

Inhaltsverzeichnis

1. Technische Umsetzung.....	2
1.1. Auswahl der Technologien.....	2
1.1.1. Kriterien für die Nutzwertanalyse.....	2
1.1.2. Durchführung der Nutzwertanalyse.....	2
1.1.3. Begründung der Entscheidung.....	2



1. Technische Umsetzung

1.1. 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.

1.1.1. 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.

1.1.2. 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])

1.1.3. 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:

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.