

# Game Design Document

## “Sock Snatcher”



Modul: Open Real Time Games Workshop  
SS 2016  
Technische Universität München

Stefan Wolf | Robert Brand | Sabine Beyer

# 1 Game Design Overview

## 1.1 Kernidee, Genre

Sock Snatcher ist ein "Hide and Seek"-Spiel für zwei Spieler, in dem es darum geht, Socken, die im Raum verteilt sind, zu bewachen bzw. diese unbemerkt zu stehlen. Dabei schlüpft ein Spieler in die Rolle eines Kindes, welches Nachts durch ein Rascheln geweckt wird und versucht, die Geräuschquelle (ein kleines Monster, welches der andere Spieler steuert) ausfindig zu machen.

Das Kind befindet sich stationär in der Mitte des Raumes und muss in der First-Person-Perspektive auf die Socken aufpassen, die um es herum verteilt sind. Das Monster muss sich im Raum bewegen und die Socken stehlen, ohne dabei vom anderen gesehen zu werden. Dazu sieht der Monster-Spieler den Raum in Vogelperspektive, in der zusätzlich visuell angezeigt wird, wo sein Gegenspieler gerade hin sieht (Field-Of-View-Kegel).

Es handelt sich somit um ein asymmetrisches Multiplayer-Spiel, bei dem die beiden Spieler verschiedene Ziele verfolgen und dazu unterschiedliche Möglichkeiten haben. Bezüglich Genre lässt sich das Spiel am ehesten als Stealth/Party-Game-Hybrid klassifizieren.

## 1.2 Key-Features

- Asymmetrischer Multiplayer mit zwei unterschiedlichen Spielerfahrungen
- Egoperspektive vs. Vogelperspektive
- Puls des Monster-Spielers beeinflusst Sichtbarkeit (nervöser Spieler = Spieler wird leichter gefunden)

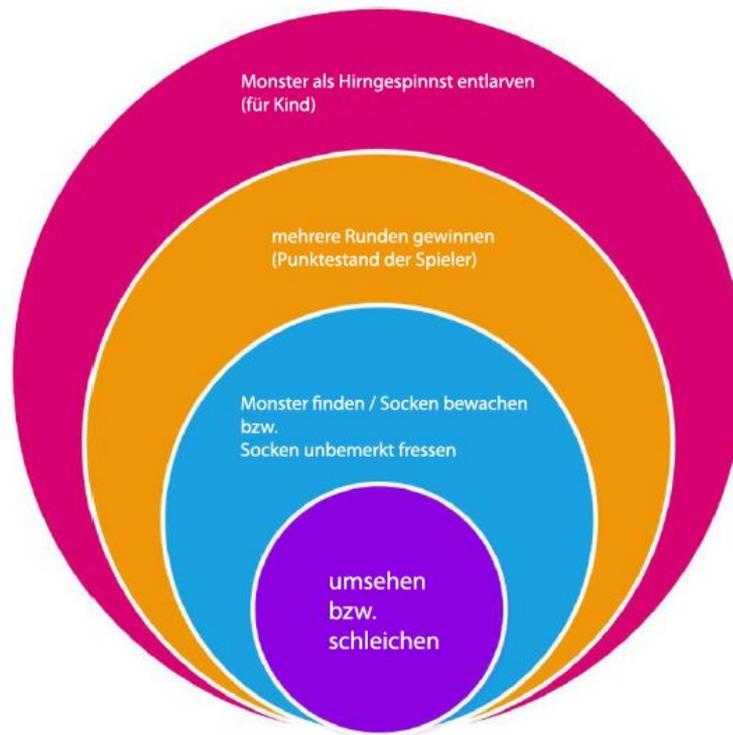
## 1.3 Zielgruppe

Zielgruppe sind Kinder ab ca. 10 Jahren, Familien, VR-Anfänger und Gelegenheitsspieler, da eine Runde "Sock Snatcher" keine übermäßig lange Spielzeit hat. Außerdem ist die Steuerung einfach gehalten (ausschließlich über Bewegung des Kopfes für stationären Spieler bzw. Control-Stick plus einen Knopf für den dynamischen Spieler), so dass der Lernaufwand gering bleibt.

Zudem können auch Kinder, die aus irgendeinem Grund bewegungseingeschränkt sind (z.B. Krankenhausaufenthalt) "Sock Snatcher" problemlos spielen.

## 2 Detailed Design

### 2.1 Game Mechanics



### 2.2 Regelwerk

- Game Space: Kinderzimmer
- Spiel beginnt, sobald "Kind" sich schlafen gelegt hat und "Monster" Raum betritt
- Gewinnbedingungen:
  - Kind: Monster sehen und mit dem Fokus-Punkt des Sichtbereiches 1 Sekunde lang ansehen (Monster "verpufft", da es nur eine Einbildung ist)
  - Monster: alle bis auf eine Socke einsammeln, ohne dabei entdeckt zu werden (optional: das Monster muss von jedem Sockenpaar eine einsammeln)
- Das Kind kann sich nicht von der Stelle bewegen, sondern nur drehen
- Das Monster kann sich nur innerhalb des Raumes bewegen
- Sobald das Monster eine Socke aufsammelt, macht es ein Geräusch welches der andere Spieler wahrnehmen kann
- Das Monster ist unsichtbar, wird aber abhängig vom Puls seines Spielers und der Anzahl der bereits eingesammelten Socken schemenhaft sichtbar für den anderen Spieler

### 3 Technical Design

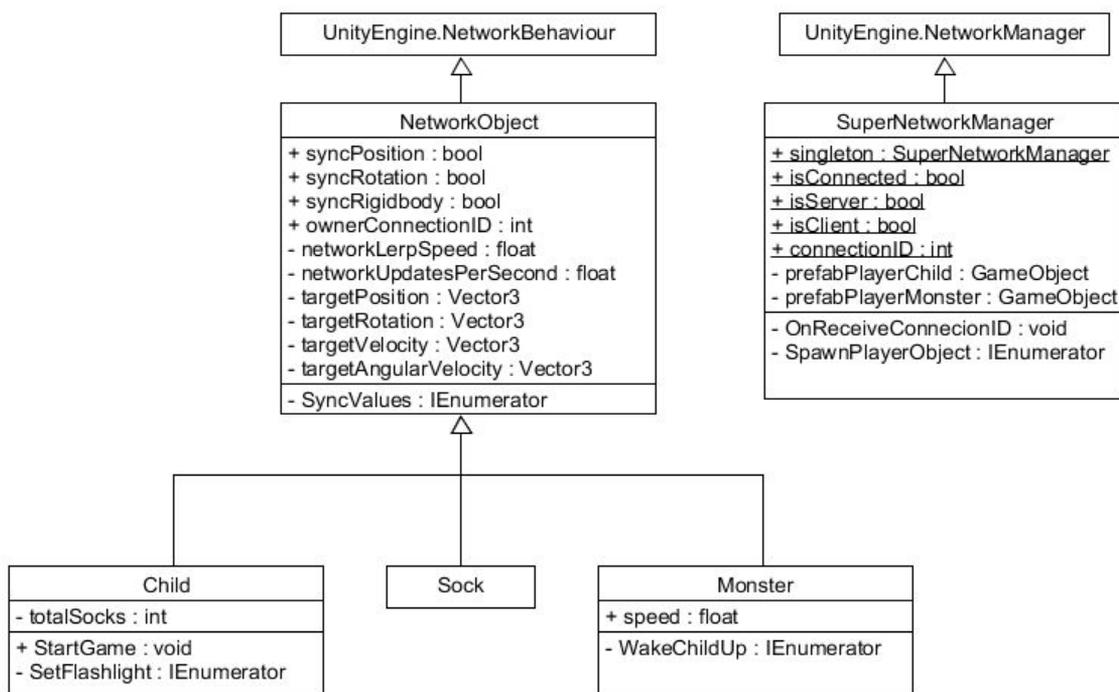
#### 3.1 Verwendete Interaktionsgeräte

Es kommen verschiedene Geräte zum Einsatz, um die beiden Spielerfahrungen zu realisieren. Der Spieler in der Rolle des Kindes verwendet eine Oculus Rift um sich im Zimmer umzusehen, dieser Spieler verwendet ansonsten keine weiteren Eingabemethoden. Der Monster-Spieler hingegen spielt an einem regulären PC-Display, verwendet einen XBOX-Controller zur Steuerung und trägt ein Zephyr Bioharness 3, welches Puls und Atemfrequenz erfasst.

#### 3.2 Networking und UML-Diagramm

Der Mehrspielermodus wird durch eine leicht abgewandelte Version des von Unity bereitgestellten NetworkManagers ermöglicht. Dieser baut eine Client-Server-Struktur auf, wobei der Spieler in der Rolle des Kindes als Host fungiert. Anders als in der von Unity bereitgestellten Standardimplementierung ermöglicht diese abgewandelte Variante das Spawnen von unterschiedlichen Spielcharakteren für einzelne Clients. Zusätzlich verfügt sie über statische Zugriffsfunktionen über den aktuellen Verbindungsstatus zum Server.

Sämtliche Objekte, die über das Netzwerk synchronisiert werden müssen, erben von einem NetworkObject - dieses bietet optionale Funktionen zur jeweiligen Synchronisierung der Position, Rotation und Physik der einzelnen Spielobjekte. Um das Netzwerk nicht zu sehr auszulasten, werden pro Sekunde nur ein paar Daten übertragen und die fehlenden Werte dazwischen durch Interpolation ermittelt.



## 4 Ästhetik

### 4.1 Moodboard



Die Spielwelt wird primär durch die Augen eines Kindes (welches nachts verschreckt aufwacht) gesehen. Daher wird kein realistisches Aussehen für das Spiel angestrebt, sondern ein Comic-artiges Visual Design mit leicht surrealen Elementen. Daher werden Objekte im Kinderzimmer nicht mit realistischen Proportionen, sondern verzerrt dargestellt. Zusätzlich könnte ein Fischaugen-Effekt eingesetzt werden. Um trotzdem einen kindlichen, Comic-artigen Stil zu erreichen werden kräftige (Primär-)farben verwendet und Objekte schwarz umrandet. Außerdem sollten Texturen stilisiert sein.

Es handelt sich dennoch um ein 3D-Spiel, da das Kind sich frei im Raum umsehen können muss.

## 4.2 Concept Art

