

Konzeption Abschlusssaufgabe: „Tweety Bird“

1. Funktionale Analyse

a. Benutzererlebnis und Art der Interaktion

- Der Spieler übernimmt die Steuerung (Maus Bewegung) von Tweety und muss alle Fliegen und Marienkäfer einsammeln.
- Spiel ist gewonnen, wenn alle Fliegen und Marienkäfer eingesammelt sind
- Spiel ist verloren wenn zu viele Bienen gefressen werden
- Durch einen Alert wird der Nutzer ins Spiel eingebunden

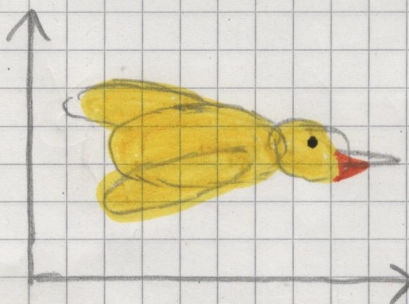
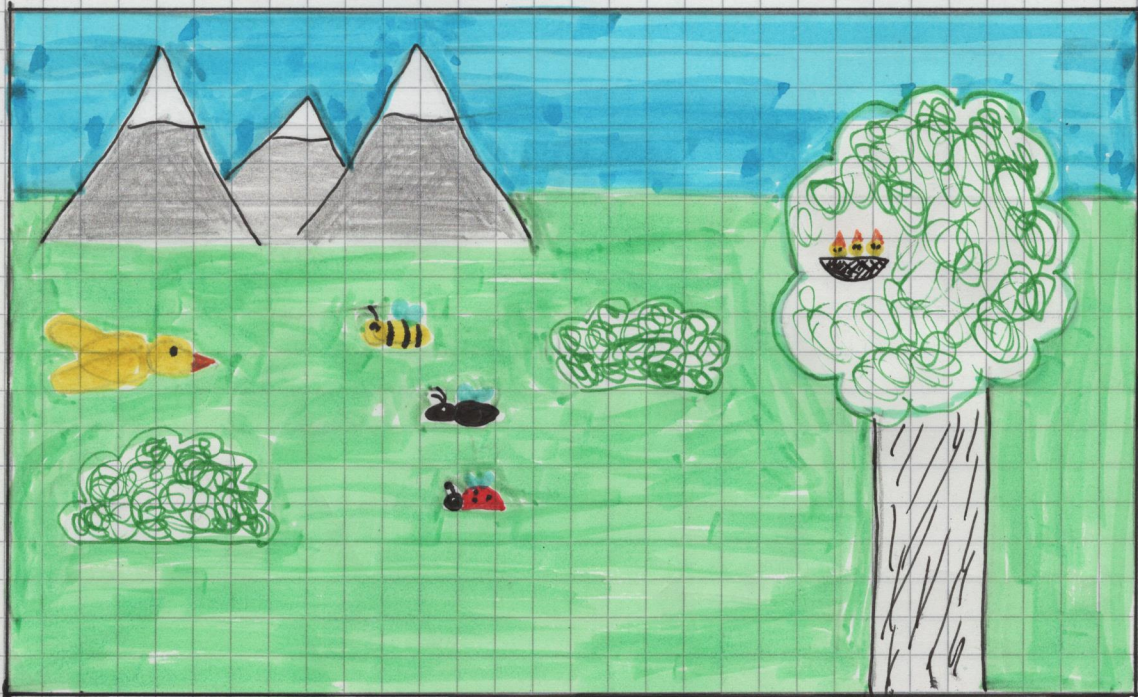
b. Detailbeschreibung von Aufbau & Ablauf

- Alert zu Beginn (noch vor Aufbau der Tiere) führt in die Geschichte des Spiel ein und gibt einen Einblick was der Spieler machen muss
- Interface: Spielfenster (Canvas) -> Landschaft mit einem Baum in dem ein Vogelnest ist
- Tweety als Vogel, wird durch den Cursor (am PC) und durch Touch (Smartphone) gesteuert
- Die einzelnen Insekten: Fliegen, Marienkäfer und Bienen -> fliegen von rechts nach links durch den Canvas, sobald sie auf der linken Seite aus dem Bild herausfliegen kommen sie am rechten Bildrand wieder herein geflogen
- Die Anzahl der Fliegen und Marienkäfer ist zufällig, um zu gewinnen müssen alle der erschienen Insekten „gefressen“ werden, daraufhin wird ein WinningScreen geöffnet
- Das Spiel ist verloren sobald 3 Bienen gefressen werden, daraufhin wird der LoosingScreen geöffnet und der Spieler wird daraufhin hingewiesen, dass er das Spiel neu laden muss

c. Nutzerinteraktionen und Systemreaktionen

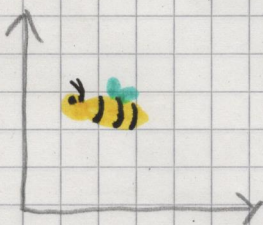
- Klick auf OK-Button im Alert-Fenster: Canvas wird aufgebaut, Animation der Insekten startet, der Cursor wird in der Bewegung durch Objekt Tweety ersetzt
- Durch die Bewegung der Maus wird Tweety über den Canvas bewegt
- Fährt der Spieler mit Tweety über ein Insekt wird dieses „gefressen“ (es wird aus dem Array herausgenommen)
 - ➔ Werden alle Fliegen und Marienkäfer gefressen erscheint der Winning Screen
 - ➔ Werden 3 Bienen gefressen erscheint der Loosing Screen
- Durch Reload der Seite wird das Spiel neu gestartet

Konzept



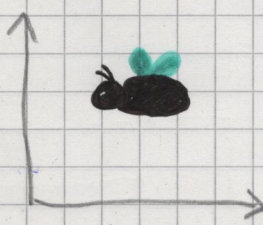
Tweety Bird

=> ersetzt den Cursor,
Spieler soll Tweety helfen
die Bugs zu fangen



Bee

=> Diese Biene
sind gemein und
stechen Tweety.
Sie dürfen nicht
gefressen werden



Fly

=> Fressen für
Tweety



Ladybug

=> Fressen für
Tweety

2. Technische Analyse

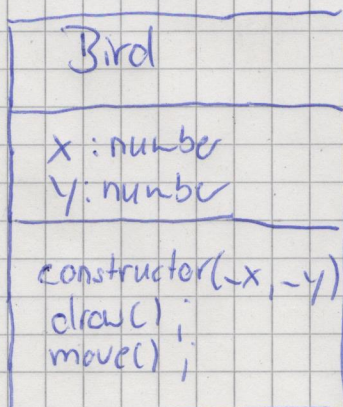
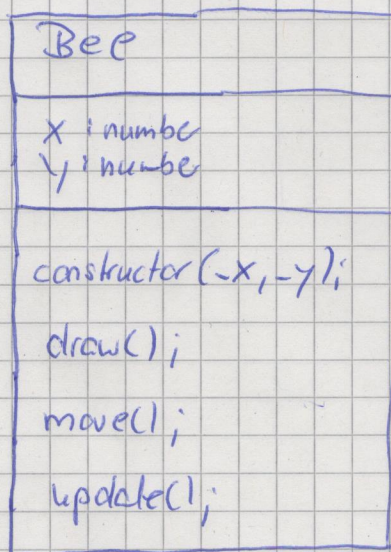
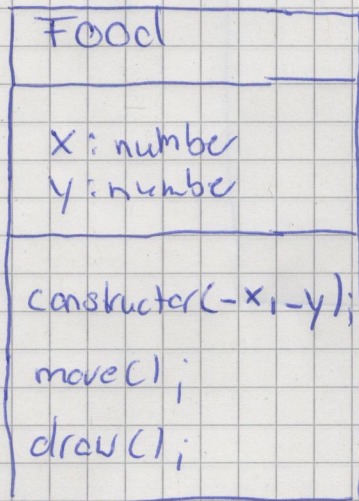
a. Grundlegende Überlegungen:

- OOP durch Klassenzuweisung der Ballontypen
- Background in separater Typescript-Datei generieren
- Speicherung von Objekten in Arrays zur vereinfachten Verwaltung (Fliegen und Marienkäfer als essbare Insekten und die Bienen, die stechen)
- Löschung von Inhalten aus Arrays, Insekten, die gefressen werden
- Mousemove-/Touchmove-Events, die verschiedene Funktionen aufrufen
- Variable Anzahl und Art von Objekten durch mathematische Zufallsgenerierung

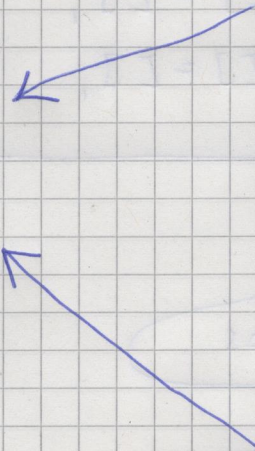
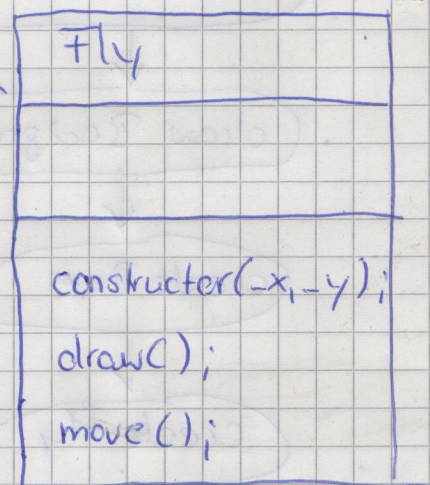
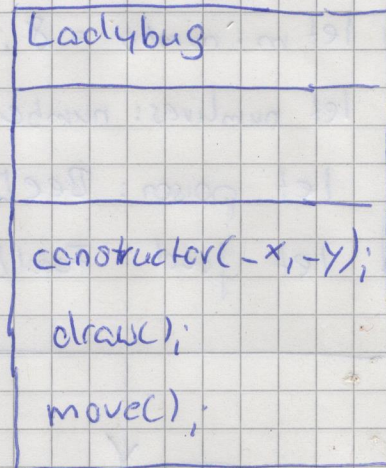
Technische Analyse

Klassendiagramme

Superklasse



Subklassen



Aktivitätsdiagramme

