Elias Kupeczki

07/02/2023

Modul 335

CsBe - API

Inhalsverzeichnis

[Aufgabenstellung 2](#_Toc126668317)

[Standards 2](#_Toc126668318)

[Schutzbedarfsanalyse 2](#_Toc126668319)

[Organisation der Arbeitsergebnisse 2](#_Toc126668320)

[Zeitplan 3](#_Toc126668321)

[Testkonzept 4](#_Toc126668322)

[Testfälle 4](#_Toc126668323)

[Anwendung der Fachsprache 4](#_Toc126668324)

[Benutzerfreundlichkeit: GUI, Bedienung 4](#_Toc126668325)

[Deployment 4](#_Toc126668326)

[Software-Ergonomie 4](#_Toc126668327)

[Design 5](#_Toc126668328)

[Scene 5](#_Toc126668329)

[Camera 5](#_Toc126668330)

[Menu 5](#_Toc126668331)

[Game Over 5](#_Toc126668332)

[Border 5](#_Toc126668333)

[Player 5](#_Toc126668334)

[Controls 5](#_Toc126668335)

[Collision 5](#_Toc126668336)

[Ball 6](#_Toc126668337)

[Direction 6](#_Toc126668338)

[Collision 6](#_Toc126668339)

[Enemy 6](#_Toc126668340)

[Collision 6](#_Toc126668341)

[Arbeitsjournal 6](#_Toc126668342)

[Reflexionsfähikeit 6](#_Toc126668343)

[Prägnänz 6](#_Toc126668344)

[Grafiken, Bilder, Diagramme und Tabellen 6](#_Toc126668345)

[Durchführung und Auswertung der Tests 6](#_Toc126668346)

[Storyboard 6](#_Toc126668347)

# Aufgabenstellung

# Standards

# Schutzbedarfsanalyse

# Organisation der Arbeitsergebnisse

# Zeitplan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **06-Feb** | | **07-Feb** | | **08-Feb** | | **09-Feb** | | **10-Feb** | | **11-Feb** | | **12-Feb** | |
|  | SOLL | IST | SOLL | IST | SOLL | IST | SOLL | IST | SOLL | IST | SOLL | IST | SOLL | IST |
| **Planen** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Docum-entation  Schreiben** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Scene erstellen** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C# Classes erstellen** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **GUI  erstellen** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **zusatz funktionen** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Testkonzept

# Testfälle

# Anwendung der Fachsprache

# Benutzerfreundlichkeit: GUI, Bedienung

# Deployment

Android only exclusive

# Software-Ergonomie

# Design

General Design

Diagram

Description automatically generated

## Scene

### Camera

### Menu

### Game Over

### Border

## Player

if (Input.GetKey(KeyCode.A)){

transform.Translate(Vector3.left \* Time.deltaTime \* 10);

        }



### Controls

The user is given the ability to move left and right. This is done using (example right) to make the player go left and is then changed to go right by swapping “left” with “right” and (KeyCode.A) to (KeyCode.D). this is all done in the Update() Function so that every frame (60 times every second) the check to see if the User is pressing that key will be checked.

### Collision

if (collision.gameObject.tag == "ball")

        {

            rigidbody.AddForce(new Vector2(0, 1000));

        }



## Ball

### Direction

### Collision

## Enemy

### Collision

# Arbeitsjournal

# Reflexionsfähikeit

# Prägnänz

# Grafiken, Bilder, Diagramme und Tabellen

# Durchführung und Auswertung der Tests

# Storyboard