



INTERNATIONALE  
HOCHSCHULE

## **Abstract**

**DLMCSPSE01\_D**

**Projekt: Software Engineering**

**2D Spiel „SAND“**

<b>Erstellt von:</b>	Keanu Semmel
<b>Matrikelnummer:</b>	32005464
<b>Studiengang:</b>	Informatik (M.Sc.)
<b>Tutor:</b>	Herr Dr. Markus Kleffmann
<b>Datum:</b>	23.12.2023

## Abstract

Das Indie-Spielprojekt "SAND" wurde mithilfe der Pygame-Bibliothek in Python entwickelt. Die ursprüngliche Schätzung für die Fertigstellung betrug 14 Tage, wurde jedoch aus Sicherheitsgründen beziehungsweise zur Sicherstellung eines ausreichenden Puffers bei Aufkommen von Schwierigkeiten auf 21 Tage verlängert. Hierbei lag der Fokus auf einer effizienten Nutzung der vorgesehenen Zeiträume der jeweiligen Entwicklungsbeziehungsweise Projektphasen. Das Projekt setzt auf die Implementierung von Features wie Pfadfinding in Form des A\*-Algorithmus, einer intuitiven Benutzeroberfläche (UI) und vorteilhaft durchdachte Spielmechaniken. Die Entscheidung zur Verlängerung der Projektlaufzeit erwies sich als vorteilhaft, da hiermit ausreichend Pufferzeit für unvorhergesehene Herausforderungen geschaffen wurde und die Gesamtstabilität des Projekts somit verbessert wurde. Grundsätzlich kann festgehalten werden, dass die ursprüngliche Schätzung, welche sich auf 14 Tage belief, ausreichend war, jedoch der Puffer eine zusätzliche Sicherheit bot. Insofern im Rahmen der Implementierung Schwierigkeiten und somit daraus resultierend Verzögerungen eingetreten wären, wäre ein Einhalten des Projektplans als nicht möglich anzusehen gewesen. Entsprechend dessen gilt es festzuhalten, dass ein Projektplan einen entsprechenden Puffer berücksichtigen beziehungsweise im Rahmen der Aufwandschätzung eine Abweichung berücksichtigt werden muss.

Die Implementierung des A\*-Algorithmus für das Pfadfinding wurde während der Entwicklung in die Logik integriert, was aufgrund der verzögerten beziehungsweise nachträglichen Umsetzung eine Herausforderung darstellte. Eine Schlüsselerkenntnis daraus, besteht darin, dass ein solch komplexer Algorithmus direkt in die anfängliche Spiellogik eingebunden werden sollte. Mittels einer frühzeitigen Implementierung kann sichergestellt werden, dass die jeweils genutzte Third-Party-Library optimal implementiert wird. Somit kann auf eine aufwändige beziehungsweise nachträgliche Anpassung der Logik und ihre verwendeten Punkte verzichtet werden.

Positive Aspekte, welche es zu benennen gilt, umfassen die reibungslose Implementierung der Benutzeroberfläche, welche eine effektive und ansprechende UI hervorbrachte, und die musikalische Abgrenzung des Menüs, des Spiels sowie der Game-Over Sequenz.

Eine ausführliche Test-Phase in Form von User-Tests stellt die korrekte und gewünschte Funktionsweise der festgehaltenen Anforderungen fest. Mittels einer ausführlichen Dokumentation der Anforderungen zu Beginn des Projekts, war es im Nachgang, sprich im Rahmen der Testung, möglich, transparent die Testergebnisse mit den Anforderungen abzugleichen. Hiermit war es möglich die Qualität des Spiels zu gewährleisten und gleichzeitig Rahmenbedingungen einzuhalten.

Ein Weiterer wichtiger Aspekt bestand in der ausführlichen und nachvollziehbaren Dokumentation des Spiels. Mittels einer korrekten und vollständigen Dokumentation ist es nun im Nachgang möglich die Funktionsweise der Software auf effizienten Wegen nachzuvollziehen. Hiermit wird dafür Sorge getragen, dass etwaige Anpassungen und/oder Wartungen im Nachgang von differenzierten Entwicklern durchgeführt werden können.

Alles in allem kann festgehalten werden, dass das Projekt in einem effizienten Rahmen durchgeführt werden konnte. Dies geht darauf zurück, dass im Vorfeld mittels Recherchen etwaige Konzepte, Technologien sowie Vorgehensweisen ausgearbeitet wurden. Des Weiteren gilt es festzuhalten, dass bei der Verwendung von etwaigen komplexen Algorithmen eine rechtzeitige/frühzeitige Implementierung vorgenommen werden muss, zwecks Vermeidung eines Mehraufwandes im Rahmen der Implementierung der Spiellogik.