

Donnerstag, 6. Julie 2023

Projektarbeit

## **Runner 96 – Computerspiel in Visual Basic**

Linus von Maltzan, Erik Siegel, Philipp Brocher

Studiengang TWE22

Dualen Hochschule Baden-Württemberg Ravensburg

## Erklärung zum Programmentwurf

gem. § 5(2) der „Studien- und Prüfungsordnung DHBW Technik“ vom 18. Mai 2009.

Hiermit erkläre ich, dass ich den vorgelegten Programmentwurf selbständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe.

_____ Ort	_____ Datum	_____ Mat.-Nr.	_____ Unterschrift
--------------	----------------	-------------------	-----------------------

_____ Ort	_____ Datum	_____ Mat.-Nr.	_____ Unterschrift
--------------	----------------	-------------------	-----------------------

_____ Ort	_____ Datum	_____ Mat.-Nr.	_____ Unterschrift
--------------	----------------	-------------------	-----------------------

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>1</b>
1.1	Aufgabe .....	1
1.2	Spielidee.....	1
<b>2</b>	<b>Anforderungskatalog .....</b>	<b>2</b>
<b>3</b>	<b>Struktogramme des Spielablaufs.....</b>	<b>2</b>
3.1	Main.....	2
3.2	SoundManager .....	2
3.3	KeyInput .....	3
3.4	MenuLoop .....	4
3.5	RoundStart.....	4
3.6	ArraySetUp .....	4
3.7	PlayerAnimator .....	4
3.8	GroundAnimator .....	5
3.9	CollisionManager.....	5
3.10	GameOver .....	5
<b>4</b>	<b>Programmierung.....</b>	<b>6</b>
4.1	Informationsquellen.....	6
4.2	Tools.....	7
<b>5</b>	<b>Programmtest.....</b>	<b>7</b>
5.1	Testpläne .....	7
5.2	Testergebnis .....	14
<b>6</b>	<b>Fazit und Ausblick .....</b>	<b>14</b>
<b>7</b>	<b>Literaturverzeichnis .....</b>	<b>16</b>

# 1 Einleitung

Diese Arbeit beschäftigt sich mit dem Entwicklungsprozess eines konsolenbasierten Spiels in Visual Basic, das sich von den beliebten Arcade-Spielen der 70er Jahre inspirieren lässt.

## 1.1 Aufgabe

Die Spieler sollen sich wie für dieses Genre typisch durch eine Vielzahl von Hindernissen manövrieren und dabei versuchen, einen hohen Punktestand zu erreichen, während die Level zunehmend anspruchsvoller werden. Um dieses Projekt erfolgreich umzusetzen, soll zuerst ein umfassender Anforderungskatalog erstellt werden auf dessen Basis später das Spiel programmiert wird. Zusätzlich sollen Struktogrammen angelegt werden, um den Ablauf des Spiels zu veranschaulichen. Um die Zuverlässigkeit und Qualität des Spiels zu gewährleisten, werden wir im letzten Schritt umfangreiche Testpläne erstellen und den Programcode auf seine Robustheit prüfen.

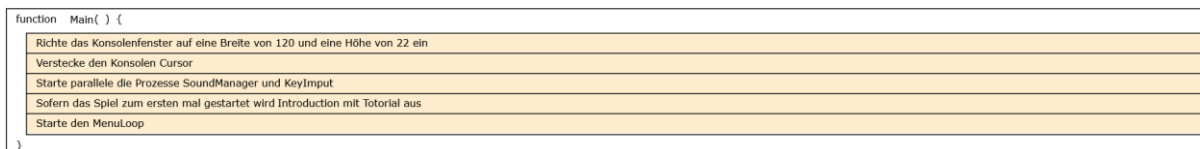
## 1.2 Spielidee

In "Runner 96" besteht das Hauptziel des Spielers darin, so weit wie möglich zu laufen. Allerdings ist er einer tückischen Welt voller Hindernissen ausgesetzt. Der Spieler kann nicht anhalten, ihm bleibt daher nur eine Möglichkeit: Springen und den Hindernissen ausweichen. Mit zunehmendem Fortschritt im Spiel werden immer mehr Hindernisse auf den Spieler zukommen und sein Reaktionsvermögen auf die Probe stellen.

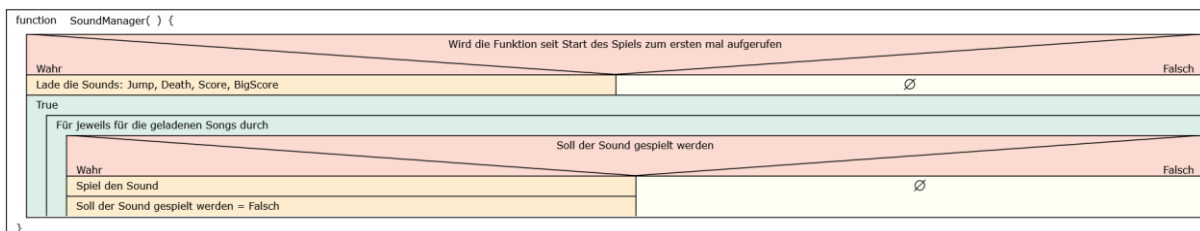
## 2 Anforderungskatalog

## 3 Struktogramme des Spielablaufs

### 3.1 Main

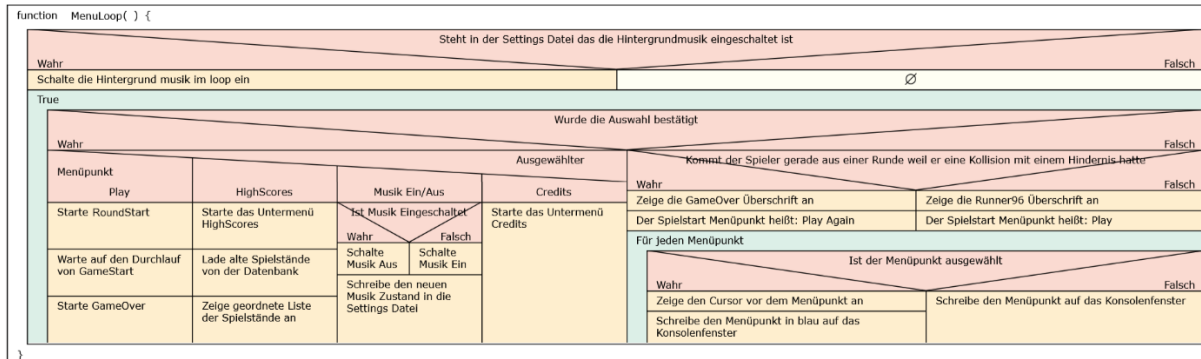


### 3.2 SoundManager

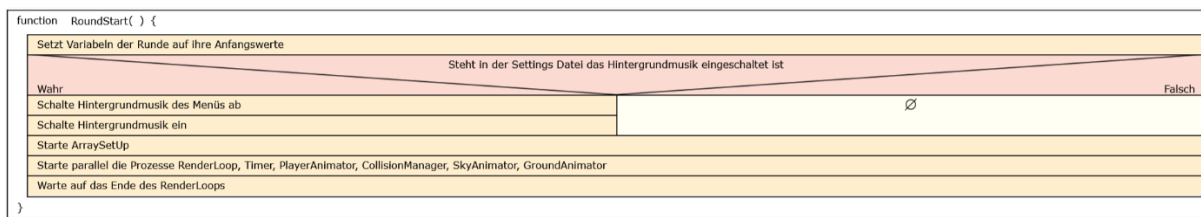




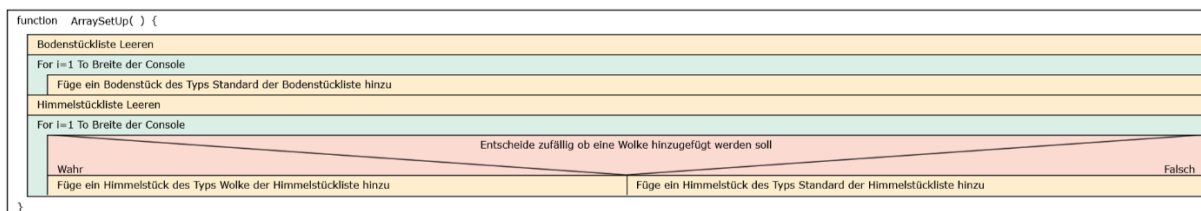
### 3.4 MenuLoop



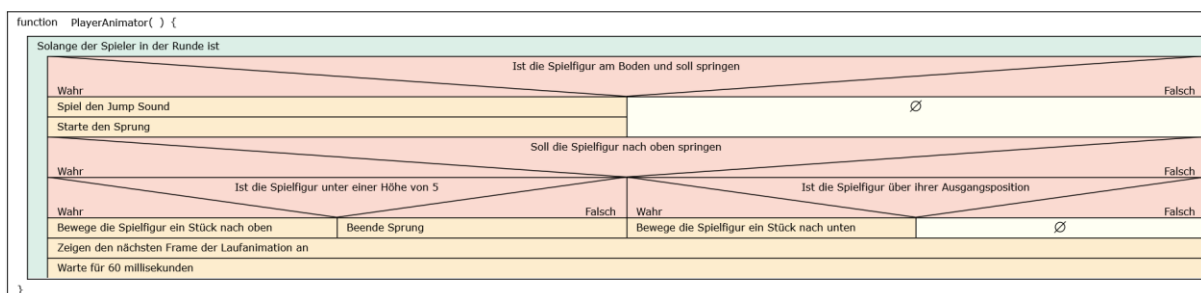
### 3.5 RoundStart



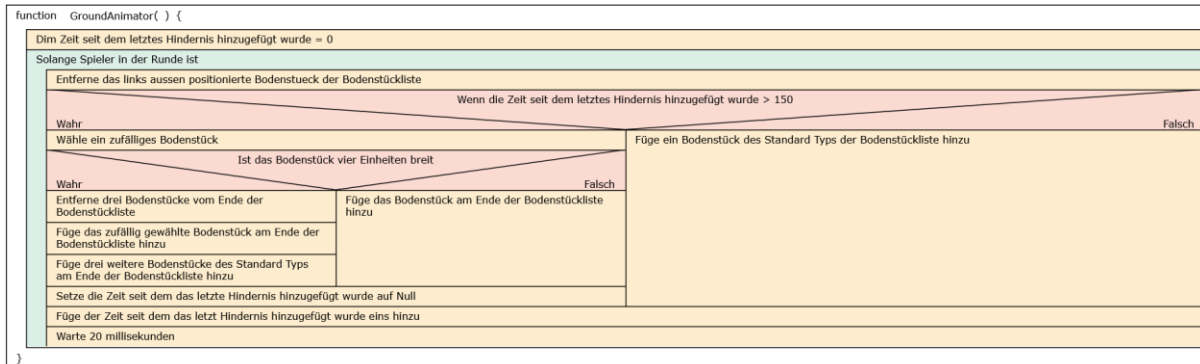
### 3.6 ArraySetUp



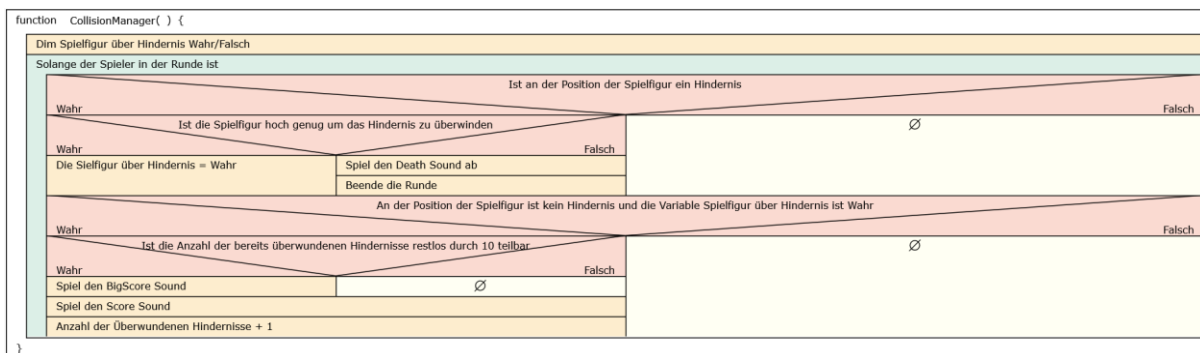
### 3.7 PlayerAnimator



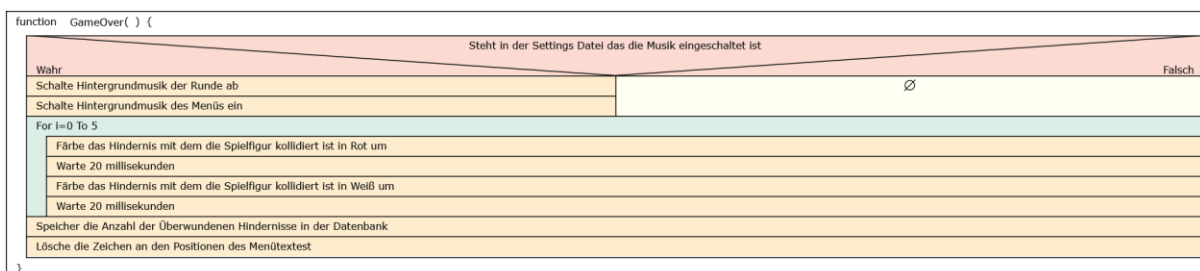
### 3.8 GroundAnimator



### 3.9 CollisionManager



### 3.10 GameOver





## 4 Programmierung

### 4.1 Informationsquellen

Visual Basic ist eine schon etwas in die Jahre gekommene Programmiersprache. Doch obwohl keiner von uns VB zuvor genutzt hatte, war uns die Programmiersprache von Anfang an vertraut. Was daran liegt, dass sie wie die uns bereits bekannte Programmiersprache C# auf dem robusten .NET Framework basiert. Das heißt jedoch nicht, dass wir ohne zu Googeln ausgekommen sind. Die offizielle Microsoft-Dokumentation [1] stellte sich als sehr gute Informationsquelle für Visual Basics heraus. Aber auch Stack Overflow, der beste Freund eines jeden Programmierers, half uns oft weiter, wenn wir nach Lösungen suchten.

Ein Feature, das uns sehr wichtig war, ist die Möglichkeit, mehrere Sounds parallel abspielen zu können, beispielsweise einen Sprungsound, ohne die Hintergrundmusik zu unterbrechen. Da diese Funktion nicht nativ in Visual Basic enthalten ist, griffen wir auf die Windows Multimedia API zurück, deren Implementierung Yang Kok Wah oder in seinem Artikel [2] sehr gut erklärt hat.

Da die Highscore-Funktion ein wichtiger Bestandteil unserer Spielidee war, planten wir von Anfang an die Möglichkeit ein, den eigenen Highscore mit dem anderer Spieler vergleichen zu können. Firebase ist ein sehr leistungsfähiges Tool für solche Aufgaben, da man mit nur wenigen Zeilen Code vollwertige Datenbanklösungen kostenlos implementieren kann. Ein sehr hilfreiches Tutorial zur Integration mit FireSharp [3] war das Video [4] von Hassaan Raza von The Amazing Codeverse.

## 4.2 Tools

Der Großteil der Entwicklung erfolgte in Visual Studio [5]. Diese leistungsstarke IDE bot uns effiziente Tools und Funktionen zum Schreiben, Debuggen und Testen des Codes.

Um effektiver im Team zusammenarbeiten zu können, haben wir von Anfang an unseren Programmcode mit GitHub [6] verbunden. Diese Integration ermöglichte es uns, gemeinsam an dem Projekt zu arbeiten und Änderungen nachzuverfolgen.

Bei der Erstellung der Struktogramme haben wir den Open-Source-Editor Struktog [7] genutzt, der von der TU Dresden entwickelt wurde. In Struktog lassen sich die Diagramme nicht nur als Bild zu speichern, sondern auch als JSON-Datei. Dadurch konnten die Struktogramme im weiteren Verlauf des Projekts problemlos angepasst werden, was ein großer Vorteil war.

# 5 Programmtest

## 5.1 Testpläne

Nr.	Bezeichnung	Nr.	Durchführung	Erwartetes Ergebnis	X
1	Start der Software	1	Software starten	Software startet	X
2	Startmenüs	1	Drücken der Enter Taste im Startbildschirm	Startmenü 1 öffnet sich	X
		2	Im Startmenü 1 die Enter/Leertaste drücken	Startmenü 1 schließt sich und Startmenü 2 öffnet sich	X

		3	Im Startmenü 2 die Enter/Leertaste drücken	Startmenü 2 schließt sich und Startmenü 3 öffnet sich	X
		4	Startmenü 3 öffnet sich	Startmenü 3 fordert Spieler auf einen Namen einzugeben	X
		5	Eingabe des Namens im Startmenü 3	Eingegebener Name wird Textfeld angezeigt	X
		6	Kein Name wird eingegeben und Enter/Leertaste wird gedrückt	Spieler wird automatisch „PLAYER 1“ genannt	X
		7	Drücken der Enter Taste	Name wird gespeichert und „Continue“ links oben blau hinterlegt	X
		8	Drücken der Enter Taste im Startmenü 3	Startmenü 3 wird geschlossen und Startmenü 4/ Hauptmenü öffnet sich	X
		9	Menüpunkt „Play“ ist im Startmenü 4/ Hauptmenü vorausgewählt	Play wird in Blau angezeigt	X
3	Spielstart	1	Drücken der Enter Taste mit vorausgewähltem Menüpunkt „Play“	Spiel wird gestartet	X
4	Bewegung Spielfigur	1	Spiel starten ohne drücken jeglicher Tasten	Spielfigur läuft unverändert auf derselben Stelle weiter	X
		2	Kurzes Drücken der Leertaste (ca. 1s)	Spielfigur „springt“ einmalig um 5 Zeilen nach oben	X
		3	Loslassen der Leertaste nach dem Sprung	Spielfigur landet und läuft weiter	X
		4	Gedrückt halten der Leertaste über	Spielfigur „springt“ bei Bodenkontakt (0,1 Zeile vor dem Boden) erneut	X

			beliebigen Zeitraum >1s	und wiederholt dies so lang bis Leertaste losgelassen wird	
5	Hindernisse	1	Spiel starten	Hindernisse (maximal 2 Spieleinheiten 2 breit und hoch) erscheinen zufällig am rechten Spielfeldrand	X
		2	Spiel starten/laufen lassen	Erschienenene Hindernisse bewegen sich von rechts nach links auf Spielfigur zu	X
		3	Spiel starten/laufen lassen	Überwundene Hindernisse verschwinden hinter dem linken Spielfeldrand	X
		4	Spiel lang laufen lassen	Hindernisdichte nimmt zu (Maximal 50 Hindernisse auf der Spieloberfläche)	X
		5	Spiel lang laufen lassen	Hindernisgeschwindigkeit nimmt zu	X
		6	Spiel laufen lassen	Hindernisse werden ausschließlich am rechten Spielfeldrand erzeugt	X
		7	Drücken der Leertaste vor einem Hindernis	Überspringen des Hindernisses	X
6	Kollisionserkennung	1	Spiel laufen lassen ohne drücken jeglicher Tasten	Spielfigur kollidiert mit Hindernis (nur wenn Spielfigur am Boden ist, während sich ein Hindernis in Spalte 6 befindet) → Hindernis das Kollision verursacht hat wird rot markiert	X
		2	Spiel laufen lassen ohne drücken jeglicher Tasten	Nach Kollision wechselt Spiel in Game-Over Bildschirm/Hauptmenü	X

7	Hauptmenü	1	Drücken der Pfeiltaste unten	Menüpunkt unterhalb derzeit ausgewähltem Menüpunkt wird ausgewählt und in blau angezeigt	X
		2	Drücken der Pfeiltaste unten	Menüpunkt oberhalb derzeit ausgewähltem Menüpunkt wird ausgewählt und in blau angezeigt	X
		3	Menüpunkt „Play Again“ auswählen und mit Enter/Leertaste bestätigen	Spiel wird erneut gestartet	X
		4	Menüpunkt „High Scores“ auswählen und mit Enter/Leertaste bestätigen (mit Internet)	High-Score Liste öffnet sich und zeigt die bisher höchst erzielten Ergebnisse (sofern man mit dem Internet verbunden ist)	X
		5	Menüpunkt „High Scores“ auswählen und mit Enter/Leertaste bestätigen (ohne Internet)	High Score Listen Menü öffnet sich und zeigt Fehlermeldung „- UNABLE TO CONNECT TO DATABASE -“, an	X
		6	In High-Score Liste Enter/Leertaste drücken	High-Score Liste wird geschlossen und es öffnet sich das Hauptmenü	X
		7	Menüpunkt „Change Name“ auswählen	Menüpunkt wird in blu angezeigt	X
		8	Menüpunkt Change Name mit Enter/Leertaste bestätigen	Change Name Menü wird aufgerufen	X
		9	Aufrufen des Change Name Menüs	Bisher verwendeter Name erscheint im Textfeld	X

		10	Eingeben des neuen Namens	Alter Name kann durch einfache Eingabe des neuen Namens überschrieben werden	X
		11	Drücken der Rücktaste im Change Game Menü	Einzelnes Zeichen des bisherigen Namens wird entfernt und der Cursor bewegt sich um eine Spalte nach links	X
		12	Erneutes Drücken der Rücktaste im Change Game Menü	Weiteres Zeichen des bisherigen Namens wird entfernt und Cursor bewegt sich um eine Spalte nach Links	X
		13	Drücken der Entfernen Taste	Der ausgewählte Buchstabe wird entfernt und der Cursor bleibt an der gleichen Stelle	X
		14	Drücken der Enter/Leertaste	Name wird gespeichert, das Change Name Menü beendet und das Hauptmenü geöffnet	X
		15	Bestätigen des leeren Textfeldes im Change Name Menü, ohne einen neuen Namen eingegeben zu haben	Ungültige Eingabe → alter Name wird weiterhin verwendet	X
		16	Auswählen des Menüpunktes „Music Enabled/Disabled“	Menüpunkt wird in blau angezeigt	X
		17	Drücken der Enter/Leertaste bei ausgewähltem Menüpunkt „Music Enabled“	Spielhintergrundmusik wird ausgeschaltet und Menüpunkt wechselt von „Music Enabled“ zu „Music Disabled“	X
		18	Drücken der Enter/Leertaste bei ausgewähltem	Spielhintergrundmusik wird eingeschaltet und Menüpunkt wechselt von	X

			Menüpunkt „Music Disabled“	„Music Disbled“ zu „Music Enabled“	
		19	Auswählen des Menüpunktes „Credits“	Ausgewählter Menüpunkt wird in blau angezeigt	X
		20	Drücken der Enter/Leertaste bei ausgewähltem Menüpunkt „Credits“	„Credits Menü“ wird aufgerufen und zeigt Ersteller und Mitwirkende Personen	X
		21	Drücken der Enter/Leertaste	„Credits Menü“ wird geschlossen und Hauptmenü aufgerufen	X
8	Sounds	1	Aufrufen der Software	Ein kurzer Sound wird beim Starten des Spiels abgespielt	X
		2	Aufrufen der Software	Die Spielmusik beginnt direkt im Anschluss zum Startsound zu spielen und läuft dauerhaft weiter	X
		3	Drücken der Leertaste im Spielverlauf	Ein spezifisch auf den Sprung abgestimmter Sound wird abgespielt	X
		4	Überspringen von 10 Hindernissen	Ein Sound wird alle 10 „Scorepunkte“ zusätzlich zum Sprung-Sound abgespielt	X
		5	Kollision mit einem Hindernis	Verzerrter Sound wird abgespielt	X
9	Sonstige Funktionen	1	Spiel starten	Score wird in der oberen linken Ecke zum Spielbeginn als „0“ angezeigt	X
		2	Spiel starten	Timer wird in der linken oberen Ecke unterhalb des Scores zum Spielbeginn als „0“ angezeigt und zählt die Sekunden hoch	X

		3	Spiel starten	Anzahl der „FPS“ wird angezeigt	X
		4	Überwinden eines Hindernis	Der Score wird um eins erhöht	X
		5	Spiel laufen lassen	Der Timer zählt die Sekunden aufwärts	X
		6	Überwinden mehrerer Hindernisse im Spielverlauf	Der Score zählt aufwärts hoch	X
		7	Game Over durch Kollision mit einem Hindernis	Timer und Highscore hören auf zu zählen	X
		8	Spiel starten	Im oberen Bereich des Spielfeldes wird eine zufällige Anzahl an Wolken eingebendet	X
		9	Spiel laufen lassen	Neue Wolken erscheinen zufällig auf der rechten Bildschirmseite	X
		10	Spiel laufen lassen	Wolken bewegen sich im Verlauf des Spiels nach links und verschwinden aus dem linken Spielfeldrand	X



## 5.2 Testergebnis

Die Testergebnisse haben wertvolle Erkenntnisse geliefert und Bereiche aufgezeigt, in denen weitere Verbesserungen vorgenommen werden können. Das Programm läuft trotz seiner Komplexität sehr stabil, insbesondere in Bezug auf die Tastatureingabe. Die Hinderniserkennung funktioniert in der Regel zuverlässig, könnte jedoch durch zusätzliche Feinabstimmungen weiter verbessert werden, insbesondere bei der Erkennung von Hindernissen von oben. Außerdem bietet die Interaktion mit dem Konsolenfenster, Optimierungsspielraum. Die Anpassung der Fenstergröße kann zu visuellen Unstimmigkeiten führen, und bei minimieren und maximieren des Fensters wird der Cursor sichtbar. Des Weiteren können bei der Ausführung des Programms als eigenständige Executable außerhalb der IDE, Anzeigeprobleme auftreten. Die Renderfähigkeiten des Programms sind im Allgemeinen zufriedenstellend, können jedoch bei schnellen Bewegungen, zu Anzeigefehlern führen. Darüber hinaus kann es auf bestimmten Geräten Einschränkungen bei der gleichzeitigen Soundwiedergabe geben, und die Wiedergabeflüssigkeit des Spiels kann durch unzureichende Rechenressourcen Verzögerungen erfahren.

## 6 Fazit und Ausblick

Zusammenfassend lässt sich sagen, dass sich unsere Zusammenarbeit in der Gruppe als sehr effektiv erwiesen und zu einem zufriedenstellenden Ergebnis geführt hat.

Obwohl wir unsere Ziele mit Visual Basic erreicht haben, wäre es interessant gewesen, die Möglichkeiten von fortschrittlicheren Programmierumgebungen wie Unity oder Unreal Engine zu erkunden. Mit solchen Game Engines lassen sich oft in kürzester Zeit anspruchsvolle Spiele entwickeln, weil sie arbeitsintensive

Aufgaben übernehmen und so auch dem leihen das erstellen von komplexen Anwendungen ermöglichen.

Mit Blick auf die Zukunft eröffnen sich zahlreiche Möglichkeiten, um das Spiel weiter auszubauen. Es wäre naheliegend, die in den Testergebnissen aufgeführten Probleme zu beheben und somit die Stabilität und Zuverlässigkeit des Spiels weiter zu verbessern. Zusätzlich könnte man Features wie ein Münzsystem einführen, bei dem die Spieler durch das Sammeln von Münzen Upgrades und Power-Ups erwerben können. Oder die Implementierung von benutzerspezifischen Profilen, welche den Spielern ermöglichen würde, ihren Spielfortschritt zu speichern und personalisierte Einstellungen beizubehalten.

Zum Schluss bleibt nur noch zu sagen: Schnür' deine Laufschuhe, meistere die Hindernisse und renne, als gäbe es kein Morgen! Hol dir den Highscore und lass die Hindernisse alt aussehen.

## 7 Literaturverzeichnis

- [1] Microsoft, „Microsoft,“ [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/visual-basic/>. [Zugriff am 24 Mai 2023].
- [2] Y. K. Wah, „Codeproject,“ 4 Juni 2014. [Online]. Available: <https://www.codeproject.com/Articles/781792/Fun-with-Sound>. [Zugriff am 8 Juni 2023].
- [3] lucasratske, „<https://github.com/ziyasal/FireSharp>,“ 26 Januar 2023. [Online]. Available: <https://github.com/ziyasal/FireSharp>. [Zugriff am 8 Juni 2023].
- [4] H. Raza, „Youtube,“ 9 Januar 2020. [Online]. Available: <https://youtu.be/mK2MErhxIO0>. [Zugriff am 7 Juni 2023].
- [5] Microsoft, „Visualstudio,“ [Online]. Available: <https://visualstudio.microsoft.com/>. [Zugriff am 19 April 2023].
- [6] GitHub, „GitHub,“ [Online]. Available: <https://desktop.github.com/>. [Zugriff am 19 April 2023].
- [7] T. Leonhardt, „GitLab,“ 23 Mai 2023. [Online]. Available: <https://gitlab.com/ddi-tu-dresden/cs-school-tools/struktog>. [Zugriff am 9 Juni 2023].