Technical Design Document

**Volamus**

Team: Lena Spitz (Teamleiterin), Lars Haider, Mareen Allgaier

# Project Overview

## Team

Das Spiel Volamus wird von einem Team von drei Personen entwickelt:  
- Lena Spitz (Teamleiterin)  
- Lars Haider  
- Mareen Allgaier

## Game Concept

Einige Gelehrte der Acagamics haben sich das Ziel gesetzt die legendäre goldene Wolke zu finden, um sie zu studieren. Für dieses gewagte Unternehmen suchen sie aus den Delfinen, Pinguinen und den Hummel den jeweils Besten. Diese drei sollen mithilfe eines Ballspielturniers ermittelt werden.  
In dem Spiel spielen zwei Spieler gegeneinander. Dabei spielen sie die jeweiligen Charaktere des Turniers.

## Technical Goals

Ziel ist, dass das fertige Spiel eine vollständige und ästhetische 3D Graphik enthält. Auch während dem Spielen sollen die Animationen der Bewegungen und Schläge flüssig und passend sein. Die Motivation der Spieler wird währenddessen sowohl durch visuelle als auch auditive Feedbacks gesteigert.   
Um den Spielern das Treffen des Balls zu vereinfachen, ist am Boden der Schatten des Balls zu sehen. Wichtig beim Schlagen ist, dass es gelingt, eine realistische Flugbahn des Balls zu simulieren, indem die Gravitation und Begrenzungen berücksichtigt werden.

Weiterhin muss das Spiel ein selbst-updatendes Interface beinhalten.

## Technical Risks

Da kein Teammitglied über Vorwissen in der Spieleprogrammierung verfügt, können während der Entwicklung häufiger Schwierigkeiten auftreten.

Schwierig wird es, die technischen Abläufe genau zu simulieren, sodass das Spiel realistisch und korrekt läuft. Dazu zählen insbesondere die Kollisionen der Spielcharaktere mit dem Ball, sowie richtig berechnete Flugbahnen des Balls. Außerdem muss darauf geachtet werden, dass die Bewegungen richtig skaliert sind. Hier ist es wichtig den richtigen Mittelweg zwischen zu schnell und zu langsam zu finden, aber auch die Schrittgröße passend einzustellen.

Schwierigkeiten bei der Graphik und dem Feedback werden vor allem bei den Animationen (insbesondere des Wassers) auftreten und bei der zeitlich genauen Zuweisung des entsprechenden Sounds zu den jeweiligen (Bewegungs-)Animationen.

## Development Environment

Die Implementierung des Spiels findet in der Entwicklungsumgebung Visual Studio mit dem Framework Monogame statt. Es wird sowohl Windows 7 als auch Windows 10 genutzt.

## System

* + Windows Application
  + Requirements:
    - Von Monogame
    - DirectX11 und höher oder OpelGL 3.0
    - Windows XP oder höher
    - Visual C# 2010 oder höher
    - Grafikkarte

# Game Play

## Game Structure/Game Objects

Im Endspiel sollen die Spieler die Möglichkeit haben, drei verschiedene Charaktere zu spielen. Der Fokus liegt zunächst bei der Implementierung des Pinguins, der über den größten Schlagradius verfügt. Weiterhin werden dann ein schneller Delfin und eine Hummel, die am höchsten springen kann, hinzugefügt.

Weitere Personen, wie der Schiedsrichter und die Zuschauer, werden als Acagamics-Männchen dargestellt.

Zusätzlich werden Modelle für den Ball, das Spielfeld und seine Begrenzung, sowie das Netz benötigt.

## Physics

Bei der Implementierung des Spiels müssen drei physikalische Eigenschaften berücksichtigt werden.

Die Gravitation spielt bei der Flugbahn des Balls eine große Rolle. Hierfür wird die Wurfparabel, jedoch mit Vernachlässigung des Luftwiderstandes, verwendet. Auch beim Springen der Charaktere muss Gravitation beachtet werden. Hierbei wird das Springen auf einen senkrechten Sprung auf der Stelle beschränkt.

Laut des physikalischen Gesetzes, kann an einer Stelle, an der ein Körper ist, kein zweiter sein. Deshalb gibt es einige Kollisionen, die beachtet werden müssen. Für das Spiel relevant sind jegliche Kollisionen der Spieler mit Ball, Feld und Netz. Der Ball wiederum kann mit Spielern, dem Netz, dem Boden und den Zuschauern, beziehungsweise einer Barriere vor den Zuschauern, kollidieren.

Zusätzlich haben die jeweils unterschiedlichen Charaktere und der Ball individuelle Geschwindigkeiten, die so aufeinander abgestimmt werden müssen, dass der Spaß an dem Spiel langfristig erhalten bleibt.

## Player Actions

Beide Spieler haben die Möglichkeit ihren Charakter nach rechts-links und vorne-hinten zu bewegen. Darüber hinaus haben sie die Fähigkeit zu springen. Zur Vereinfachung der Steuerung ist für einen der Spieler die Tastatur und für den anderen Spieler ein Controller vorgesehen.

Der Spieler, der seinen Charakter per Tastatur steuert, kann über W/A/S/D laufen und über Space springen. Mit dem Controller………

Den Spielern stehen zwei unterschiedliche Schläge zu Verfügung, die über die Tastatur mit Q und E ausgeführt werden. Soll der Ball leicht geschlagen werden erfolgt das über Q, während über E ein starker Schlag ausgeführt wird. Die Richtung der Flugbahn wird entweder über den Mauszeiger oder über Tasten angegeben. Mit dem Controller kann mit AÖXBU leicht und mit OIBADO stark geschlagen werden. Die Richtung wird über IBUB gesteuert.   
Um beiden Spielern die Steuerung der Flugrichtung zu erleichtern, gibt es bei beiden Charakteren ein Vektorpfeil, der diese anzeigt.

## Victory Conditions

Startet man das Spiel Volamus zum ersten Mal, kann zunächst nur der Story Modus gespielt werden. Hier beginnt ein Match immer in der ersten Welt, der Eishalle. Erreicht ein Spieler 5 Punkte, gewinnt er das Match. Werden drei Spiele in der jeweiligen Welt gespielt, ist diese Welt und der dazugehörige Charakter nun in der Collection freigeschaltet. Nach Abschluss einer Welt gelangen die Spieler in die nächste Welt. Sind alle drei Welten freigeschaltet, ist der Story Modus abgeschlossen und es erscheint ein Endscreen.

Während eines Matches müssen beide Spieler versuchen, Punkte zu erreichen. Ein Spieler erhält einen Punkt, wenn der Ball den Boden im gegnerischen Spielfeld berührt, der Gegner den Ball aus dem Feld raus schlägt oder der Gegner mehr als drei Ballkontakte direkt nacheinander hatte.

# User Interface

## Game Menus

Das Spiel ist in acht verschiedene Bereiche aufgeteilt. Zu Beginn ist ein Titelscreen zu sehen. Auf diesem wird die Geschichte eingeblendet. Über einen „Play“-Button gelangen die Spieler in das Menü. Dort haben die Spieler verschiedene Auswahlmöglichkeiten: Das Spiel wieder zu schließen, in die Settings zu gehen, ein Tutorial zu starten, die freigeschalteten Charaktere und Welten in der Collection anzusehen oder über den „Play“-Button in das Mode Select zu gelangen.  
Bevor ein Wettkampf gestartet werden kann, muss in dem Mode Select ausgewählt werden, ob der Story Modus oder ein Match gespielt werden soll. Ist ein Match gewünscht, müssen die Spieler im Character Select über den „Select“-Button ihren Charakter, beziehungsweise die Welt aussuchen. Anschließend kann über „Play“ das Match gestartet werden.  
Während eines Spieles kann jederzeit über „Escape“ das Spiel pausiert werden. Der Pause-Screen bietet die Möglichkeiten in das Menü und die Settings zu gelangen, das Match über „Retry“ zu wiederholen oder über „Return“ fortzusetzen.  
Ist ein Match beendet, erhält der Gewinner einen „Win-Screen“ und der Verlierer einen „Loose-Screen“. Jetzt können sie sich entscheiden, ob ein weiteres Match gespielt werden soll oder ob sie zur Auswahl zurück in das Menü wollen.

Eventuell unser Diagramm noch mal reinmachen?

# Graphics

## Features

Die Modelle, zu denen die Spielfiguren, der Ball und das Netz zählen, werden überwiegend mit der freien 3D Graphiksoftware „Blender“ modelliert. Die Charaktere werden entsprechend den unterschiedlichen Schlagarten und Bewegungen animiert.

Eine wichtige Eigenschaft des Balls ist seine physikalisch korrekte Flugbahn. Weiterhin soll der Schatten des Balls auf dem Boden des Spielfelds dargestellt werden. Anhand dessen soll es den Spielern erleichtert werden, die Position des Balls abzuschätzen.

## View Modes

Da es sich bei Volamus um ein Zwei-Spieler-Spiel handelt, ist es wichtig, dass beide Spieler die gleichen Spielbedingungen haben, damit ein faires Match gewährleistet werden kann. Um beiden eine gleich gute Sicht auf das Geschehen zu ermöglichen, gibt es einen Split Screen. Die jeweilige Kamera ist hinter und schräg über (ca. 55°) dem jeweiligen Spieler platziert. So haben beide eine optimale Sicht auf ihren Spieler und den heranfliegenden Ball.

## Schlageffekte

Um die verschiedenen Schlagarten optisch zu differenzieren, gibt es für diese auch unterschiedliche Effekte. Bei einem starken Schlag holt der Spieler viel weiter aus als bei einem leichten Schlag.   
Der Aufschlag wird entweder ein starker oder ein schwacher Schlag aus dem Stand sein.

## Bump Mapping

Durch die Verwendung von Bump Mapping sollen die Umgebung und der Ball realistischer wirken, ohne die Geometriekomplexität zu erhöhen.

# Audio

Für einen erhöhten Spielspaß ist es wichtig, viele verschiedene Feedbacks einzubauen. Da diese nicht nur visuell sein sollen, werden wir als Hintergrundmusik Musik aus kostenlosen Bibliotheken verwenden. Beim Erhalten von Punkten wird der jeweilige Spieler von den Zuschauern beklatscht und bejubelt. Um dieses Feedback dem Spieler anzupassen, wir es nur auf dem Lautsprecher abgespielt, auf dessen Seite der Split-Screen des Gewinners ist.

Aber auch beim Schlagen und Springen gibt es ein entsprechendes auditives Feedback.

Welche Bibliothek?

Erläutern, wozu wird sie gebraucht?

Was können Sie und was wollt ihr benutzen?

Was müsst ihr noch erweitern? – Jubelsound nur auf der seite des gewinners (wird links abgespielt wenn Spieler im linken splitscreen gewonnen hat)

# Data Management

Da das Spiel ein kleines Level System beinhaltet, ist es nötig den jeweiligen Fortschritt, der im Story Modus erreicht wird, in der Collection zu speichern.

Für den Storymodus sollen Gewinne der einzelnen Spieler gespeichert und gegebenenfalls auch auf den Zwischen-Match-Screens angezeigt werden.

Volamus wird über ein Pause System verfügen. In Anbetracht dessen, ist es auch hier wichtig die Positionen der Spieler und des Balls einzufrieren und die Zeit zu stoppen.

Gewinne der Spieler werden gespeichert, damit es am Schluss auch einen Gewinner des Story Modus gibt

# Klassen

Die Implementierung des gesamten Spiels wird in verschiedenen Klassen erfolgen. Diese Unterteilung ist in folgendem Klassendiagramm zu sehen.

Use Case diagramm

Klassendiagramm:

* Klassen:
  + Game
  + Spielfeld
    - Eis/Strand/Garten
  + Player (abstrakt)
    - Pinguin/Delfin/Hummel
  + Schiedsrichter und Regeln/Punktestand
  + Interface?
  + Ball
  + Physics
  + Gamestates?

# Milestones

Bis zum zweiten Meilenstein, der am 3.Juni sein wird, soll das Spiel über die groben Bewegungsabläufe und Handlungen verfügt. Das heißt es soll zwei Spieler geben, bei denen es sich aber noch nicht um fertige Modelle handeln muss, die sich innerhalb ihres Feldes bewegen können. Dabei steht auch noch nicht die Optik des Feldes oder gar der Umgebung im Vordergrund.   
Ziel ist es auch eine funktionierende Flugbahnberechnung des Balls zu haben und diese einzubauen. Kollisionen sollen so weit funktionieren, dass die grobe Spielhandlung zu erkennen ist.   
Weiterhin soll bereits ein kleines Menü existieren.

Für die Verwirklichung des zweiten Milestones, wurden die jeweiligen Teilziele grob auf die Teammitglieder verteilt:

Bei einem noch Kollisionen dazu? Vllt Lena, da ja Modelle erst später dazukommen? Oder bei allen ;)  
Lena Spitz: Modelle  
Lars Haider: Game States  
Mareen Allgaier: Flugbahn des Balls

Das gesamte Projekt wird bis Mitte/Ende September abgeschlossen sein. Bis dahin wird es noch zwei weitere Meilensteine geben.

Wie organisiert ihr euch?

Nutzt ihr Projektmanagementtools?

Planung von Arbeitspaketen (Was, Wer, Wann)