Softwarepraktikum

Nico Hauff, Vincent Langenfeld, Frank Schüssele

October 19, 2023

Kontext

[Büro von Mitarbeiter S., Donnerstag 1400, Wochenendstimmung]

Chef: S., wir müssen ein Computerspiel für die WiKe AG (Wichtiger Kunde AG) entwickeln!

S.: Ein Computerspiel? Aber das haben wir noch nie gemacht!

Chef: Stimmt, aber Sie und ihr Team bekommen das hin. Man soll sich ja nie etwas Neuem gegenüber verschließen.

Kontext

```
[Büro von Mitarbeiter S., Donnerstag 1400, Wochenendstimmung]

Chef:S., wir müssen ein Computerspiel für die WiKe AG
(Wichtiger Kunde AG) entwickeln!

S.:Ein Computerspiel?  Aber das haben wir noch nie gemacht!

Chef:Stimmt, aber Sie und ihr Team bekommen das hin.
Man soll sich ja nie etwas Neuem gegenüber verschließen.
```

Im Softwarepraktikum ...

- sind Sie Mitarbeiter S.
- sind die Dozenten die Vertreter der WiKe AG (Wichtiger Kunde AG)
- simulieren wir ein Softwareentwicklungsprojekt

Vorgehen

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

Vorgehen

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

- die Domäne analysieren und verstehen

Vorgehen

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

- die Domäne analysieren und verstehen
- um die Anforderungen zu erheben zu und verstehen

Vorgehen

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

- die Domäne analysieren und verstehen
- um die Anforderungen zu erheben zu und verstehen
- um die Umsetzung zu konzipieren

Vorgehen

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

- die Domäne analysieren und verstehen
- um die Anforderungen zu erheben zu und verstehen
- um die Umsetzung zu konzipieren
- um das benötigte Produkt zu implementieren.

Agenda

- Organisatorisches
- Vorgehen
- Gesamtablauf
- Vorgehensmodell: Scrum
- Scrum im Sopra
- Die ersten Schritte

Organisatorisches

Lernziele

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

Lernziele

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

```
Wir setzen voraus,
dass Sie bereits
programmieren
können!
```

Team

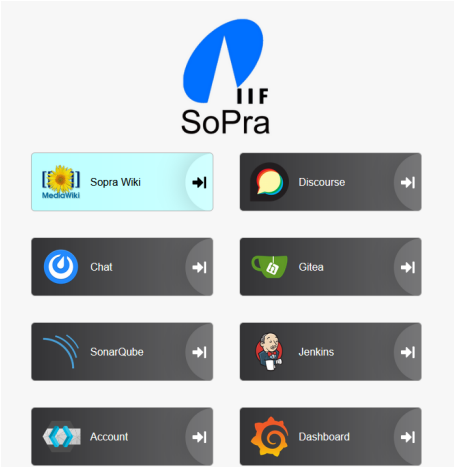
- **Tutoren**
  Thierry Meiers, Tobias Kolzer, Gerrit Freiwald, Luis Schaffner, Felix Leimenstoll, Zacharias Häringer, Dejan Dudukovic, Tobias Vonier
- **Dozenten**
  Frank Schüssele, Nico Hauff, Vincent Langenfeld
- **Infrastruktur**
  Daniel Dietsch
- **Verantwortlich**
  Prof. Dr. A. Podelski

Organisation

- Gruppen mit 5-7 Studenten
- 6 ECTS (180h) über 17 Wochen
  - 14 Arbeitsabschnitte (Sprints), je eine Woche
- 4 Vorlesungen
- Termine
  - 3 Abgaben
  - 3 Präsentationen vor dem Kurses (Do. 14-18 Uhr)
  - 1 Besprechung mit den Dozenten
  - Wöchentliches Gruppentreffen (Mo-Do)

Dienste und Werkzeuge

- Serviceübersicht
- Fragen und Informationen
  Gruppenmitglieder, Tutoren, Wiki, Discourse, Sprechstunde und Mattermost
- Primärdienste
  Gitea, Git, Jenkins, Sonar, Dashboard, Mailinglisten (`sopra-crew@...`, `sopraXX@...`)
- Sekundärdienste
  Mattermost, Discourse
- Werkzeuge
  C#, .NET 6, Monogame 3.8, Visual Studio Community, Resharper



services.sopranium.de

Zulassung

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

Zulassung

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:

$$\frac{(180h - 48)/14}{\textbf{9,43h}}$$

Zulassung

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:`

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

`Arbeitszeit:`

$$\frac{(180h - 48)/14}{\textbf{9,43h}}$$

Zulassung

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:`

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

`Arbeitszeit:`

$$\frac{(180h - 48)/14}{\textbf{9,43h}}$$

Benotung

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 sinnvoll erledigt?
  - Note ergibt sich aus Punkten

Vorgehen

Domäne

- Die Domäne analysieren und verstehen
- um die Anforderungen zu erheben und verstehen
- um die Umsetzung zu konzipieren
- um das benötigte Produkt zu implementieren.

Domäne

Unter einer (Problem)-Domä es Problemfeld oder einen bestimmten Einsatzbe

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, d.h. welche Objekte gibt es in der Domäne?
- Funktionen, d.h. welche kontinuierlich messabren Größen gibt es in der Domäne?
- Ereignisse, d.h. welche Ereignisse treten in der Domäne auf?
- Interaktionen, d.h. wie können Elemente der Domäne interagieren?

Domäne

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

- Entität, d.h. welche Objekte gibt es in der Domäne?
- Funktionen, d.h. welche kontinuierlich messabren Größen gibt es in der Domäne?
- Ereignisse, d.h. welche Ereignisse treten in der Domäne auf?
- Interaktionen, d.h. wie können Elemente der Domäne interagieren?
- Aus Sicht verschiedener Stakeholder
  - Stakeholder sind in irgendeiner Weise von der Domäne betroffen
  - z.B.: Auftraggeber, Programmierer/Spieledesigner, Freiwillige Selbstkontrolle (FSK), Hardwarehersteller

Domäne · Beispiel: Brettspiel

(*Monopoly Barcelona (40222232563)* by *Jorge Franganillo* licensed under CC BY 2.0)

- Entität
- Funktionen
- Ereignisse
- Interaktionen

Domäne · Beispiel: Brettspiel

(*Monopoly Barcelona (40222232563)* by *Jorge Franganillo* licensed under CC BY 2.0)

- Entität
  Spielfigur, Würfel, Spielbrett, Spieler, Verlag,...
- Funktionen
- Ereignisse
- Interaktionen

Domäne · Beispiel: Brettspiel

(*Monopoly Barcelona (40222232563)* by *Jorge Franganillo* licensed under CC BY 2.0)

- Entität
  Spielfigur, Würfel, Spielbrett, Spieler, Verlag,...
- Funktionen
  Punktestand, Position, Spieldauer, Spielermotivation, (Flow), ...
- Ereignisse
- Interaktionen

Domäne · Beispiel: Brettspiel

(*Monopoly Barcelona (40222232563)* by *Jorge Franganillo* licensed under CC BY 2.0)

- Entität
  Spielfigur, Würfel, Spielbrett, Spieler, Verlag,...
- Funktionen
  Punktestand, Position, Spieldauer, Spielermotivation, (Flow), ...
- Ereignisse
  Spielstart, Feld erreicht, Karte ziehen, ...
- Interaktionen

Domäne · Beispiel: Brettspiel

(*Monopoly Barcelona (40222232563)* by *Jorge Franganillo* licensed under CC BY 2.0)

- Entität
  Spielfigur, Würfel, Spielbrett, Spieler, Verlag,...
- Funktionen
  Punktestand, Position, Spieldauer, Spielermotivation, (Flow), ...
- Ereignisse
  Spielstart, Feld erreicht, Karte ziehen, ...
- Interaktionen
  Spieler verrückt Spielfigur, Geld bezahlen/empfangen, Karte aufdecken, Tisch umstoßen, ...

Domäne · Beispiel: Brettspiel

(*Monopoly Barcelona (40222232563)* by *Jorge Franganillo* licensed under CC BY 2.0)

- Entität
  Spielfigur, Würfel, Spielbrett, Spieler, Verlag,...
- Funktionen
  Punktestand, Position, Spieldauer, Spielermotivation, (Flow), ...
- Ereignisse
  Spielstart, Feld erreicht, Karte ziehen, ...
- Interaktionen
  Spieler verrückt Spielfigur, Geld bezahlen/empfangen, Karte aufdecken, Tisch umstoßen, ...

Was ist mit dem Drumherum?

Vier grundlegende Elemente eines Spiels[1]

- Ästhetik
- Handlung
- Mechanik
- Technologie

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

Vier grundlegende Elemente eines Spiels[1]

- Ästhetik
- Handlung
- Mechanik
- Technologie

- Jedes Element lässt sich weiter zerlegen
  - Ästhetik z.B.: Aussehen, Stimmung, Metaphern
  - Handlung z.B.: Akte, Spannungskurve, Ereignisse in der Handlung
  - Mechanik z.B.: Spielobjekte und Interaktion (Entitäten, Funktionen, Ereignisse, Interaktionen)
  - Technologie z.B: Engine, Bibliotheken, Computergrafik

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

Vier grundlegende Elemente eines Spiels[1]

- Ästhetik
- Handlung
- Mechanik
- Technologie

- Jedes Element lässt sich weiter zerlegen
  - Ästhetik z.B.: Aussehen, Stimmung, Metaphern
  - Handlung z.B.: Akte, Spannungskurve, Ereignisse in der Handlung
  - Mechanik z.B.: Spielobjekte und Interaktion (Entitäten, Funktionen, Ereignisse, Interaktionen)
  - Technologie z.B: Engine, Bibliotheken, Computergrafik
- Das Spiel sollte aus Sicht jedes dieser Elemente betrachtet werden.

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

Anforderungen

- Die Domäne analysieren und verstehen
- um die Anforderungen zu erheben und zu verstehen
- um die Umsetzung zu konzipieren
- um das benötigte Produkt zu implementieren.

Anforderungen[2]

Anforderungen ist eine Aussage darüber, was man von einem (Software) System als Eigenschaften erwartet.

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

Anforderungen[2]

Anforderungen ist eine Aussage darüber, was man von einem (Software)-System als Eigenschaften erwartet.

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

Anforderungen[2]

Anforderungen ist eine Aussage darüber, was man von einem (Software)-System 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.
- Rahmenbedinungen legen organisatorische und/oder technische Restriktionen für das Softwaresystem und/oder den Entwicklungsprozess fest.

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

Anforderungen

```
Chef: Die WiKe AG hat die folgenden
Anforderungen geschickt.
Das Spiel soll ...
```

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

Anforderungen

```
Chef: Die WiKe AG hat die folgenden
Anforderungen geschickt.
Das Spiel soll ...
```

- 2D oder 3D Grafik (kein ASCII)
- Soundeffekte und Musik
- Mindestens 2 Spieler, min. einer menschlich
- Echtzeit
- 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

Anforderungen

```
Chef: Die WiKe AG hat die folgenden
Anforderungen geschickt.
Das Spiel soll ...
```

- 2D oder 3D Grafik (kein ASCII)
- Soundeffekte und Musik
- Mindestens 2 Spieler, min. einer menschlich
- Echtzeit
- 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.

Anforderungen

```
Chef: Die WiKe AG hat die folgenden
Anforderungen geschickt.
Das Spiel soll ...
```

- 2D oder 3D Grafik (kein ASCII)
- Soundeffekte und Musik
- Mindestens 2 Spieler, min. einer menschlich
- Echtzeit
- 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”).

Anforderungen

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

```
Chef: Finden Sie heraus, was das alles bedeuten soll.
```

Rahmenbedienungen

- C# und/oder F# mit .NET 6
- MonoGame 3.8
- Lauffähig auf Windows 10 (x64) und Linux
- Visual Studio Community
- Keine Warnings oder Errors vom Compiler (wöchentlich), keine Buildfehler.

Anforderungen · Beispiele

Anforderungen · Gegenbeispiele

Anforderungen

- Die Domäne analysieren und verstehen
- um die Anforderungen zu erheben und zu verstehen
- um die Umsetzung zu konzipieren
- um das benötigte Produkt zu implementieren.

Game Design Documen

Beschreibung der wesentlich

Anforderungen

Game Design Document (GDD)

Beschreibung der wesentlichen Merkmale des Spiels.

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
  Ästhetik, Handlung, Mechanik und Technologie
- Details: GDD-Vorlesung, Wiki

Ablauf

Ablauf

Woche	Organisation	Entwurf	MS 01	MS 02	MS 03	MS 04	MS 05	Was?	Wann und Wo?
0	✓	✗	✗	✗	✗	✗	✗	• Vorlesung "Organisation und Prozess" (Einführungsveranstaltung) • Gruppeneinteilung abwarten	• Einführungsveranstaltung: 19.10., 14:00 - 16:00, Geb. 82, HS 00-006 • Fragebogen: 19.10. bis 23:59 • Gruppen ab 22.10. in Gitea
1	✓	✓	✗	✗	✗	✗	✗	• Vorlesung "Game Design Document (GDD)" • Abgabe Hausaufgabe	• Vorlesung: 26.10., 14:00 - 16:00, Geb. 82, HS 00-006 • Anschließend Fragestunde, Computerpool • Abgabe: 28.10 bis 23:59
2	✗	✓	✓	✗	✗	✗	✗	• Vorlesung "Grundlagen Softwarearchitektur" • Wiederkehrende Aufgaben: Product Owner • Abgabe GDD (beta)	• Vorlesung: 02.11., **13:00 - 15:00**, Geb. 82, HS 00-006 • Anschließend Fragestunde, Computerpool • Abgabe: 04.11. bis 23:59
3	✗	✓	✓	✗	✗	✗	✗	• Vorlesung "Architektur von Videospielen" • Wiederkehrende Aufgaben: Architektur und Coaching • Wiederkehrende Aufgaben: Qualitätssicherung und Clean-Code	• Vorlesung: 9.11., 14:00 - 16:00, Geb. 82, HS 00-006 • Anschließend Fragestunde, Computerpool
4	✗	✓	✓	✓	✗	✗	✗	• MS01 erreicht (Spielobjekt in der Welt bewegbar, bewegliche Ansichten, Level laden/speichern, Soundausgabe) • Präsentation der Spielidee	• Präsentation: 16.11. 14:00 - 18:00, Geb. 82, HS 00-006 • Anschließend Fragestunde
5	✗	✓	✗	✓	✗	✗	✗	• Architekturpräsentation	• Architekturpräsentation nach Vereinbarung (ca. 2h) • Doodles: TBA
6	✗	✓	✗	✓	✓	✗	✗		
7	✗	✓	✗	✗	✓	✗	✗	• MS02 erreicht (Mehrere Spielobjekte bewegen, Interaktionen zwischen Spielobjekten, Screen-Management, Menü, HUD, Musik)	
8	✗	✗	✗	✗	✓	✗	✗	• Präsentation Programm (beta) • Abgabe Programm (beta)	• Präsentation: 14.12. 14:00 - 16:00, Geb. 82, HS 00-006 • Abgabe: 16.12. bis 23:59
9	✗	✗	✗	✗	✓	✓	✗		
10	**Ferien (23.12.2023- 06.01.2024)**								
11									

sopranium.de

Ablauf · Gruppeneinteilung

- Ziel: Funktionierende und kompetente Gruppen zusammenstellen
- Fragebogen
  - Namen und Emails für Dienste abfragen
  - Bis heute Abend, 23:59
- Wir teilen Sie in Gruppen ein
- Einteilung vor Sonntag
- Sonntag Terminumfrage für Gruppentreffen beantworten

Ablauf · 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

Ablauf · 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


```
Chef: Aber was ist eigentlich ein
"gutes Produkt"?
```

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

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
- Vermeiden Sie Regressionen

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 Proble
  - Müssen entsprechend des Spielkonzepts modifiziert
- Vermeiden Sie Regressionen

Meilenstein #1 (Woche 4)
- Spielobjekt in der Welt bewegbar
- Interaktive Kamera
- Ein Level laden
- 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 Proble
  - Müssen entsprechend des Spielkonzepts modifiziert
- Vermeiden Sie Regressionen

Meilenstein #1 (Woche 4)
- Spielobjekt in der Welt bewegbar
- Interaktive Kamera

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

Scrum als Vorgehensmodell

Scrum

- Gehört zu den agilen Methoden
- Entwicklung in Scrum ist iterativ und inkrementell

- 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

Scrum · Rollen im Team

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

Scrum · Artefakte

Product Backlog

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

```
#2: Anforderung:
Das Spiel soll sich an
Usabilitygrundsätze
halten.
```

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
  - Was kann man (neues) tun wenn es fertig ist?

```
#47: Speichern
Als Spieler möchte ich den
aktuellen Spielstand zu
einem beliebigen Zeitpunkt
abspeichern, um später
weiterzuspielen zu können.
                    Points:8
```

Sprint Backlog

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

#47: **Speichern**
Als Spieler möchte ich den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern, um später weiterzuspielen.
Points:8

Task

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

Sprint Backlog

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

#47: **Speichern**
Als Spieler möchte ich den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern, um später weiterzuspielen.
Points:8

Task

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

#47-1 **Savegame Ordner**
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

#47: **Speichern**
Als Spieler möchte ich den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern, um später weiterzuspielen.
Points:8

Task

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

#47-2 **Json Serialiser**
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

#47: **Speichern**
Als Spieler möchte ich den aktuellen Spielstand zu einem beliebigen Zeitpunkt auf der Festplatte speichern, um später weiterzuspielen.
Points:8

Task

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

#47-1 Das Sp…
#47-2 **Json Serialiser**
…N
8h
#47-5 **Speichern Stresstest**
Speichern von mindestens > 1500 Objekten auf Windows und Linux testen.
3h

Product Increment

- Neue Produktversion am Ende des Sprints
- Ist direkt an den Kunden auslieferbar

Definition of Done

- Definiert wann ein Item als fertig angesehen wird
- Wird vom Team erstellt
- Darf sich im Laufe des Projekts ändern

Product Increment

- Neue Produktversion am Ende des Sprints
- Ist direkt an den Kunden auslieferbar

Definition of Done

- Definiert wann ein Item als fertig angesehen wird
- Wird vom Team erstellt
- Darf sich im Laufe des Projekts ändern

```
Definition of Done Beispiele
- Team einverstanden (incl. PO)
- Auf master eingecheckt
- 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
    ... am Ende des nächsten Sprints sollte die Kernmechanik vollständig spielbar sein, damit wir ausprobieren können ob sie Spaß macht.
- 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

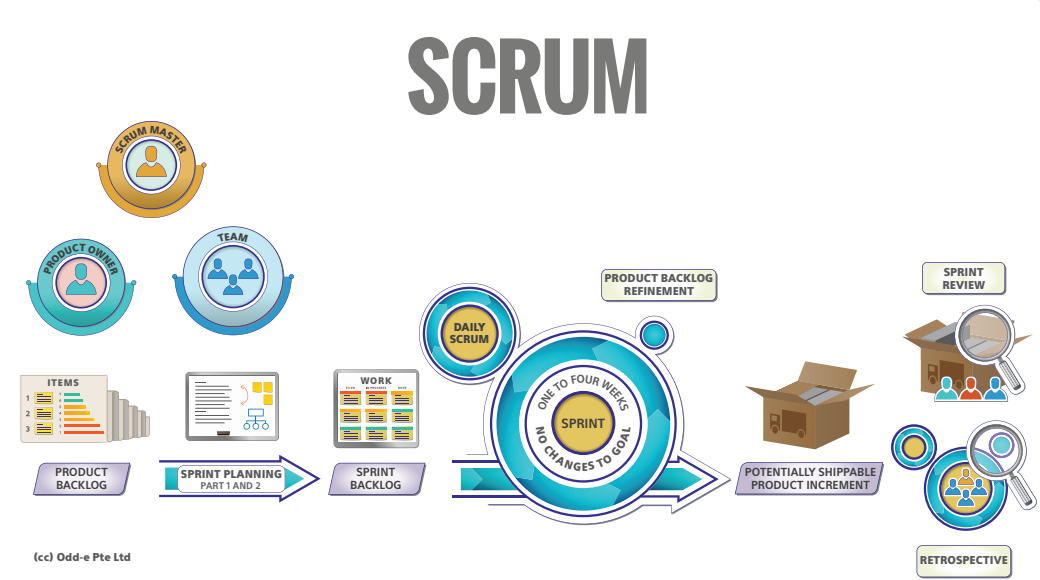
- Tägliches Treffen
- Maximal 15 Minuten
- Developer beantworten nacheinander
  - Was habe ich seit dem letzten Treffen getan?
    ..., habe Spielfigur jumper angelegt, ...
  - Was plane ich bis zum nächsten Treffen zu tun?
    ..., werde jumper heute an die Physik anbinden, ...
  - Welche Probleme hatte ich und wo benötige ich Hilfe?
    ..., verstehe aber die Schnittstelle zum RigidBodyCollider nicht.
- Detailfragen und Lösungssuche gehören nicht zum Daily Scrum
  - Gesondertes Treffen für Detailfragen und Lösungssuche vereinbaren

Sprint Review

- Treffen am Ende des Sprints
- Länge abhängig von Sprintlänge (ca. 1h je Woche)
- Developer präsentieren das Product Incemrent
- 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 und vor dem Sprint Planning
- Länge abhängig von Sprintlänge (ca. 45min je Woche)
- Mit Ziel, 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?
- Unterstützt durch Scrum Master


SCRUM
SCRUM MASTER
PRODUCT OWNER
TEAM
ITEMS
1
2
3
PRODUCT BACKLOG
SPRINT PLANNING
PART 1 AND 2
WORK
SPRINT BACKLOG
DAILY SCRUM
PRODUCT BACKLOG REFINEMENT
ONE TO FOUR WEEKS
SPRINT
NO CHANGES TO GOAL
POTENTIALLY SHIPPABLE PRODUCT INCREMENT
SPRINT REVIEW
RETROSPECTIVE
(cc) Odd-e Pte Ltd

Scrum im Sopra

Scrum im Sopra · Rollen

- Dozenten sind Kundenvertreter
- Tutoren sind Scrum Master
- Sie sind Developer
- Sie können in Ihrer Gruppe Product Owner werden

Scrum im Sopra · Rollen

- Dozenten sind Kundenvertreter
- Tutoren sind Scrum Master
- Sie sind Developer
- Sie können in Ihrer Gruppe Product Owner werden

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

Scrum im Sopra · Rollen

- Dozenten sind Kundenvertreter
- Tutoren sind Scrum Master
- Sie sind Developer
- Sie können in Ihrer Gruppe Product Owner werden

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

Tutoren fungieren zusätzlich als Kundenversteher und Berater

Product Backlog

Menge von Items

- Titel
- Sinnvolle Beschreibung
- Labels nach Funktion

  bug, question, ...
- Wichtigkeit

  low, mid, high
- Zeitschätzung (Gesamtarbeitszeit)

  est: 0h, 1h, 2h, ...,∞

Product Backlog

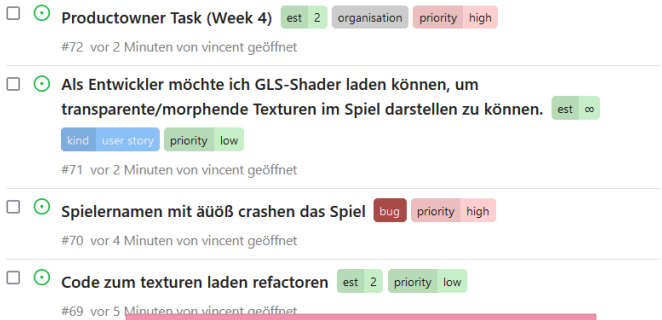
Menge von Items

- Titel
- Sinnvolle Beschreibung
- Labels nach Funktion

  bug, question, ...
- Wichtigkeit

  low, mid, high
- Zeitschätzung (Gesamtarbeitszeit)

  est: 0h, 1h, 2h, ...,∞

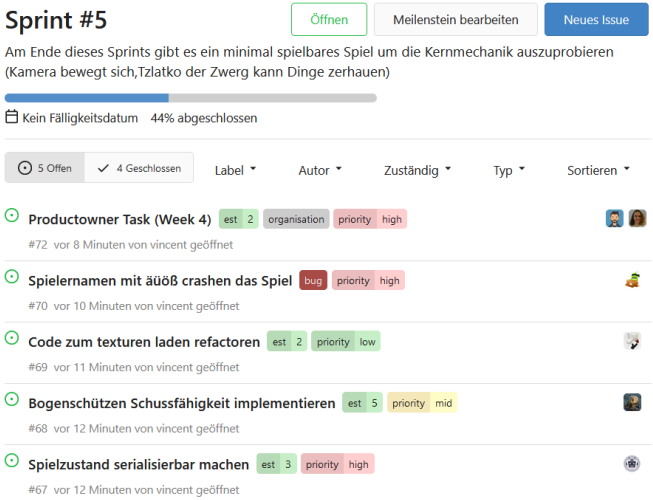


Annahmen:

- Das Game Design Document ersetzt die User Stories.
- Es können direkt Items abgeleitet werden (in Task-Größe).

Sprint Backlog

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



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
3. Sprint Planning

Sprint Review

- Developer führen Product Increment v
- Gruppe entscheidet was gemäß DoD
  - Nicht erledigte Items zurück ins Proc
- 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
  - Was lief gut? Was lief schlecht? Was muss sich ändern? Was soll so bleiben?
- 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 das Ziel des nächsten Sprints vor
- Product Backlog durchgehen, angefangen bei den wichtigsten Items:
  - Gemeinsam gewünschte Funktionalität besprechen und Aufwand schätzen
  - Ins Sprint Backlog verschieben
  - Wiederholen bis verfügbare Arbeitszeit aufgebraucht
- Items im Sprint Backlog an die Gruppenmitglieder verteilen

Scrum im Sopra · Backlog Items ableiten

Items werden aus dem GDD

- Das GDD übernimmt im
  User Stories
- Kapitel: Zusammenfass
  (Objekte, Aktionen, Ablauf, ...)

- Die Domäne analysieren und verstehen
- um die Anforderungen zu erheben und zu verstehen
- um die Umsetzung zu konzipieren
- um das benötigte Produkt zu implementieren.

Scrum im Sopra · Backlog Items ableiten

Items werden aus dem GDD abgeleitet

- Das GDD übernimmt im Sopra die Rolle von User Stories
- 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.

Scrum im Sopra · Backlog Items ableiten

Items werden aus dem GDD abgeleitet

- Das GDD übernimmt im Sopra die Rolle von User Stories
- Kapitel: Zusammenfassung, Spiellogik (Objekte, Aktionen, Ablauf, ...)

#23 **Objekt Dunkelheit erstellen**
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.

Scrum im Sopra · Backlog Items ableiten

Items werden aus dem GDD abgeleitet

- Das GDD übernimmt im Sopra die Rolle von User Stories
- 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.

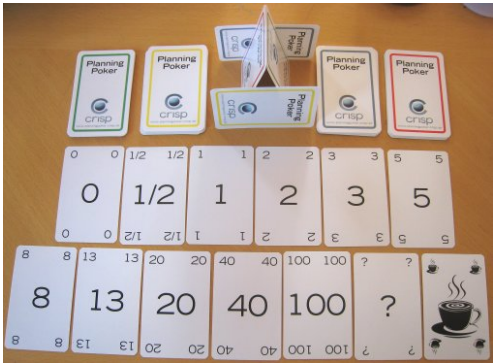
#23 **Objekt Dunkelheit erstellen**
Asset für die Dunkelheit (unendlich breite Wand mit Schatteneffekten) erstellen und im Code verfügbar machen.
?

#24 **Verhalten Dunkelheit implementieren**
Verhaltenskomponente für die Dunkelheit implementieren: verfolgt den Spieler mit zunehmender Geschwindigkeit.
?

Scrum im Sopra · Zeiten schätzen

Mit Planning Poker

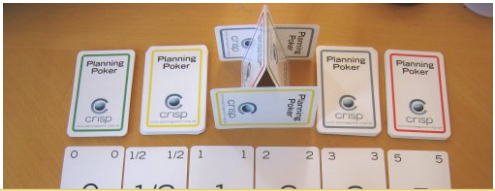
1. Item besprechen,
   jeder soll versteht worum es geht
2. Verdeckt wählt jeder einen Wert aus
   (Story Points/Stunden)
3. Gleichzeitig aufdecken
4. Wer den höchsten bzw. niedrigsten Wert
   aufdeckt, erklärt warum
5. Wiederholen bis alle den gleichen Wert zeigen



Scrum im Sopra · Zeiten schätzen

Mit Planning Poker

1. Item besprechen,
   jeder soll versteht worum es geht
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 implementieren**

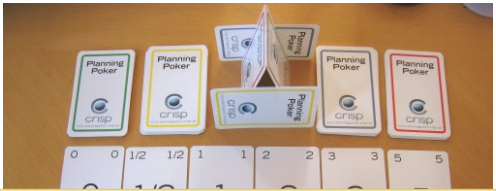
Verhaltenskomponente für die Dunkelheit implementieren: verfolgt den Spieler mit zunehmender Geschwindigkeit.

?

Scrum im Sopra · Zeiten schätzen

Mit Planning Poker

1. Item besprechen,
   jeder soll versteht worum es geht
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 implementieren**

Verhaltenskomponente für die Dunkelheit implementieren: verfolgt den Spieler mit zunehmender Geschwindigkeit. ... wie ein Anhänger an einem langen Gummiband ... .

5h

Scrum im Sopra · Beispieltickets

#46: **Feuerball hinzufügen**
Die Fähigkeit Feuerball (siehe GDD) als Komponente für Einheiten implementieren.

est:8

#46: **Feuerball hinzufügen**
Die Fähigkeit Feuerball (siehe GDD) als Komponente für Einheiten implementieren.

Akzeptanzkriterien:

- Es gibt eine (debug) Einheit, die die Fähigkeit Feuerball besitzt.
- Der Feuerball fliegt graphisch sichtbar auf ein selektiertes Ziel zu.
- Das Ziel bekommt entsprechend der Fähigkeit Feuerball Schaden.

est:8

Scrum im Sopra · Beispieltickets

#46: **Feuerball hinzufügen**
Die Fähigkeit Feuerball (siehe GDD) als Komponente für Einheiten implementieren.

Akzeptanzkriterien:

- Es gibt eine (debug) Einheit, die die Fähigkeit Feuerball besitzt.
- Der Feuerball fliegt graphisch sichtbar auf ein selektiertes Ziel zu.
- Das Ziel bekommt entsprechend der Fähigkeit Feuerball Schaden.

est:8

#42: **Räumliche Datenstruktur implementieren**
Effiziente Verwaltung von *IPosition* Objekten.

est:16

#46: **Feuerball hinzufügen**
Die Fähigkeit Feuerball (siehe GDD) als Komponente für Einheiten implementieren.

Akzeptanzkriterien:

- Es gibt eine (debug) Einheit, die die Fähigkeit Feuerball besitzt.
- Der Feuerball fliegt graphisch sichtbar auf ein selektiertes Ziel zu.
- Das Ziel bekommt entsprechend der Fähigkeit Feuerball Schaden.

est:8

#42: **Räumliche Datenstruktur implementieren**
Effiziente Verwaltung von *IPosition* Objekten.

- Einfügen eines Objekts per *.Add(IPosition o)*
- Finden eines Objekts per *.CheckAt(Vector2 v)*
- Abfragen von *.InFrontOf(Vector2 v)*
- Updatet wenn sich Objekt bewegt.
- Tausend Objekte Minimum !!!111!!1

est:16

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
- In Ticket dokumentieren
  - Kurzvorstellung in Sprintreview

- Alle Recurring Tasks müssen verteilt werden
- Ticket mit Zeitschätzung (max. 2h) je Aufgabe und Sprint

Die ersten Schritte

Spielidee überlegen

- Anforderungen
  Bekannte Spiele? Einschränkungen? Naheliegende Mechaniken? Begriffe unklar?
- Bekanntes verändern und kombinieren
  Mechanisch? Setting und Handlung?
- Sie sind nicht allein!
  - Was ist Ihnen wichtig?
  - Was wollen Sie gar nicht?
  - Reden Sie mit ihrer Gruppe
  - Fragen Sie nach

Spielidee überlegen

- Anforderungen
  Bekannte Spiele? Einschränkungen? Naheliegende Mechaniken? Begriffe unklar?
- Bekanntes verändern und kombinieren
  Mechanisch? Setting und Handlung?
- Sie sind nicht allein!
  - Was ist Ihnen wichtig?
  - Was wollen Sie gar nicht?
  - Reden Sie mit ihrer Gruppe
  - Fragen Sie nach

Ihre Idee . . .

- muss von der Gruppe verstanden werden.
- kann verworfen werden.
- wird sich (stark) verändern.
- wird zusammen weiterentwickelt.

Spielidee ausarbeiten

- 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 (Entitäten, Funktionen, Ereignisse, Interaktionen) gibt es?
- Was ist Ihnen unklar (in der Domäne und der Spielidee)?
- Lücken im Ablauf Spielablauf füllen

Spielidee ausarbeiten

- 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 (Entitäten, Funktionen, Ereignisse, Interaktionen) gibt es?
- Was ist Ihnen unklar (in der Domäne und der Spielidee)?
- Lücken im Ablauf Spielablauf füllen

```
Der Spieler kämpft als Krieger in einem Verlies.
```

Spielidee ausarbeiten

- 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 (Entitäten, Funktionen, Ereignisse, Interaktionen) gibt es?
- Was ist Ihnen unklar (in der Domäne und der Spielidee)?
- Lücken im Ablauf Spielablauf füllen

```
Der Spieler kämpft als Krieger in einem Verlies.
- Objekte: Verlies und Krieger
- Aktion: Kämpfen
- Hat er ein Schwert? Zauber? Was ist ein Verlies? Gibt es nur Krieger?
```

Spielidee ausarbeiten

- 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 (Entitäten, Funktionen, Ereignisse, Interaktionen) gibt es?
- Was ist Ihnen unklar (in der Domäne und der Spielidee)?
- Lücken im Ablauf Spielablauf füllen

```
Der Spieler kämpft als Krieger in einem Verlies.
- Objekte: Verlies und Krieger
- Aktion: Kämpfen
- Hat er ein Schwert? Zauber? Was ist ein Verlies? Gibt es nur Krieger?
Die Dunkelheit verfolgt den Spieler und setzt ihn unter Druck.
```

Spielidee ausarbeiten

- 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 (Entitäten, Funktionen, Ereignisse, Interaktionen) gibt es?
- Was ist Ihnen unklar (in der Domäne und der Spielidee)?
- Lücken im Ablauf Spielablauf füllen

```
Der Spieler kämpft als Krieger in einem Verlies.
- Objekte: Verlies und Krieger
- Aktion: Kämpfen
- Hat er ein Schwert? 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?
```

Spielidee ausarbeiten

Das beta-GDD ist bereits eine vollständige Beschreibung des Spiels

- 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 (Entitäten, Funktionen, Ereignisse, Interaktionen) gibt es?
- Was ist Ihnen unklar (in der Domäne und der Spielidee)?
- Lücken im Ablauf Spielablauf füllen

```
Der Spieler kämpft als Krieger in einem Verlies.
- Objekte: Verlies und Krieger
- Aktion: Kämpfen
- Hat er ein Schwert? 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?
```

Spielidee dokumentieren

Spielkonzept

- Zusammenfassung
- Zentrale Spielmechanik

Benutzeroberfläche

- Spielerinterface

Spiellogik

- Aktionen und Optionen
- Spielobjekte
- Spielstruktur

Was nun?

Ab sofort

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

Danach

- Termin für Gruppentreffen (Mo-Do) finden (Doodles ausfüllen, bis Sonntag 23:59)
- Hausaufgaben fertig machen
- Spielidee entwickeln

Was nun?

Ab sofort

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

Überlegen Sie vorher, ob Sie:

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

Danach

- Termin für Gruppentreffen (Mo-Do) finden (Doodles aus
- Hausaufgaben fertig machen
- Spielidee entwickeln

Was nun?

Ab sofort

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

Danach

- Termin für Gruppentreffen (Mo-Do) finden (Doodles aus
- Hausaufgaben fertig machen
- Spielidee entwickeln

Überlegen Sie vorher, ob Sie:

- dieses Semester genug Zeit haben
- zu allen Pflichtterminen anwesend sein können
- zugriff auf einen zum Entwickeln geeigneten

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