

IHK Abschlussprüfung Teil 2 – Sommer 2025	 Industrie- und Handelskammer zu Rostock
IT - Berufe Deckblatt für die Dokumentation zur Betrieblichen Projektarbeit	Ausbildungsberuf: Bitte Beruf auswählen:

Titel der Betrieblichen Projektarbeit: Simulationsprojekt: „Murmelsimulation – Wie beeinflussen physikalische Parameter die Murmel?“
--

Prüfungsteilnehmer/in	Ausbildungs-/ Praktikumsunternehmen:
Vor- und Familienname: Benjamin Seidel Anschrift des Prüfungsteilnehmers: OT: Klicken Sie hier, um den Ortsteil einzugeben. Straße: Friedrich-Wolf-Straße 27 PLZ Ort 18435 Telefon: -491729289623 E-Mail: benjamin.159@web.de	Verantwortliche/r für die Durchführung des betrieblichen Auftrages: Klicken Sie hier, um Vor- und Zunamen des Ansprechpartners im Unternehmen einzugeben. Anschrift des Unternehmens: OT: Klicken Sie hier, um den Ortsteil einzugeben. Straße: Klicken Sie hier, um die Straße einzugeben. PLZ Ort Klicken Sie hier, um Postleitzahl und Ort einzugeben. Telefon: Klicken Sie hier, um die Telefonnr. des Ansprechpartners einzugeben. E-Mail: Klicken Sie hier, um die E-Mail-Adresse des Ansprechpartners einzugeben.

_____	Klicken Sie hier, um ein Datum einzugeben. _____	_____
Ort	Datum	Unterschrift Antragsteller/-in (Prüfling)
_____	Klicken Sie hier, um ein Datum einzugeben. _____	_____
Ort	Datum	Stempel/ Unterschrift Verantwortliche/r für den betrieblichen Auftrag

Inhaltsverzeichnis

Abkürzungsverzeichnis.....	1
1. Einleitung.....	2
1.1. Ausgangssituation und Problemstellung.....	2
1.2. Projektziele.....	2
1.3. Rahmenbedingungen und Abgrenzungen.....	3
1.4. Anforderungen an die Anwendung.....	4
1.4.1 Funktionale Anforderungen.....	4
1.4.2 Nicht-funktionale Anforderungen.....	4
2. Projektplanung.....	5
2.1. Vorgehensmodell.....	5
2.2. Ressourcen- und Ablaufplanung.....	5
2.2.1. Zeitplanung.....	5
2.2.2. Kostenplanung.....	5
2.3. Risikoanalyse.....	5
3. Technische Umsetzung.....	6
3.1. Architektur und Design.....	6
3.1.1. Architektur-Konzept (MVC).....	6
3.1.2. Datenauswertung.....	6
3.2. Auswahl der Technologien.....	6
3.2.1. Kriterien für die Nutzwertanalyse.....	6
3.2.2. Durchführung der Nutzwertanalyse.....	6
3.2.3. Begründung der Entscheidung.....	7
3.3. Implementierung.....	8
3.3.1. Kernkomponenten.....	8
3.3.2. Zufallselemente und Verteilung.....	8
3.4. Benutzeroberfläche.....	8
3.4.1. UI-Konzept und Benutzbarkeit.....	8
3.4.2. Elemente und Animation.....	8
4. Qualitätssicherung.....	9
4.1. Testkonzept.....	9
4.2. Testdurchführung.....	9
4.3. Validierung der Zufallsmodelle.....	9
4.4. Soll-Ist-Vergleich.....	9
5. Zusammenfassung und Ausblick.....	10
5.1. Fazit.....	10
5.2. Ausblick.....	10

Literaturverzeichnis.....	11
Anhang.....	i



Abkürzungsverzeichnis



1. Einleitung

1.1. Ausgangssituation und Problemstellung

Das hier dokumentierte Projekt „Gravity Rollers“ ist die vorgegebene Projektarbeit für das dritte Ausbildungsjahr. Es dient als praktische Übung und Generalprobe für die bevorstehende IHK-Abschlussprüfung, indem es die Erstellung einer vollständigen Anwendung nach IHK-Standards simuliert.

Das Kernziel der Anwendung ist die Entwicklung einer physikbasierten, zeitabhängigen Simulation. Dem Anwender soll ein Werkzeug bereitgestellt werden, um die Auswirkungen von konfigurierbaren physikalischen Parametern und definierten Zufallsfaktoren auf einer Murmelbahn visuell zu analysieren und auszuwerten.

Die Realisierung erfolgt als Windows-Anwendung mit grafischer Benutzeroberfläche. Eine wesentliche technische Herausforderung ergibt sich aus der Anforderung, die Lauffähigkeit auf den Schulrechnern als Zielsystem zu gewährleisten, was eine optimierte Implementierung der Echtzeit-Physikberechnungen erfordert.

1.2. Projektziele

Die primären Projektziele umfassen die Entwicklung einer intuitiven und benutzerfreundlichen grafischen Benutzeroberfläche, die sich an den Interaktionsprinzipien der ISO 9241-110 orientiert.

Ebenso zentral ist die nachvollziehbare visuelle Darstellung der Simulation. Der Anwender muss die Auswirkungen der individuell konfigurierten physikalischen Parameter sowie der implementierten Zufallsereignisse auf das Endergebnis analysieren und auswerten können.

Die Abgabe der Anwendung erfolgt als einzelne, auf Windows lauffähige, ausführbare Datei.



1.3. Rahmenbedingungen und Abgrenzungen

Die methodischen und technischen Rahmenbedingungen wurden getroffen, um die Projektziele effizient und im vorgegebenen Zeitrahmen zu erreichen.

Vorgehensmodell: Als Vorgehensmodell wurde das Wasserfallmodell gewählt. Die Anforderungen des Projektauftrags waren von Beginn an klar definiert und fest vorgegeben. Dieses Modell eignet sich ideal für Projekte mit festem Anforderungskatalog, da es eine strukturierte Abarbeitung der Phasen ermöglicht.

Technologie: Die Projektanforderung, die Auswirkungen physikalischer Parameter visuell darzustellen und auszuwerten, legte die Verwendung einer 3D-Simulation nahe. Eine 3D-Darstellung bietet im Gegensatz zu einer 2D-Simulation oder reinen Datenblättern eine intuitivere Erfassung der komplexen räumlichen Interaktionen, was für die Analyse der Ergebnisse unerlässlich ist.

Auf Basis dieser Entscheidung wurde die technische Grundlage auf die Unreal Engine 5 gelegt, da die integrierte Chaos Physics Engine eine robuste und realistische Physiks simulation "out-of-the-box" bietet. Als valide Option wurde auch die Unity-Engine evaluiert. Die Entscheidung fiel jedoch zugunsten von Unreal Engine, da die native C++-Schnittstelle die performante Umsetzung der geforderten Physiks simulation auf der Zielhardware besser unterstützt als C# in Unity.

Architektur: Aufbauend auf der Engine-Wahl wird ein hybrider Ansatz verfolgt. Diese Architektur nutzt die Stärken beider Paradigmen:

1. **C++:** Die performancekritische Kernlogik der Murmel, inklusive der dynamischen Anwendung physikalischer Parameter, wird in C++ umgesetzt, um maximale Rechenleistung zu gewährleisten.
2. **Blueprints:** Die Benutzeroberfläche und die Ablaufsteuerung werden in Blueprints realisiert. Dies ermöglicht eine schnelle, iterative Entwicklung und Anpassung der visuellen Elemente und der Benutzerinteraktion.



Zielsystem und Abgrenzung: Die Anwendung wird als 64-Bit-Windows-Anwendung entwickelt. Die Lauffähigkeit muss auf den Schulrechnern gewährleistet sein, was den Einsatz optimierter Assets begründet.

Aufgrund des festen Abgabetermins am 06.02.2025 wird der Projektumfang klar abgegrenzt. Es wird kein Speichersystem, kein Level-Editor und kein Mehrspielermodus implementiert. Der Fokus liegt ausschließlich auf der Konfiguration, Simulation und Auswertung eines einzelnen oder mehreren Renndurchläufe.

1.4. Anforderungen an die Anwendung

1.4.1 Funktionale Anforderungen

1.4.2 Nicht-funktionale Anforderungen



2. Projektplanung

2.1. Vorgehensmodell

2.2. Ressourcen- und Ablaufplanung

2.2.1. Zeitplanung

2.2.2. Kostenplanung

2.3. Risikoanalyse



3. Technische Umsetzung

3.1. Architektur und Design

3.1.1. Architektur-Konzept (MVC)

3.1.2. Datenauswertung

3.2. Auswahl der Technologien

Die Wahl der geeigneten Entwicklungsumgebung und der Werkzeuge ist entscheidend, um das Projekt im vorgegebenen Zeitrahmen und mit der geforderten Qualität umzusetzen.

3.2.1. Kriterien für die Nutzwertanalyse

Für die objektive Auswahl wurden folgende Kriterien definiert und gewichtet (Skala 1–4):

1. Physik-Engine (Gewichtung: 4):

Da es sich um eine physikalische Simulation handelt, ist die Qualität und Stabilität der Physik-Engine das wichtigste Kriterium.

2. GUI-Framework (Gewichtung: 3):

Die Anforderung einer ISO-konformen Benutzeroberfläche zur Steuerung und Auswertung erfordert leistungsfähige, integrierte UI-Werkzeuge.

3. Programmiersprache (Gewichtung: 2):

Die Unterstützung einer performanten Sprache ist wichtig für die Optimierung auf Schulrechnern, wird aber niedriger gewichtet als die Kernfeatures der Engine.

4. Dokumentation & Community (Gewichtung: 2):

Wichtig für die Problemlösung während der Entwicklung.

5. Einarbeitungszeit (Gewichtung: 1):

Da die technische Eignung im Vordergrund steht, wird der Einarbeitungsaufwand als weniger kritisch eingestuft.

3.2.2. Durchführung der Nutzwertanalyse

Verglichen wurden die Engines, Unreal Engine 5, Unity und Godot. Die Bewertung erfolgt auf einer Skala von 1 (ungenügend) bis 3 (sehr gut).



Kriterium	Gewichtung	Unreal Engine	Unity	Godot	Begründung
Physik-Engine	4	3 (Chaos Physics)	2 (PhysX)	1 (Basis)	Unreal bietet mit Chaos Physics die robusteste integrierte Lösung für komplexe Interaktionen.
GUI-Framework	3	3 (UMG)	2 (UI Toolkit)	2 (Control Nodes)	Das Unreal Motion Graphics Framework bietet einen visuellen Designer, der ideal für Menüs und HUDs ist.
Programmiersprache	2	3 (C++ / Blueprint)	1 (C#)	2 (C++/ GDScript)	Der Hybrid-Ansatz vereint maximale Performance mit schneller Iteration.
Dokumentation & Community	2	3 (Sehr gut)	3 (Sehr gut)	2 (Gut)	Exzellente offizielle Dokumentation und riesige Community bei Unreal.
Einarbeitungszeit	1	3 (Vorwissen)	2 (Basis)	1 (Keine)	Vorkenntnisse sind vorhanden.
Gesamtnutzwert		36	26	19	

(Berechnung: Summe aus [Bewertung x Gewichtung])

3.2.3. Begründung der Entscheidung

Basierend auf der Nutzwertanalyse fiel die Entscheidung eindeutig auf die Unreal Engine 5. Neben den persönlichen Vorkenntnissen war die Eignung der Programmiersprachen-Architektur ausschlaggebend, sowie auch die bereitgestellte Physik-Engine. Unreal ermöglicht einen hybriden Entwicklungsansatz, der die spezifischen Anforderungen des Projekts optimal abdeckt:

1. C++ für Performance:

Die performancekritische Kernlogik der Murmel (Physikberechnungen, Parameter-Updates) wird in C++ implementiert. Dies gewährleistet die geforderte Lauffähigkeit auf den leistungsschwächeren Schulrechnern.

2. Blueprints für Effizienz:

Die Entwicklung der GUI und der Logik der Murmel wird über Blueprints realisiert, was die Entwicklung erleichtert und die Wartung vereinfacht.



Für die Benutzeroberfläche (UI) und High-Level-Logik wird das visuelle Scripting-System "Blueprints" verwendet. Dies ermöglicht eine extrem schnelle Entwicklung und Anpassung der GUI-Elemente (Slider, Diagramme), was in reinem C++ unverhältnismäßig aufwendig wäre.

Dieser Mix aus Hardwarenähe (C++) und Entwicklungseffizienz (Blueprints) ist das Alleinstellungsmerkmal, das zur Wahl der Unreal Engine führte.

3.3. Implementierung

3.3.1. Kernkomponenten

3.3.2. Zufallselemente und Verteilung

3.4. Benutzeroberfläche

3.4.1. UI-Konzept und Benutzbarkeit

3.4.2. Elemente und Animation



4. Qualitätssicherung

4.1. Testkonzept

4.2. Testdurchführung

4.3. Validierung der Zufallsmodelle

4.4. Soll-Ist-Vergleich



5. Zusammenfassung und Ausblick

5.1. Fazit

5.2. Ausblick



Literaturverzeichnis

Anhang