



# Projektplan

Project: *GlowingBroccoli*

**Version 2.3**

## Änderungsgeschichte

Datum	Version	Autor	Beschreibung
2020-10-05	1.0	Wegmann S.	Dokument erstellt
2020-11-02	2.0	Pross N.	Dokument bearbeitet
2020-11-02	2.1	Roy J.	Dokument bearbeitet
2020-11-02	2.2	Robertz J.	Dokument bearbeitet
2020-11-02	2.3	Wegmann S.	Dokument bearbeitet

	Datum	Unterschrift
Pross Naoki	_____	_____
Robertz Joanne	_____	_____
Roy Jérôme	_____	_____
Wegmann Steven	_____	_____

## Inhaltsverzeichnis

<b>1 Team</b>	<b>3</b>
<b>2 Projektstrukturplan</b>	<b>4</b>
<b>3 Arbeitspakete</b>	<b>5</b>
<b>4 Meilensteine</b>	<b>5</b>
4.1 Debug / Normale . . . . .	5
4.2 Release . . . . .	6
<b>5 Terminplan</b>	<b>6</b>

## Abbildungsverzeichnis

## Tabellenverzeichnis

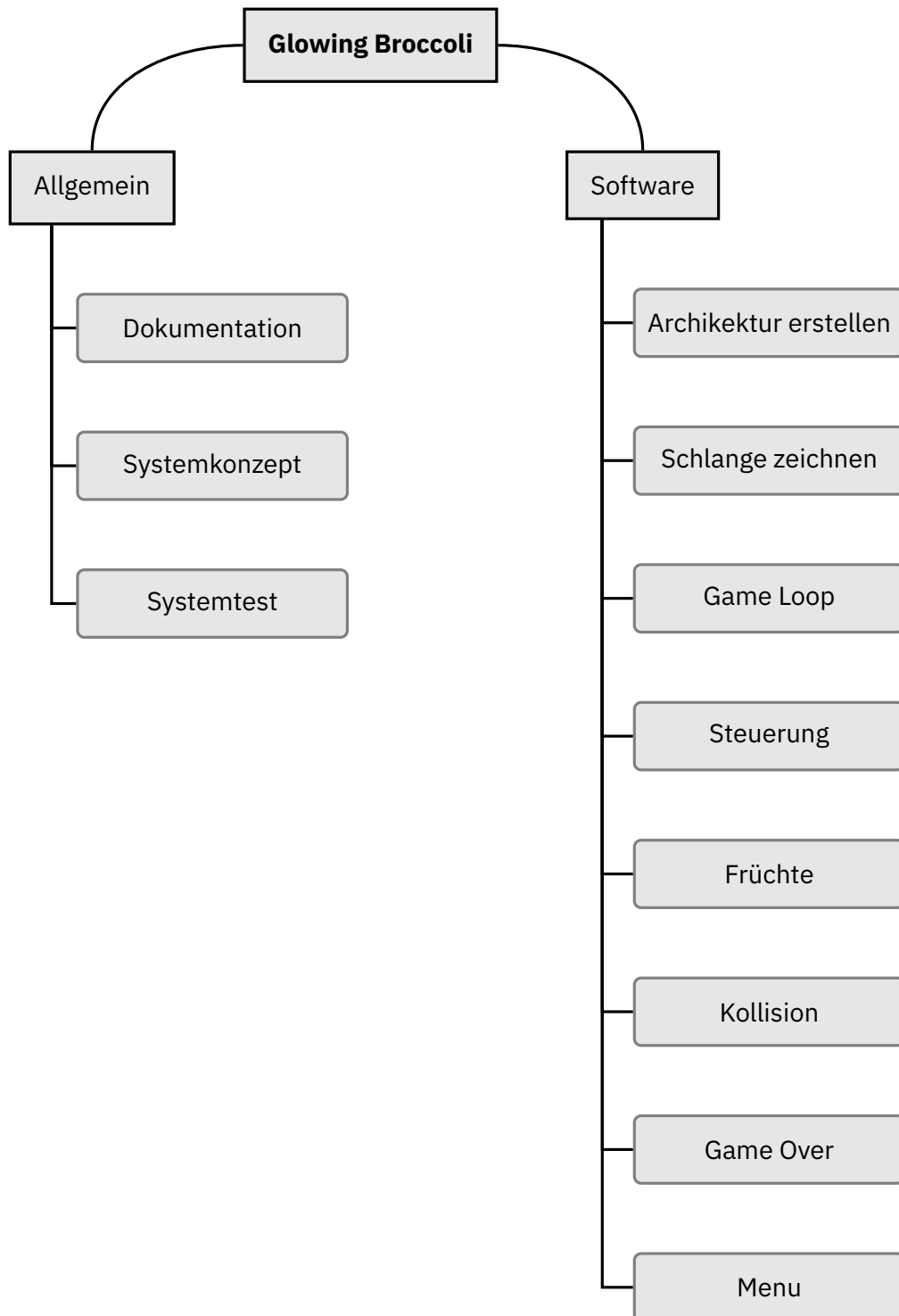
1 Arbeitspakete . . . . .	5
---------------------------	---

## 1 Team

Das Team besteht aus folgenden Studenten:

- Naoki Pross
- Joanne Robertz
- Jérôme Roy
- Steven Wegmann

## 2 Projektstrukturplan



### 3 Arbeitspakete

Tabelle 1: Arbeitspakete

Allgemein	Beschreibung
Dokumentation	Dokumentation beinhaltet folgende Themen: Requirements Specification (separates Dokument), Beschreibung alle Software und Hardware APs und deren Resultat. Beinhaltet nicht: Benutzerhandbücher.
Systemkonzept	Erstellen der Systemarchitektur mit deren Komponenten und Schnittstellen.
Systemtest	Erstellen eines Systemtest welche alle User Anforderungen erfüllen.
Architektur erstellen	Erstellen der Softwarearchitektur mit UML.
Schlange mit Qt zeichnen	Die Schlange wird auf ein Window gezeichnet.
Game Loop implementieren	Ein inkrementierender Timer aktualisiert die Position der Schlange.
Steuerung hinzufügen	Der Spieler muss in der Lage sein, die Schlange mit der Tastatur zu steuern.
Früchte	Auf der Karte wird zufällig ein Stück Nahrung generiert, das die Schlange fressen kann.
Kollision der Objekte ermöglichen	Nach einer Kollision mit einer Frucht oder sich selbst wird ein Event ausgeführt.
Game Over Status	Use Case Game Over gemäss SRS implementieren.
Menu Status	Use Case Menu gemäss SRS implementieren.
Spiel pausieren ermöglichen	Use Case Pause gemäss SRS implementieren.
Scoreboard einfügen	Use Case Scoreboard gemäss SRS implementieren.

### 4 Meilensteine

#### 4.1 Debug / Normale

1. Grafische Oberfläche und leere Game Loop realisieren
2. Schlange bewegt sich und kann gesteuert werden

3. Collision Checking und Game Over sind implementiert
4. Score für ein Spiel wird gespeichert
5. Menu und Scoreboard sind implementiert
6. Spiel kann pausiert werden

#### **4.2 Release**

1. Spiel ist fertig und spielbar
2. Framerate und Update-Rate sind stabil

### **5 Terminplan**

Siehe der Anhang Gantt-Diagramm.pdf.