 * 2h = 26h}{48h}$$

Arbeitszeit:

$$\frac{(180h - 48)}{9,43h}$$

Note

Berechnet sich aus:

- ▶ 50% Endprodukt
- ▶ 50% Einzelnote

Endprodukt bewertet nach Kriterien:

F eatures
A rtefakte
U sability
S pass
T echdemo

Einzelnote

- ▶ Pro Sprint bekommt jeder Studierende 5 Punkte
 - ▶ Zugeteilte Arbeit erledigt?
 - ▶ Note ergibt sich aus Punkten

Vorgehen

Mitarbeiter *S.* beginnen mit der Arbeit am Projekt.
Sie beginnen damit ...

- ▶ die *Domäne* analysieren und verstehen
- ▶ die *Anforderungen* erheben und verstehen
- ▶ das *Produkt* entwerfen

Domäne

Unter einer (Problem)-Domäne versteht man [...] in der Softwaretechnik ein abgrenzbares Problemfeld oder einen bestimmten Einsatzbereich für Computersysteme oder Software.

Domäne

Unter einer (Problem)-Domäne versteht man [...] in der Softwaretechnik ein abgrenzbares Problemfeld oder einen bestimmten Einsatzbereich für Computersysteme oder Software.

- ▶ *Entität* (z.B.: Tastatur, Spiel, Spieler),
- ▶ *Funktionen* (z.B.: gedrückte Tasten, Hardwaretemperatur, Spielermotivation, Flow)
- ▶ *Ereignisse* (z.B.: Einschlafen, Einschalten)
- ▶ *Interaktionen* (z.B.: Spieler drückt Taste, Spiel zerstört Grafikkarte)

Domäne

Unter einer (Problem)-Domäne versteht man [...] in der Softwaretechnik ein abgrenzbares Problemfeld oder einen bestimmten Einsatzbereich für Computersysteme oder Software.

- ▶ *Entität* (z.B.: Tastatur, Spiel, Spieler),
- ▶ *Funktionen* (z.B.: gedrückte Tasten, Hardwaretemperatur, Spielermotivation, Flow)
- ▶ *Ereignisse* (z.B.: Einschlafen, Einschalten)
- ▶ *Interaktionen* (z.B.: Spieler drückt Taste, Spiel zerstört Grafikkarte)
- ▶ Aus Sicht verschiedener *Stakeholder*
 - ▶ Stakeholder sind in irgendeiner Weise von der Domäne betroffen
 - ▶ z.B.: Programmierer, Spieledesigner, Freiwillige Selbstkontrolle (FSK), Hardwarehersteller

Domäne

Unter einer (Problem)-Domäne versteht man [...] in der Softwaretechnik ein abgrenzbares Problemfeld oder einen bestimmten Einsatzbereich für Computersysteme oder Software.

- ▶ *Entität* (z.B.: Tastatur, **Spiel**, **Spieler**),
- ▶ *Funktionen* (z.B.: gedrückte Tasten, Hardwaretemperatur, **Spielermotivation**, **Flow**)
- ▶ *Ereignisse* (z.B.: Einschlafen, Einschalten)
- ▶ *Interaktionen* (z.B.: Spieler drückt Taste, Spiel zerstört Grafikkarte)
- ▶ Aus Sicht verschiedener *Stakeholder*
 - ▶ Stakeholder sind in irgendeiner Weise von der Domäne betroffen
 - ▶ z.B.: Programmierer, **Spieledesigner**, Freiwillige Selbstkontrolle (FSK), Hardwarehersteller

Vier grundlegende Elemente eines Spiels¹

- ▶ *Ästhetik*
- ▶ *Handlung*
- ▶ *Mechanik*
- ▶ *Technologie*

¹Jesse Schell, The Art of Game Design, 2008, ISBN 978-0-12-369496-6

Vier grundlegende Elemente eines Spiels¹

- ▶ *Ästhetik*
 - ▶ *Handlung*
 - ▶ *Mechanik*
 - ▶ *Technologie*
-
- ▶ Jedes Element lässt sich weiter zerlegen
 - ▶ Ästhetik z.B.: Grafische Gestaltung, Stimmung
 - ▶ Handlung z.B.: Akte, Spannungskurve und überleitende Ereignisse
 - ▶ Mechanik z.B.: Spielobjekte mit Werten, Interaktionen

¹ Jesse Schell, The Art of Game Design, 2008, ISBN 978-0-12-369496-6

Vier grundlegende Elemente eines Spiels¹

- ▶ *Ästhetik*
 - ▶ *Handlung*
 - ▶ *Mechanik*
 - ▶ *Technologie*
-
- ▶ Jedes Element lässt sich weiter zerlegen
 - ▶ Ästhetik z.B.: Grafische Gestaltung, Stimmung
 - ▶ Handlung z.B.: Akte, Spannungskurve und überleitende Ereignisse
 - ▶ Mechanik z.B.: Spielobjekte mit Werten, Interaktionen
 - ▶ Zur Beschreibung des Programms (Spiels) sollte jedes dieser Elemente berücksichtigt werden

¹ Jesse Schell, The Art of Game Design, 2008, ISBN 978-0-12-369496-6

Anforderungen²

Anforderungen ist eine Aussage darüber, was man von einem Softwaresystem als Eigenschaften erwartet.

²Helmut Balzert, Lehrbuch der Softwaretechnik - Basiskonzepte und Requirementsengineering, 2009, ISBN 978-3-8274-1705-3

Anforderungen²

Anforderungen ist eine Aussage darüber, was man von einem Softwaresystem als Eigenschaften erwartet.

Gewöhnlich werden *3 Arten* von Anforderungen unterschieden:

- ▶ *Funktionale Anforderungen* legen eine vom Softwaresystem oder einer seiner Komponenten bereitzustellende Funktionen [...] fest.
- ▶ *Qualitätsanforderungen* beschreiben zusätzliche Eigenschaften, die diese Funktionen haben sollen.
- ▶ *Rahmenbedingungen* legen organisatorische und/oder technische Restriktionen für das Softwaresystem und/oder den Entwicklungsprozess fest.

²Helmut Balzert, Lehrbuch der Softwaretechnik - Basiskonzepte und Requirementsengineering, 2009, ISBN 978-3-8274-1705-3

Chef: Die WiKe AG hat die folgenden Anforderungen geschickt.

- ▶ 2D oder 3D Grafik (kein ASCII)
- ▶ Soundeffekte und Musik
- ▶ Mindestens 2 Spieler, min. einer menschlich
- ▶ Echtzeit
- ▶ Indirekte Steuerung (Point & Click)
- ▶ Pausefunktion
- ▶ Eigenes Menü (komplett mit der Maus steuerbar, außer Tastatureingaben)

Chef: Die WiKe AG hat die folgenden Anforderungen geschickt.

- ▶ 2D oder 3D Grafik (kein ASCII)
- ▶ Soundeffekte und Musik
- ▶ Mindestens 2 Spieler, min. einer menschlich
- ▶ Echtzeit
- ▶ Indirekte Steuerung (Point & Click)
- ▶ Pausefunktion
- ▶ Eigenes Menü (komplett mit der Maus steuerbar, außer Tastatureingaben)
- ▶ Spielobjekte
 - a. Min. 5 Kontrollierbare
 - b. Min. 5 Auswählbare
 - c. Min. 5 Nicht kontrollierbare, davon min. 3 Kollidierende
 - d. Min. 3 Kontrollierbare, Kollidierende und Bewegliche

Chef: Die WiKe AG hat die folgenden Anforderungen geschickt.

- ▶ 2D oder 3D Grafik (kein ASCII)
- ▶ Soundeffekte und Musik
- ▶ Mindestens 2 Spieler, min. einer menschlich
- ▶ Echtzeit
- ▶ Indirekte Steuerung (Point & Click)
- ▶ Pausefunktion
- ▶ Eigenes Menü (komplett mit der Maus steuerbar, außer Tastatureingaben)
- ▶ Spielobjekte
 - a. Min. 5 Kontrollierbare
 - b. Min. 5 Auswählbare
 - c. Min. 5 Nicht kontrollierbare, davon min. 3 Kollidierende
 - d. Min. 3 Kontrollierbare, Kollidierende und Bewegliche
- ▶ Min. 5 Statistiken
- ▶ Achievements
- ▶ Min. 1000 gleichzeitig aktive Spielobjekte der Art (d) möglich (Tech-Demo).
- ▶ Speichern und Laden muss potenziell zu jedem Zeitpunkt möglich sein, jedoch nicht zwingend vom Spieler gesteuert werden.

Chef: Die WiKe AG hat die folgenden Anforderungen geschickt.

- ▶ 2D oder 3D Grafik (kein ASCII)
- ▶ Soundeffekte und Musik
- ▶ Mindestens 2 Spieler, min. einer menschlich
- ▶ Echtzeit
- ▶ Indirekte Steuerung (Point & Click)
- ▶ Pausefunktion
- ▶ Eigenes Menü (komplett mit der Maus steuerbar, außer Tastatureingaben)
- ▶ Spielobjekte
 - a. Min. 5 Kontrollierbare
 - b. Min. 5 Auswählbare
 - c. Min. 5 Nicht kontrollierbare, davon min. 3 Kollidierende
 - d. Min. 3 Kontrollierbare, Kollidierende und Bewegliche
- ▶ Min. 5 Statistiken
- ▶ Achievements
- ▶ Min. 1000 gleichzeitig aktive Spielobjekte der Art (d) möglich (Tech-Demo).
- ▶ Speichern und Laden muss potenziell zu jedem Zeitpunkt möglich sein, jedoch nicht zwingend vom Spieler gesteuert werden.
- ▶ Min. 10 verschiedene Aktionen
 - ▶ z.B.: Laufen o. Fähigkeiten
 - ▶ Allen Spielobjekten der Art (d) muss es möglich sein, von jedem beliebigen Punkt in der Welt zu jedem anderen begehbaren Punkt zu gelangen, ohne sich gegenseitig übermäßig zu behindern, festzustecken, usw. ("Pathfinding").

Qualitätsanforderungen

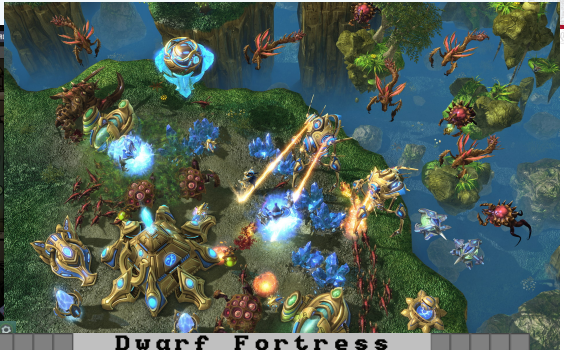
- ▶ Entwickeln Sie ein *gutes* Produkt
- ▶ Qualität der Grafik ist nicht relevant muss aber in sich stimmig sein
- ▶ Akustische Effekte sollen in sich stimmig sein

Rahmenbedingungen

- ▶ C# und/oder F# mit .NET Core 3.1
- ▶ MonoGame 3.8
- ▶ Lauffähig auf Windows 10 (x64)
- ▶ Visual Studio Community
- ▶ Keine Warnings oder Errors vom Compiler oder ReSharper (wöchentlich), keine Buildfehler.

Chef: Die WiKe AG hat die folgenden Anforderungen geschickt.

Anforderungen · Beispiele



Anforderungen · Gegenbeispiele



Game Design Document (GDD)

Beschreibung der *wesentlichen Merkmale* des Spiels.

- ▶ Lastenheft
 - ▶ Dokumentiert die Anforderungen und Rahmenbedingungen eines Produktes
 - ▶ Aus Sicht des *Anwenders* oder *Kunden*
 - ▶ Beinhaltet keine Details über die technische Umsetzung
- ▶ Hilft bei Aufwandsabschätzung und Organisation
- ▶ Dokumentation wie die Anforderungen umgesetzt werden
- ▶ Details: GDD-Vorlesung, Wiki

Ablauf

Woche	Organisation	Entwurf	MS 01	MS 02	MS 03	MS 04	MS 05	Was?	Wann und Wo?
0	✓	✗	✗	✗	✗	✗	✗	<ul style="list-style-type: none"> Vorlesung "Organisation und Prozess" (Einführungsveranstaltung) Gruppeneinteilung abwarten 	<ul style="list-style-type: none"> Einführungsveranstaltung: 21.10., 14:00 - 16:00, Zoom, TBA Fragebogen: 21.10. bis 23:59 Gruppen ab 24.10. in Gitea Gitea
1	✓	✓	✗	✗	✗	✗	✗	<ul style="list-style-type: none"> Vorlesung "Game Design Document (GDD)" Abgabe Hausaufgabe 	<ul style="list-style-type: none"> Fragestunde zur Vorlesung: 28.10., 14:00 - 16:00, Zoom, TBA Abgabe: 30.10 bis 23:59
2	✗	✓	✓	✗	✗	✗	✗	<ul style="list-style-type: none"> Vorlesung "Grundlagen Softwarearchitektur" Wiederkehrende Aufgaben: Product Owner Abgabe GDD (beta) 	<ul style="list-style-type: none"> Fragestunde zur Vorlesung: 04.11., 14:00 - 16:00, Zoom, TBA Abgabe: 06.11. bis 23:59
3	✗	✓	✓	✗	✗	✗	✗	<ul style="list-style-type: none"> Vorlesung "Architektur von Videospielen" Wiederkehrende Aufgaben: Architektur und Coaching Wiederkehrende Aufgaben: Qualitätssicherung und Clean-Code 	<ul style="list-style-type: none"> Fragestunde zur Vorlesung: 11.11., 14:00 - 16:00, Zoom, TBA
4	✗	✓	✓	✓	✗	✗	✗	<ul style="list-style-type: none"> MS01 erreicht (Spielobjekt in der Welt bewegbar, bewegliche Ansichten, Level laden/speichern, Soundausgabe) Präsentation der Spielidee Abgabe Architektur (beta) 	<ul style="list-style-type: none"> Präsentation: 18.11. 14:00 - 16:00, Zoom, TBA Abgabe: 20.11. bis 23:59
5	✗	✓	✗	✓	✗	✗	✗	<ul style="list-style-type: none"> Architekturpräsentation 	<ul style="list-style-type: none"> Nach Vereinbarung (ca. 2h)
6	✗	✓	✗	✓	✓	✗	✗		
7	✗	✓	✗	✗	✓	✗	✗	<ul style="list-style-type: none"> MS02 erreicht (Mehrere Spielobjekte bewegen, Interaktionen zwischen Spielobjekten, Screen-Management) 	sopranium.de
8	✗	✗	✗	✗	✓	✗	✗	<ul style="list-style-type: none"> Präsentation Programm (beta) Abgabe Programm (beta) 	<ul style="list-style-type: none"> Präsentation: 16.12. 14:00 - 18:00, Zoom, TBA Abgabe: 18.12. bis 23:59
9	✗	✗	✗	✗	✓	✓	✗		
10	Ferien (23.12.2021 - 06.01.2022)								
11									

Gruppeneinteilung

- ▶ Ziel: *Funktionierende* und *kompetente* Gruppen zusammenstellen
- ▶ Fragebogen
 - ▶ Namen und Emails für Dienste abfragen
 - ▶ Bis **heute Abend, 23:59**
- ▶ Wir teilen Gruppen ein
- ▶ Einteilung bis Sonntag

Hausaufgabe

- ▶ Ziel: *Werkzeuge* und *Vorgehen* kennenlernen und *Probleme* frühzeitig aufdecken
- ▶ Beschreibung auf dem Wiki
- ▶ Ungefähr:
 - ▶ Werkzeuge installieren und testen
 - ▶ Dienste testen
 - ▶ Git und Gitea kennenlernen
 - ▶ Texte zu Clean Code lesen
 - ▶ Ein MonoGame Programm schreiben
- ▶ Die Hausaufgabe ist der erste Sprint d.h. es können 5 Punkte erreicht werden

Meilenstein

Ein *Meilenstein* ist ein überprüfbares Ziel, dass zu einem Termin erreicht werden soll

- ▶ Meilensteine teilen den Projektverlauf in konkrete Schritte ein
 - ▶ Je eine *Etappen mit überprüfbaren Zwischenzielen*.
 - ▶ Erleichtert damit die *Projektplanung* und
 - ▶ Kontrolle des *Projektfortschritts*
- ▶ Entwicklung gegliedert in *5 Meilensteine*
 - ▶ Referenzablauf ermöglicht das Erkennen von Problemen
 - ▶ **Müssen** entsprechend des Spielkonzepts modifiziert werden
 - ▶ Regressionen sollten vermieden werden

Meilenstein

Ein *Meilenstein* ist ein überprüfbares Ziel, dass zu einem Termin erreicht werden soll

- ▶ Meilensteine teilen den Projektverlauf in konkrete Schritte ein
 - ▶ Je eine *Etappen mit überprüfbaren Zwischenzielen*.
 - ▶ Erleichtert damit die *Projektplanung* und
 - ▶ Kontrolle des *Projektfortschritts*
- ▶ Entwicklung gegliedert in *5 Meilensteine*
 - ▶ Referenzablauf ermöglicht das Erkennen von Problemen
 - ▶ **Müssen** entsprechend des Spielkonzepts modifiziert werden
 - ▶ Regressionen sollten vermieden werden

Meilenstein #1 (Woche 4)

- Spielobjekt in der Welt bewegbar
- Interaktive Kamera
- Karte laden/speichern
- Soundausgabe

Meilenstein

Ein *Meilenstein* ist ein überprüfbares Ziel, dass zu einem Termin erreicht werden soll

- ▶ Meilensteine teilen den Projektverlauf in konkrete Schritte ein
 - ▶ Je eine *Etappen mit überprüfbaren Zwischenzielen*.
 - ▶ Erleichtert damit die *Projektplanung* und
 - ▶ Kontrolle des *Projektfortschritts*
- ▶ Entwicklung gegliedert in *5 Meilensteine*
 - ▶ Referenzablauf ermöglicht das Erkennen von Problemen
 - ▶ **Müssen** entsprechend des Spielkonzepts modifiziert werden
 - ▶ Regressionen sollten vermieden werden

Meilenstein #1 (Woche 4)
- Spielobjekt in der Welt
bewegbar
- Interaktive Kamera
Karte laden/ansichten

Meilenstein #5 (Woche 15)
- Spiel ist fehlerfrei
- Spielmechanik ist ausbalanciert

Scrum als Vorgehensmodell

Scrum

- ▶ Ist ein *iteratives Vorgehensmodell*
- ▶ Gehört zu den *agilen Methoden*
- ▶ Entwicklung wird in *Sprints* aufgeteilt (1 bis 4 Wochen)
- ▶ Es existiert eine *auslieferbare* Software am Ende jedes Sprints
- ▶ Scrum definiert sich über *Rollen*, *Events* und *Artefakte*

Developer

- ▶ Aufgabe: Programmieren, Design, technische Lösung, Projektablauf
- ▶ Verantwortung: Produktqualität, Zeit

Product Owner

- ▶ Aufgabe: Kundengespräche, Vermarktung, Zielvorgabe, Anforderungserhebung
- ▶ Verantwortung: Produktmerkmale, Budget, Markterfolg

Scrum Master

- ▶ Aufgabe: Organisation, Mediation, Prozesskontrolle
- ▶ Verantwortung: Compliance, Prozess

Product Backlog

- ▶ Liste von *Anforderungen* mit abgeleiteten *User Stories*
- ▶ Nach *Wichtigkeit* für den Projekterfolg geordnet

User Stories

- ▶ *Kompakte Beschreibung* einer Funktion aus Nutzersicht
- ▶ Aufwandsabschätzung mit Story Points
- ▶ Als <Rolle> möchte ich <Funktion>, um <Nutzen>.
 - ▶ *Wer* oder *was* wird diese Funktion nutzen?
 - ▶ *Was* ist die Funktion?
 - ▶ *Welchen* Nutzen hat diese Funktion?
- ▶ Manchmal Akzeptanzkriterium (wie wird es getestet)

#47: *Speichern*

Als *Spieler* möchte ich *den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern*, um *später weiterzuspielen*.

Points:8

Sprint Backlog

- ▶ Liste an Aufgaben für den aktuellen Sprint
- ▶ Liste von *User Stories* mit abgeleiteten *Tasks*
- ▶ Wird für jeden Sprint neu erstellt

Task

- ▶ Teilaufgabe einer User Story
- ▶ Innerhalb eines Sprints erfüllbar
- ▶ Aufwandsabschätzung in Personenstunden

#47: *Speichern*

Als *Spieler* möchte ich *den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern*, um *später weiterzuspielen*.

Points:8

Sprint Backlog

- ▶ Liste an Aufgaben für den aktuellen Sprint
- ▶ Liste von *User Stories* mit abgeleiteten *Tasks*
- ▶ Wird für jeden Sprint neu erstellt

Task

- ▶ Teilaufgabe einer User Story
- ▶ Innerhalb eines Sprints erfüllbar
- ▶ Aufwandsabschätzung in Personenstunden

#47: *Speichern*

Als *Spieler* möchte ich *den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern*, um *später weiterzuspielen*.

Points:8

#47-1

Das Spiel legt
Savegame-Dateien im
entsprechenden
Verzeichnis an.

2h

Sprint Backlog

- ▶ Liste an Aufgaben für den aktuellen Sprint
- ▶ Liste von *User Stories* mit abgeleiteten *Tasks*
- ▶ Wird für jeden Sprint neu erstellt

Task

- ▶ Teilaufgabe einer User Story
- ▶ Innerhalb eines Sprints erfüllbar
- ▶ Aufwandsabschätzung in Personenstunden

#47: *Speichern*

Als *Spieler* möchte ich *den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern*, um *später weiterzuspielen*.

Points: 8

#47-1

Das Spiel le
Savegame-Dat
entsprechend
Verzeichnis

#47-2

Eigenschaften von
GameObject als JSON
serialisieren.

8h

Sprint Backlog

- ▶ Liste an Aufgaben für den aktuellen Sprint
- ▶ Liste von *User Stories* mit abgeleiteten *Tasks*
- ▶ Wird für jeden Sprint neu erstellt

Task

- ▶ Teilaufgabe einer User Story
- ▶ Innerhalb eines Sprints erfüllbar
- ▶ Aufwandsabschätzung in Personenstunden

#47: *Speichern*

Als *Spieler* möchte ich *den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern*, um *später weiterzuspielen*.

Points: 8

#47-1

Das Spiel le

#47-2

#47-3

Speichern auf
Windows und Linux
testen.

3h

enschaften von
Object als JSON
alisieren.

8h

Product Increment

- ▶ Neue Produktversion am Ende des Sprints
- ▶ *Direkt auslieferbar*

Definition of Done

- ▶ Bestimmt wann eine Aufgabe *fertig* ist
- ▶ Wird vom Team erstellt
- ▶ Darf sich im Laufe des Projekts ändern

Product Increment

- ▶ Neue Produktversion am Ende des Sprints
- ▶ *Direkt auslieferbar*

Definition of Done

- ▶ Bestimmt wann eine Aufgabe *fertig* ist
- ▶ Wird vom Team erstellt
- ▶ Darf sich im Laufe des Projekts ändern

Definition of Done Beispiele

- Team einverstanden (incl. PO)
- Auf *master* eingchecked
- Keine neuen Bugs
- API dokumentiert
- Tests erfolgreich
- Codestyle eingehalten
- Keine *//TODOs*

Sprint Planning

- ▶ Vor jedem Sprint
- ▶ Länge abhängig von Sprintlänge (ca. 2h je Woche)
- ▶ *Was* wird im nächsten Sprint getan?
 - ▶ *Product Owner* präsentiert die wichtigsten Items aus dem Product Backlog
- ▶ *Wie* wird es umgesetzt?
 - ▶ *Developer* wählen ihre Aufgaben beginnend beim wichtigsten Item
 - ▶ *Developer* zerlegen ausgewählte User Stories in Tasks
 - ▶ *Developer* erstellen neuen Softwareentwurf

Daily Scrum

- ▶ *Tägliches* Treffen
- ▶ Maximal 15 Minuten
- ▶ *Developer* beantworten nacheinander
 - ▶ Was habe ich seit dem letzten Treffen getan?
 - ▶ Was plane ich bis zum nächsten Treffen zu tun?
 - ▶ Welche Probleme hatte ich und wo benötige ich Hilfe?
- ▶ Fragen werden im Daily Scrum nicht beantwortet

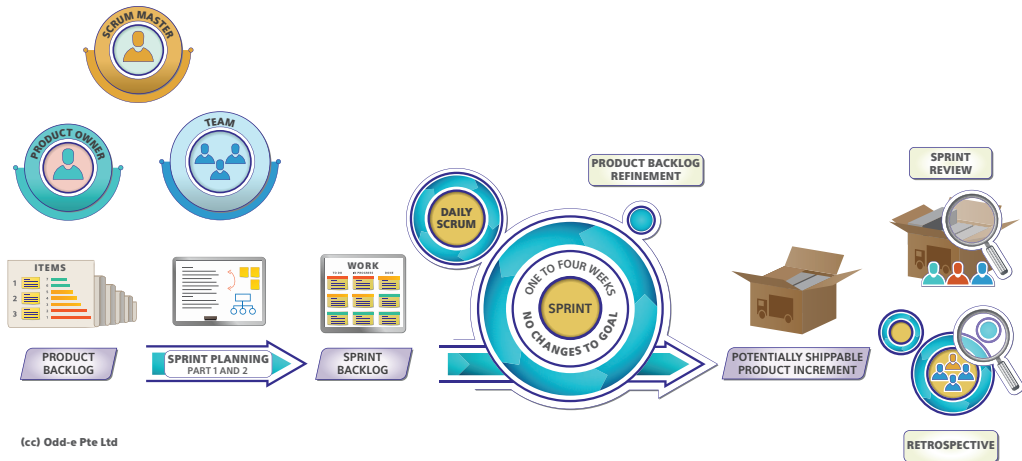
Sprint Review

- ▶ Treffen am Ende des Sprints
- ▶ Länge abhängig von Sprintlänge (ca. 1h je Woche)
- ▶ *Developer* präsentieren neuen Product Increment
- ▶ *Product Owner* bestimmt welche Tasks und User Stories fertig sind
- ▶ *Product Owner* erklärt, wie gut die Aufwandsabschätzung war

Sprint Retrospective

- ▶ Treffen des Teams *nach dem Sprint Review*
- ▶ Länge abhängig von Sprintlänge (ca. 45min je Woche)
- ▶ *Scrum Master* hilft, den Prozess zu verbessern
 - ▶ Wie lief es im letzten Sprint hinsichtlich Personen, Beziehungen, Prozessen, Werkzeugen?
 - ▶ Muss die *Definition of Done* angepasst werden?
 - ▶ Wie kann die Arbeitsweise des Teams verbessert werden?

SCRUM



Scrum im Sopra

- ▶ *Dozenten* sind Kunden
- ▶ *Tutoren* sind Scrum Master
- ▶ *Sie* sind Developer
- ▶ *Sie* können in Ihrer Gruppe Product Owner werden

- ▶ *Dozenten* sind Kunden
- ▶ *Tutoren* sind Scrum Master
- ▶ *Sie* sind Developer
- ▶ *Sie* können in Ihrer Gruppe Product Owner werden

Dozenten sind aber auch Dozenten, denen sie *Fragen stellen können* und die Ihnen helfen (z.B.: Bei Treffen, in Mattermost, oder als Mention in einem Gitea-Ticket)

- ▶ *Dozenten* sind Kunden
- ▶ *Tutoren* sind Scrum Master
- ▶ *Sie* sind Developer
- ▶ *Sie* können in Ihrer Gruppe Product Owner werden

Dozenten sind aber auch Dozenten, denen sie *Fragen stellen können* und die Ihnen helfen (z.B.: Bei Treffen, in Mattermost, oder als Mention in einem Gitea-Ticket)

Tutoren sind auch *Kundenversther* und *Berater*.

Product Backlog

Menge von *Items*

- ▶ Beschreibender Titel
- ▶ Ausführliche Beschreibung
- ▶ Labels nach Funktion
bug, question, ...
- ▶ Wichtigkeit
low, mid, high
- ▶ Zeitschätzung (Gesamtarbeitszeit)
est: 0h, 1h, 2h, 3h, 5h, ..., ∞

☐ #28 **Spielstand speichern** est: 5 high

vor 4 Minuten von [vincent](#) geöffnet

☐ #27 **Bogenschützen Schussfähigkeit** est: 5 mid

vor 4 Minuten von [vincent](#) geöffnet

☐ #26 **Texturen einfacher austauschbar machen** est: 3 low

vor 5 Minuten von [vincent](#) geöffnet

☐ #25 **Spilernamen mit äüöß crashen das Spiel** bug est: 2 high

vor 6 Minuten von [vincent](#) geöffnet

☐ #24 **Als Entwickler möchte ich Shader in GLS Importieren können für Cellshading** est: ∞ low user story

vor 6 Minuten von [vincent](#) geöffnet

☐ #23 **Einheiten zwischen zwei Punkten bewegen lassen** est: 8 high

vor 7 Minuten von [vincent](#) geöffnet

☐ #22 **Product Owner Tasks (Week 3)** est: 3 high

vor 8 Minuten von [vincent](#) geöffnet

Product Backlog

Menge von *Items*

- ▶ Beschreibender Titel
- ▶ Ausführliche Beschreibung
- ▶ Labels nach Funktion
bug, question, ...
- ▶ Wichtigkeit
low, mid, high
- ▶ Zeitschätzung (Gesamtarbeitszeit)
est: 0h, 1h, 2h, 3h, 5h, ..., ∞

☐ #28 **Spielstand speichern** est: 5 high

vor 4 Minuten von [vincent](#) geöffnet

☐ #27 **Bogenschützen Schussfähigkeit** est: 5 mid

vor 4 Minuten von [vincent](#) geöffnet

☐ #26 **Texturen einfacher austauschbar machen** est: 3 low

vor 5 Minuten von [vincent](#) geöffnet

☐ #25 **Spilernamen mit äüöß crashen das Spiel** bug est: 2 high

vor 6 Minuten von [vincent](#) geöffnet

☐ #24 **Als Entwickler möchte ich Shader in GLS Importieren können für Cellshading** est: ∞ low user story

vor 6 Minuten von [vincent](#) geöffnet

☐ #23 **Einheiten zwischen zwei Punkten bewegen lassen** est: 8 high

vor 7 Minuten von [vincent](#) geöffnet

☐ #22 **Product Owner Tasks**

vor 8 Minuten von [vincent](#) geöffnet

Annahme: Das *Game Design Document* fungiert als Sammlung von *User Stories*, aus der geeignete *Items* abgeleitet werden können.

Sprint Backlog

- ▶ Liste der im Sprint zu erledigenden *Items*
- ▶ Jedes *Items* ist dem *Sprint* zugeordnet (ein Meilenstein in Gitea)
- ▶ Jedem *Items* ist ein *Developer* zugeordnet

🚩 Sprint 4

📅 2020-11-19 ④ 4 offen ① 1 geschlossen ✎ Bearbeiten ✕ Schließen 🗑 Löschen

Ziel diesen Sprint: Minimal spielbares Spiel (Objekte und Kamera bewegen sich, Zlatko der Zwerg kann rumlaufen)

Sprint 4

Meilenstein bearbeiten

Neues Issue

📅 2020-11-19 20% abgeschlossen

④ 4 offen

① 1 geschlossen

Label ▾

Zuständig ▾

Typ ▾

Sortieren ▾

☐ #28 Spielstand speichern est: 5 high
vor 33 Minuten von [vincent](#) geöffnet

☐ #27 Bogenschützen Schussfähigkeit est: 5 mid
vor 34 Minuten von [vincent](#) geöffnet

☐ #26 Texturen einfacher austauschbar machen est: 3 low
vor 34 Minuten von [vincent](#) geöffnet

☐ #25 Spielernamen mit äüöß crashen das Spiel bug est: 2 help wanted high
vor 35 Minuten von [vincent](#) geöffnet



Gruppentreffen

Zweistündiges Treffen mit dem *Tutor* als Moderator und *Scrum Master*

- | | |
|-------------------------|-------------|
| 1. Sprint Review | (ca. 30min) |
| 2. Sprint Retrospective | (ca. 15min) |
| 3. Sprint Planning | (ca. 45min) |

Gruppentreffen

Zweistündiges Treffen mit dem *Tutor* als Moderator und *Scrum Master*

- | | |
|-------------------------|-------------|
| 1. Sprint Review | (ca. 30min) |
| 2. Sprint Retrospective | (ca. 15min) |
| 3. Sprint Planning | (ca. 45min) |

Sprint Review

- ▶ Developer führen *Product Increment* vor
- ▶ Gruppe entscheidet was gemäß *DoD* (nicht) erledigt ist
 - ▶ Nicht erledigte Items zurück ins *Product Backlog* verschieben
- ▶ Wie gut waren die Zeitschätzungen?
- ▶ Ist der nächste Meilenstein erreicht?

Gruppentreffen

Zweistündiges Treffen mit dem *Tutor* als Moderator und *Scrum Master*

1. Sprint Review (ca. 30min)
2. Sprint Retrospective (ca. 15min)
3. Sprint Planning

Sprint Review

- ▶ Developer führen *Product Increment* vor
- ▶ Gruppe entscheidet was gemäß *DoD* (nicht) erledigt
 - ▶ Nicht erledigte Items zurück ins *Product Backlog* ver
- ▶ Wie gut waren die Zeitschätzungen?
- ▶ Ist der nächste Meilenstein erreicht?

Sopra DoD

- Das Item ist in Gitea *geschlossen*.
- Im Item sind die geschätzte und die tatsächliche *Arbeitszeit* eingetragen.
- Alle für das Item relevanten Dateien sind im aktuellen Stand des remote *release Branch* integriert.
- Der Tutor hat die Fertigstellung des Items im Sprint Review anhand des aktuellen Standes des remote release Branch bestätigt.

Gruppentreffen

Zweistündiges Treffen mit dem *Tutor* als Moderator und *Scrum Master*

- | | |
|-------------------------|-------------|
| 1. Sprint Review | (ca. 30min) |
| 2. Sprint Retrospective | (ca. 15min) |
| 3. Sprint Planning | (ca. 45min) |

Sprint Retrospective

- ▶ Was lief in diesem Sprint *gut* oder *schlecht*?
 - ▶ Probleme mit oder Lob für *Gruppenmitglieder*, *Prozess*, *Zeiteinteilung* oder *Tools*
- ▶ Bei Bedarf *DoD* anpassen
- ▶ Entschlüsse schriftlich festhalten (am Besten in `readme.md`)

Gruppentreffen

Zweistündiges Treffen mit dem *Tutor* als Moderator und *Scrum Master*

- | | |
|-------------------------|-------------|
| 1. Sprint Review | (ca. 30min) |
| 2. Sprint Retrospective | (ca. 15min) |
| 3. Sprint Planning | (ca. 45min) |

Sprint Planning

- ▶ Verfügbare Arbeitszeit für diesen Sprint festlegen
- ▶ Product Owner stellt vor wie das nächste Increment aussehen soll (was soll am Ende funktionieren)
- ▶ Product Backlog angefangen bei der *wichtigsten* Items durchgehen:
 - ▶ Gemeinsam konkrete *Funktion* besprechen und *Aufwand* schätzen
 - ▶ Ins Sprint Backlog verschieben
 - ▶ Wiederholen bis verfügbare Arbeitszeit aufgebraucht
- ▶ Items im Sprint Backlog an die Gruppenmitglieder verteilen

Items werden aus dem *GDD* abgeleitet

- ▶ GDD nimmt für das Sopra die Rolle von *User Stories* ein
- ▶ Kapitel: Zusammenfassung, Spiellogik (Objekte, Aktionen, Ablauf, ...)

1.1 Zentrale Spielmechanik

Der Spieler kämpft sich durch einen Irrgarten aus zufällig generierten Räumen mit über die Zeit stärker werdenden Fallen, Gegnern und Schätzen. Eine der Spielfigur folgende Todeswand (die Dunkelheit) setzt den Spieler dabei unter Druck.

Items werden aus dem *GDD* abgeleitet

- ▶ GDD nimmt für das Sopra die Rolle von *User Stories* ein
- ▶ Kapitel: Zusammenfassung, Spiellogik (Objekte, Aktionen, Ablauf, ...)

#23 **Objekt Dunkelheit**

Asset für die Dunkelheit (unendlich breite Wand mit Schatteneffekten) erstellen und im Code verfügbar machen.

?

1.1 Zentrale Spielmechanik

Der Spieler kämpft sich durch einen Irrgarten aus zufällig generierten Räumen mit über die Zeit stärker werdenden Fallen, Gegnern und Schätzen. Eine der Spielfigur folgende Todeswand (die Dunkelheit) setzt den Spieler dabei unter Druck.

Items werden aus dem *GDD* abgeleitet

- ▶ GDD nimmt für das Sopra die Rolle von *User Stories* ein
- ▶ Kapitel: Zusammenfassung, Spiellogik (Objekte, Aktionen, Ablauf, ...)

#23 **Objekt Dunkelheit**

Asset für die Dunkelheit (unendlich breite Wand mit Schatteneffekten) erstellen und im Code verfügbar machen.

?

#64 **Verhalten Dunkelheit**

Verhaltenskomponente für die Dunkelheit: verfolgt den Spieler mit zunehmender Geschwindigkeit.

?

1.1 Zentrale Spielmechanik

Der Spieler kämpft sich durch einen Irrgarten aus zufällig generierten Räumen mit über die Zeit stärker werdenden Fallen, Gegnern und Schätzen. Eine der Spielfigur folgende Todeswand (die Dunkelheit) setzt den Spieler dabei unter Druck.

Items werden aus dem *GDD* abgeleitet

- ▶ GDD nimmt für das Sopra die Rolle von *User Stories* ein
- ▶ Kapitel: Zusammenfassung, Spiellogik (Objekte, Aktionen, Ablauf, ...)

#23 **Objekt Dunkelheit**

Asset für die Dunkelheit (unendlich breite Wand mit Schatteneffekten) erstellen und im Code verfügbar machen.

?

#64 **Verhalten Dunkelheit**

Verhaltenskomponente für die Dunkelheit: verfolgt den Spieler mit zunehmender Geschwindigkeit.

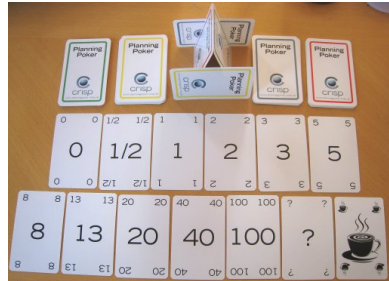
?

1.1 Zentrale Spielmechanik

Der Spieler kämpft sich durch einen Irrgarten aus zufällig generierten Räumen mit über die Zeit stärker werdenden Fallen, Gegnern und Schätzen. Eine der Spielfigur folgende Todeswand (die Dunkelheit) setzt den Spieler dabei unter Druck.

Z.B.: mit *Planning Poker*

1. Item besprechen
(jeder soll verstehen was zu tun ist)
2. *Verdeckt* wählt jeder einen Wert aus
(Story Points/Stunden)
3. Gleichzeitig aufdecken
4. Wer den höchsten bzw. niedrigsten Wert aufdeckt, erklärt *warum*
5. Wiederholen bis alle den gleichen Wert zeigen



Z.B.: mit *Planning Poker*

1. Item besprechen
(jeder soll verstehen was zu tun ist)
2. *Verdeckt* wählt jeder einen Wert aus
(Story Points/Stunden)
3. Gleichzeitig aufdecken
4. Wer den höchsten bzw. niedrigsten Wert aufdeckt, erklärt *warum*
5. Wiederholen bis alle den gleichen Wert zeigen



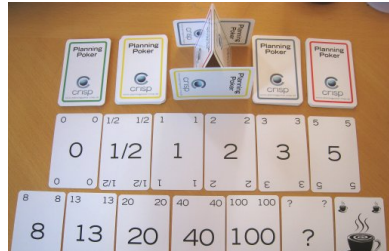
#64 Verhalten Dunkelheit

Verhaltenskomponente für die Dunkelheit: verfolgt den Spieler mit zunehmender Geschwindigkeit.

?

Z.B.: mit *Planning Poker*

1. Item besprechen
(jeder soll verstehen was zu tun ist)
2. *Verdeckt* wählt jeder einen Wert aus
(Story Points/Stunden)
3. Gleichzeitig aufdecken
4. Wer den höchsten bzw. niedrigsten Wert aufdeckt, erklärt *warum*
5. Wiederholen bis alle den gleichen Wert zeigen



#64 Verhalten Dunkelheit

Verhaltenskomponente für die Dunkelheit: verfolgt den Spieler mit zunehmender Geschwindigkeit. So, als stünde sie auf einem Anhänger ... mit einem langen Gummiband hinterher gezogen.

5h

Product Owner (ab Woche 2)

- ▶ Pflege des Product Backlog
 - ▶ Items an Entwicklungsstand *anpassen*, und nach *Wichtigkeit ordnen*
 - ▶ Weitere Items aus GDD extrahieren
- ▶ Gruppentreffen vorbereiten:
 - ▶ Was ist nach *DoD* fertig?
 - ▶ Wie waren die Aufwandsabschätzungen?
 - ▶ Product Increment vorbereiten

Architektur (ab Woche 3)

- ▶ Pflege der Architektur
- ▶ Schnittstellen und Grenzen definieren und deren Einhaltung sicherstellen

Qualitätssicherung (ab Woche 3)

- ▶ Code auf Clean Code Richtlinien prüfen
 - ▶ Code Reviews machen und Ergebnisse mit der Gruppe mitteilen (Ticket)
 - ▶ ReSharper Konformität herstellen
-
- ▶ Alle Recurring Tasks *müssen* verteilt werden
 - ▶ Ticket mit Zeitschätzung (max. 2h) je Aufgabe und Sprint

Was nun?

- ▶ *Anforderungen*

Bekannte Spiele? Einschränkungen? Naheliegende Mechaniken? Begriffe unklar?

- ▶ *Bekanntes verändern und kombinieren*

Mechanisch? Setting und Handlung?

- ▶ *Sie sind nicht allein!*

- ▶ Ihre Idee wird sich verändern, oder gar verworfen werden
- ▶ Was ist Ihnen wichtig? Was wollen Sie gar nicht?
- ▶ Fragen Sie nach

- ▶ Erzählen sie sich gegenseitig einen *exemplarischen Spielablauf*
 - ▶ Von Anfang an (Drücke auf *game.exe*)
 - ▶ Bis Gewonnen und Verloren
 - ▶ Was passiert wenn?
- ▶ Welche Spielobjekte und Aktionen gibt es?
- ▶ Unklare Dinge klären (Domänenwissen und Spielidee) und Lücken im Ablauf Spielablauf füllen

- ▶ Erzählen sie sich gegenseitig einen *exemplarischen Spielablauf*
 - ▶ Von Anfang an (Drücke auf *game.exe*)
 - ▶ Bis Gewonnen und Verloren
 - ▶ Was passiert wenn?
- ▶ Welche Spielobjekte und Aktionen gibt es?
- ▶ Unklare Dinge klären (Domänenwissen und Spielidee) und Lücken im Ablauf Spielablauf füllen

Der Spieler kämpft als Krieger in einem Verlies.

- ▶ Erzählen sie sich gegenseitig einen *exemplarischen Spielablauf*
 - ▶ Von Anfang an (Drücke auf *game.exe*)
 - ▶ Bis Gewonnen und Verloren
 - ▶ Was passiert wenn?
- ▶ Welche Spielobjekte und Aktionen gibt es?
- ▶ Unklare Dinge klären (Domänenwissen und Spielidee) und Lücken im Ablauf Spielablauf füllen

Der Spieler kämpft als Krieger in einem Verlies.

- Objekte: Verlies und Krieger
- Aktion: Kämpfen
- Also hat er ein Schwert? Gibt es Zauber? Was ist ein Verlies? Gibt es nur Krieger?

- ▶ Erzählen sie sich gegenseitig einen *exemplarischen Spielablauf*
 - ▶ Von Anfang an (Drücke auf *game.exe*)
 - ▶ Bis Gewonnen und Verloren
 - ▶ Was passiert wenn?
- ▶ Welche Spielobjekte und Aktionen gibt es?
- ▶ Unklare Dinge klären (Domänenwissen und Spielidee) und Lücken im Ablauf Spielablauf füllen

Der Spieler kämpft als Krieger in einem Verlies.

- Objekte: Verlies und Krieger
- Aktion: Kämpfen
- Also hat er ein Schwert? Gibt es Zauber? Was ist ein Verlies? Gibt es nur Krieger?

Die Dunkelheit verfolgt den Spieler und setzt ihn unter Druck.

- ▶ Erzählen sie sich gegenseitig einen *exemplarischen Spielablauf*
 - ▶ Von Anfang an (Drücke auf *game.exe*)
 - ▶ Bis Gewonnen und Verloren
 - ▶ Was passiert wenn?
- ▶ Welche Spielobjekte und Aktionen gibt es?
- ▶ Unklare Dinge klären (Domänenwissen und Spielidee) und Lücken im Ablauf Spielablauf füllen

Der Spieler kämpft als Krieger in einem Verlies.

- Objekte: Verlies und Krieger
- Aktion: Kämpfen
- Also hat er ein Schwert? Gibt es Zauber? Was ist ein Verlies? Gibt es nur Krieger?

Die Dunkelheit verfolgt den Spieler und setzt ihn unter Druck.

- Objekt: Dunkelheit (?)
- Woher weiß der Spieler wie weit die Dunkelheit entfernt ist?
- Müssen dann alle Gänge in die gleiche Richtung gehen?

- ▶ Erzählen sie sich gegenseitig einen *exemplarischen Spielablauf*
 - ▶ Von Anfang an (Drücke auf *game.exe*)
 - ▶ Bis Gewonnen und Verloren
 - ▶ Was passiert wenn?
- ▶ Welche Spielobjekte und Aktionen gibt es?
- ▶ Unklare Dinge klären (Domänenwissen und Spielidee) und Lücken im Ablauf Spielablauf füllen

Das beta-GDD ist bereits eine *vollständige* Beschreibung des Spiels

Der Spieler kämpft als Krieger in einem Verlies.

- Objekte: Verlies und Krieger
- Aktion: Kämpfen
- Also hat er ein Schwert? Gibt es Zauber? Was ist ein Verlies? Gibt es nur Krieger?

Die Dunkelheit verfolgt den Spieler und setzt ihn unter Druck.

- Objekt: Dunkelheit (?)
- Woher weiß der Spieler wie weit die Dunkelheit entfernt ist?
- Müssen dann alle Gänge in die gleiche Richtung gehen?

Ab sofort

- ▶ Fragebogen bis spätestens *heute Abend 23:59* ausfüllen.
- ▶ Auf Gruppeneinteilung warten (bis Sonntag)
- ▶ Mit der Hausaufgabe anfangen

Danach

- ▶ Hausaufgaben machen
- ▶ Termin für Gruppentreffen ausmachen (Mo-Do)
- ▶ Spielidee entwickeln

Ab sofort

- ▶ Fragebogen bis spätestens *heute Abend 23:59* ausfüllen.
- ▶ Auf Gruppeneinteilung warten (bis Sonntag)
- ▶ Mit der Hausaufgabe anfangen

Danach

- ▶ Hausaufgaben machen
- ▶ Termin für Gruppentreffen ausmachen (Mo-Do)
- ▶ Spielidee entwickeln

Überlegen Sie *jetzt* ob Sie:

- ▶ dieses Semester genug Zeit haben und zu allen Pflichtterminen anwesend sein können
- ▶ Zugriff auf einen zum Entwickeln geeigneten Rechner haben

Ab sofort

- ▶ Fragebogen bis spätestens *heute Abend 23:59* ausfüllen.
- ▶ Auf Gruppeneinteilung warten (bis Sonntag)
- ▶ Mit der Hausaufgabe anfangen

Danach

- ▶ Hausaufgaben machen
- ▶ Termin für Gruppentreffen ausmachen (Mo-Do)
- ▶ Spielidee entwickeln

Überlegen Sie *jetzt* ob Sie:

- ▶ dieses Semester genug Zeit haben und zu allen Pflichtterminen anwesend sein können
- ▶ Zugriff auf einen zum Entwickeln geeigneten Rechner haben

Stellen Sie sicher, dass Ihr *Postfach* nicht voll ist und Emails von der Uni ankommen (Spamfilter)!

