# Teilnehmer/innen des Teams:

|  |  |
| --- | --- |
| Klasse:  BI19a | Team:  Allewelt, Sureskumar, Baskaran |

# Anforderungsdefinition (Meilenstein A)

|  |  |
| --- | --- |
| „Flappy Bird aber anderst“ | |
| **Auftrag:**  (Allgemeine Beschreibung) | Nutzen: Das Spiel dient zur Unterhaltung.**Szenario:**  * Fliegendes Objekt -> muss ans Ziel gelangen ohne dabei Hindernisse zu berühren.   **Details:**   * Zwei Levels (je nach Level -> unterschiedlicher Schwierigkeitsgrad) * Welten und Hindernisse ändern sich je nach Levels |
| **MUSS**  **Kriterien:**  (Konkrete Features, die umzusetzen sind) | **Folgende Features sollen implementiert werden (Funktionalität):**   * Zwei Levels * Unterschiedliche Welten und Hindernisse je nach Level * Unterschiedlicher Schwierigkeitsgrad je nach Level * Flüssiger Übergang zwischen Levels * Pause + Restart Button |

|  |  |
| --- | --- |
| **KANN**  **Kriterien:**  (Konkrete Features, die optional sind) | **Folgende Features können zusätzlich implementiert werden: (Kreativität)**   * Diverse Hindernisse * Objekt fliegt unerwartet Rückwärts * Flugrichtung ändert sich von links nach rechts -> unten nach oben * Unerwartetes Verschwinden von Hindernissen * Durch Sammeln von Objekten -> «Leben» sammeln * Plötzliches Vergrössern / Verkleinern vom fliegenden Objekt |

## Planung LB2

|  |  |  |  |
| --- | --- | --- | --- |
| *MS* | *Tätigkeit / Abgabe* | *Soll-Datum* | *Ist-Datum* |
| A | ProjektstartTeam BildungWahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson |  |  |
| B | Teamaufgabe 1:Abgabe: Lösungsdesign  (Analyse, Design: Funktionsmodell, UseCase, GUI, Storyboard) |  |  |
| B2 | Teamaufgabe 2:Abgabe: Testvorschrift und Testfälle |  |  |
| C | Einzelaufgabe 3:Abgabe Szenario (.zip) mit Inline-Dokumentation, Systemdokumentation (UML Klassen-, Sequenzdiagramm)Fachgespräch Projektabnahme |  |  |
| C2 | Einzelaufgabe 4:Abgabe: Ausgefüllter Systemtest |  |  |

# Lösungsdesign (Meilenstein B: Teamaufgabe 1)

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

## Funktionsmodell

Identifizierung der Objekte und Konzepte (Inter-Aktionen):

Objekte:

Vogel, Hindernisse, Welt, Extra Leben Gadget, Welt dreht sich Gadget, Gadget Hindernis verschwindet

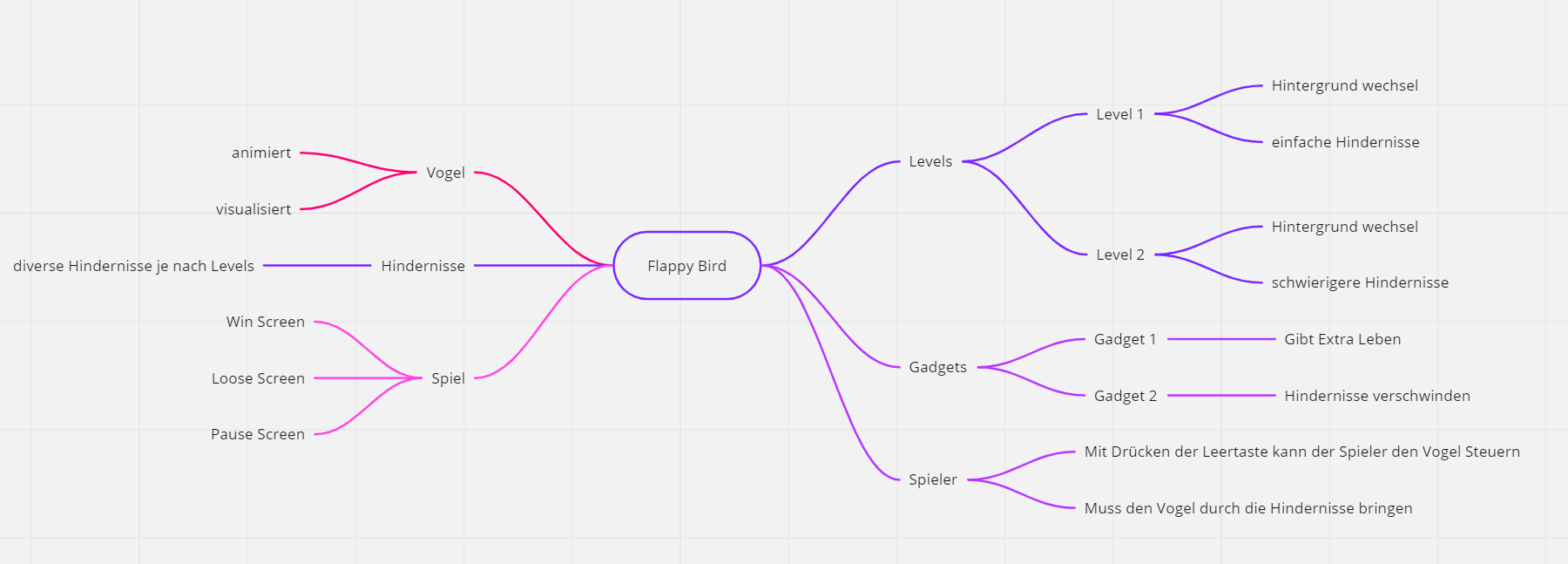
Konzepte:

Ausweichen von Hindernis, Random Gadget generieren, Mehrere Level

Im Folgenden sind die erwarteten Eingaben und Ausgaben beschrieben / dargestellt:

Legende:

* Spieler fliegt mit Vogel (Leertaste)
* Die Hindernisse ausweichen
* Gadget einsammeln und Belohnung bekommen
* Level 1 beenden und Level 2 starten

Im Folgenden sind die erwarteten Eingaben und Ausgaben beschrieben / dargestellt:

## Anwendungsfälle (UseCases)

Folgende Anwendungsfälle sind hier detailliert dokumentiert:

Detaillierte Beschreibung der UseCases

* Der Spieler bedient die Simulation, in dem er das Szenario steuert und den Vogel durch die Hindernisse fliegen lässt
* Der Vogel wird animiert und visualisiert
* Hindernisse werden visualisiert
* Gadgets werden visualisiert
* Spieler übernimmt Steuerung des Vogels

## Ablauf

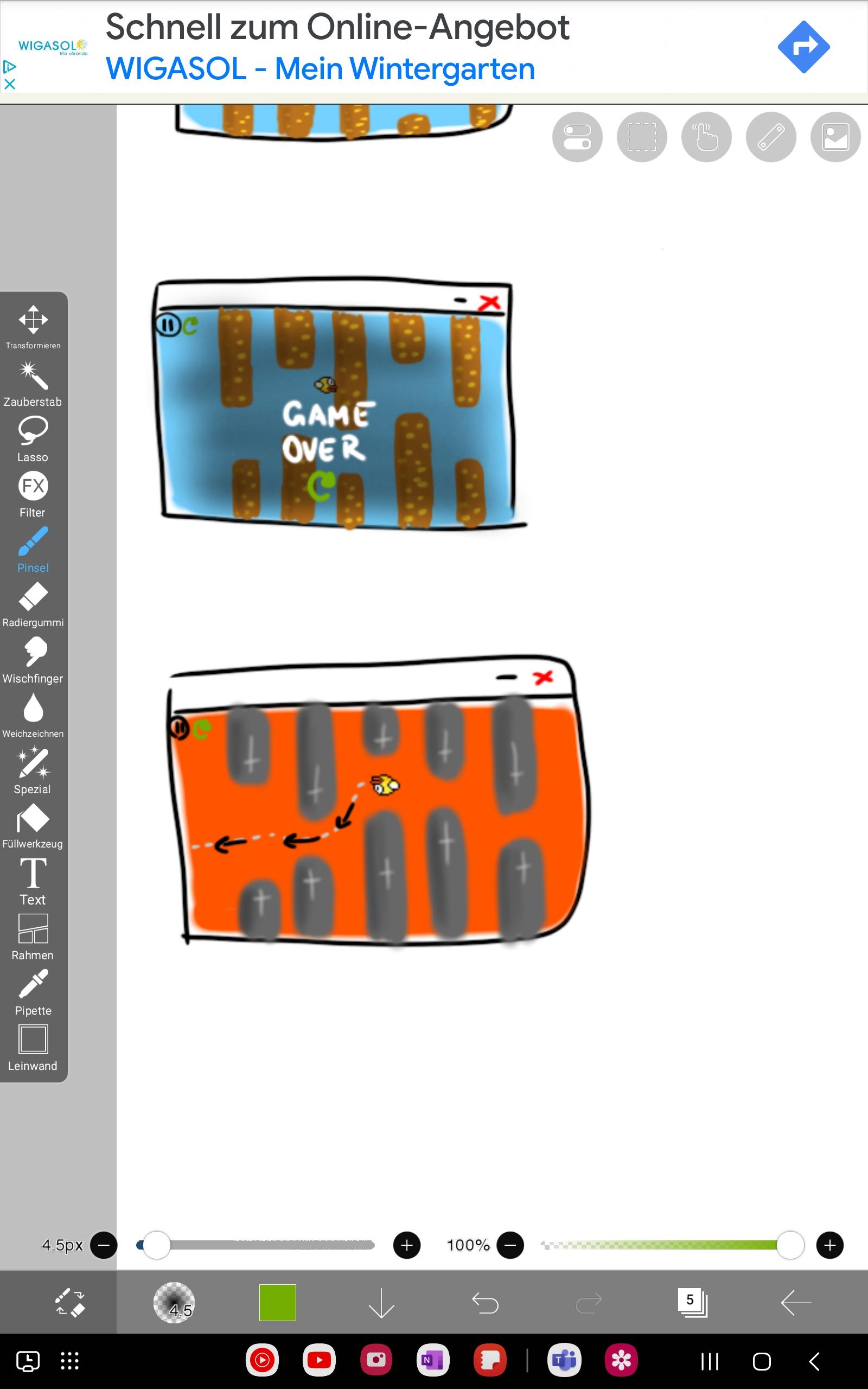
- Level 1

- Der Vogel fliegt von links nach rechts

- Der Vogel wird automatisch nach unten gezogen

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

Storyboard



- Level 2

- Die Welt hat sich gedreht (Welt 2)

- Der Vogel fliegt von rechts nach links

- Der Vogel wird automatisch nach oben gezogen

Muss Kriterien:

* Spiel starten (Play Button)
* Level 1:
  + - Fenster mit Welt 1
    - Pause + Restart Button
* Ziel: das 2.Level zu erreichen, ohne die Hindernisse zu berühren
* Beim Erreichen des 2.Levels:
  + - Fenster mit Welt 2 (schwierigere Hindernisse)
    - Pause + Restart Button
* Beim Berühren einer der Hindernisse, ist das Spiel vorbei (Game Over)
  + - 1. Möglichkeit -> Spiel erneut spielen
    - 2. Möglichkeit -> Fenster schliessen

# Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument

***M226B\_Vorlage\_Systemtest mit UC\_V1.1.docx***

# Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

Das erstellte Java-Projekt (Greenfoot-Szenario) ist hier detailliert abgelegt:

***M226B\_ Aufgabe\_3\_Szenario\_Allewelt.zip***

## Statisches Design: Klassendiagramm

## Umfang / Abgrenzung / Änderungen gegenüber Design

Aufgrund unten beschriebener Umstände sind Anpassungen des ursprünglichen Lösungsdesigns gemacht worden:

Ursprünglich war ein extra Leben gedacht, jedoch konnte ich dies nicht so umsetzten und unser Flappy ist stattdessen für ein paar Sekunden unsterblich.

Leider konnte ich die restlichen Kann-Kriterien nicht machen.

## Funktionalität der Implementation.

Zusätzlich zu der Inline-Dokumentation sind hier folgende Funktionen detailliert beschrieben:

JavaDoc!

## Dynamische Struktur: Sequenzdiagramm

# Bedienungsanleitung (Meilenstein C: individuelle Aufgabe 3)

# Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument

***M226B\_Vorlage\_Systemtest mit UC\_V1.1.docx***