Monkey Business

```
Inhalt
1--Erklärung & Installation
2--Klassendiagramm
     Superklassen dunkelblau
     Subklassen hellblau
3--Aktivitätsdiagramme
     Hauptprogramm
     init();
     startGame();
     animate();
     Funktionen für Eventlistener
     Funktionen für life, score, ende, restart
Aufrufe von Funktionen sind rot
        von Methoden sind orange
Senden & Empfangen von Events sind grün
erstellt von Laura Vogt, MtrNr. 256065, SS2018
```

1-- Erklärung & Installation

Installation:

- zum Spielen die index.html in den Browser ziehen bzw. Link öffnen (ausgelegt für Firefox)

zur Ordnerstruktur - Wo ist Was zu finden:

- -im Ordner "scripts" befinden sich alle nötigen script-Dateien
- -im Ordner "img" liegen alle von mir verwendeten png-Dateien
- -im Ordner "doc" findet sich diese Datei

Anleitung:

- zunächst gelangt man auf die Startseite
- dort ist nochmal eine kleine Anleitung
- bei click auf den Affen oder drücken der Leertaste gelangt man zum Spiel (wenn der Cursor über dem Affen liegt, ändert sich etwas =), wenn man den Cursor wieder von dem Affen weg bewegt,
- tritt der Ursprungszustand ein)
- man gelangt auf die Spielseite, das Spiel startet
- es geht darum, mit dem Affen Bananen zu sammeln und Kokosnüsse zu vermeiden
- Steuerung über Pfeiltasten links<-- & -->rechts
- man startet mit 5 Leben und einem Score von 0 (Anzeige oberer Bildrand)
- pro gefangene Banane gibt es einen Punkt
- jede Kokosnuss, die auf den Kopf des Affen fällt, zieht ein Leben ab
- das Spiel endet, wenn entweder alle Leben verbraucht oder ein Score von 50 erreicht wurde
- es erscheint ein alert() Fenster mit einer angepassten Nachricht
- (warum das Spiel zu Ende ist + wie viele Bananen man gefangen bzw. wie viele Leben man noch hat
- man gelangt wieder auf die Startseite
- über den restartButton kann man auf während des Spiels wieder auf die Startseite gelangen

2--Klassendiagramm

Superklassen

MovingObjects

x: number; y: number; borderS: number; borderE: number; speed: number; img: HTMLlmageElement;

constructor(_src: string, _typ: string)
this.img = new Image();
this.img.src = _src;
this.typ = _typ;

+ update(); + move(); + draw();

typ: string;

Subklassen

```
constructor(_src: string, _typ: string)
super(_src, _typ);
this.setParameter();

+ setParameter();
+ update():
+ move();
+ catching();
+ newX();
```

CollectorObject extends MovObjects

constructor(_src: string, _typ: string)
super(_src, _typ);
this.setParameter();

- + setParameter(); + move();
- + moveRight():
- + moveLeft();

extends

CounterObjects

constructor(x: number, y: number, src: string, counter: number)

- x: number;y: number;
- img: HTMLImageElement;
- counter: string;

this.x = _x; this.y = _y; this.img = new Image();

this.img = new image(); this.img.src = _src; this.counter = _counter.toString();

- + update(_counter: number);
- + draw();
- + newCounter(_counter: number);

Background - x: number:

- y: number;
- img: HTMLImageElement;

constructor

(_src: string, _x: number, _y: number)
this.img = new Image();
this.img.src = _src;

this.x = _x; this.y = _y;

draw()

3--Aktivitätsdiagramme Hauptprogramm document.addEventListener "load" init(); export crc2 Globale Variablen anlegen Arrays jwls für movingO/fallingO/counterO init(); Display "gamePage".display ="none" "startPage".display = "block" monkeyButt.addEventListener #monkeyButt overButton(); ("mouseover", overButton) "mouseover" monkeyButt.addEventListener #monkeyButt outButton(); ("mouseout", outButton) "mouseout" monkeyButt.addEventListener ("click", startGame) #monkeyButt "click" body.addEventListener ("keydown", pressSpace) "keydown" Start startGame(); remove.EventListener("keydown", pressSpace) Display "gamePage".display = "block" "startPage".display = "none" restartButt.addEventListener("click", restart) #restartButt "click" Variable für canvas canvas.getContext() Background erstellen bg = new Background(src, x, y) CounterObjecte erstellen scoreCounter : CounterObjects = new CounterObject(x, y, src, score) lifeCounter : CounterObjects = new CounterObjects(x, y, src, life) CounterObjects in cObj[] pushen CollectorObject erstellen $collector : Collector Object : Collector Object = new \ Collector Object (src, \ typ)$ collector in mObj[] pushen let i: number = 0; true let banana : FallingObjects = new FallingObjects(src, typ) i < anzahlBanana false banana in mObj[] und fObj[] pushen let i: number = 0 for i < anzahlCoco let coco: FallingObjects = new FallingObjects(src, typ) false i++ coco in mObj[] und fObj[] pushen body.addEventListener("keydown", keyEvent) "keydown" body. add Event Listener ("touch start", touch Event)"tochstart" animate(); clearRect putImgData score.draw() life.draw() bg.draw() let i: number = 0 setTimeout(animate, 25) for true false i<mObj.length continue mObj[i].update() let j: number = 0false true continue j<fObj.length let item = fObj[j]false false item.catching() ortsangaben if score==50 || life<1 true break case "gain" switch(item.typ) scoreEvent() true false true gameEnd(); break lifeEvent() case "destroy Ende overButton(); Variable für Zufallszahl _event.keyCode = Leertaste switch(zufallszahl) nothing happens Ende false false case 0 outButton(); ändere img.src ändere img.src ändere img.src img.src = ursprüngliche src break break break keyEvent(_event:KeyboardEvent); Start let i: number = switch(_event.keyCode) for event.touches.length true false false variable für clientX = _event.touches[i].clientX variable für clientY = _event.touches[i].clientY case 37 case 39 monkey.moveLeft(); true monkey.moveRigth(); monkey.moveLeft(); if clientX >canvas.widt/2 monkey.moveRight true restart(); gameEnd() switch true alert() location.reload(); break (score == 50)location.reload(); false location.reload(); case false true alert() break Ende | scoreEvent () lifeEvent() Start Start score++ scoreC.update(score) lifeC.update(life) Ende