

Arbeitsauftrag – 2-Spieler Top-Down BulletHell

Technologie: Unity 6000.0.62f1 + FishNet

Arbeitsform: Einzelarbeit oder 2er-Team

Abgabedatum: 23.01.2026

Abgabeform: GitHub-Repository

Ziel dieses Projekts ist die Entwicklung eines **spielbaren 2-Spieler-Arcade-BulletHell-Spiels** aus der Top-Down-Perspektive.

Das Spiel muss **online im Multiplayer (Host + Client)** funktionieren und die unten aufgeführten technischen Anforderungen erfüllen.

Pflichtanforderungen (80 Punkte)

Die folgenden Punkte sind **verpflichtend** und bilden die Grundlage der Bewertung.

1. Multiplayer-Basis (FishNet) – 10 Punkte

- FishNet ist korrekt im Unity-Projekt eingerichtet
- Eine Host/Client-Verbindung ist möglich
- Zwei Instanzen können sich stabil verbinden
- Der NetworkManager ist korrekt konfiguriert

2. Spielersteuerung & Synchronisation – 15 Punkte

- Top-Down-Bewegung (z. B. WASD, Controller optional)
- Spieler sind als **NetworkObject** mit **NetworkBehaviour** umgesetzt
- Ownership ist korrekt gesetzt (nur der eigene Spieler ist steuerbar)
- Spielerbewegung wird **server-authoritativ** synchronisiert
- Mindestens **eine SyncVar** (z. B. Name, HP, Farbe)

3. Schießen & Bullet-Hell-Mechaniken – 20 Punkte

- Netzwerkfähiges Projektilsystem
- Mindestens **zwei unterschiedliche Bullet-Patterns**(z. B. gerader Schuss, Spread-Shot)
- Projektile werden korrekt synchronisiert
- Feuerrate / Cooldown-System vorhanden
- Treffererkennung und Schaden funktionieren auf beiden Clients korrekt

4. Gegner-, Wave- oder Boss-System – 15 Punkte

- Mindestens **zwei unterschiedliche Gegnertypen**
- Gegner werden **serverseitig gespawnt**
- Entweder:
 - ein Boss **oder**
 - eine klar erkennbare Wave-Struktur
- Gegner- und Boss-Bullets sind auf beiden Clients konsistent sichtbar

5. Leben, Treffer & Spielablauf – 10 Punkte

- Spieler verfügen über ein HP- oder Lebenssystem
- Schaden und Treffer werden korrekt synchronisiert
- Klarer Gameflow:
Start → Spiel → Game Over / Victory → Endscren oder Restart

6. HUD & Punkte – 10 Punkte

- HUD mit mindestens:
 - HP / Leben
 - Score
- Punktevergabe für Kills und/oder Überlebenszeit
- Score wird synchronisiert und korrekt angezeigt
- **Highscore-Liste über PHP & SQL (serverseitig)**

Frei wählbare Punkte (max. 20 Punkte)

Zusätzlich zu den Pflichtanforderungen können **eigene Erweiterungen** umgesetzt werden.

Alle Bonusfeatures müssen in der `README.md` dokumentiert sein.

Maximal 20 Bonuspunkte werden angerechnet – unabhängig von der Anzahl der umgesetzten Features.

Beispiele für Bonusfeatures (nicht verpflichtend):

A) Lobby- & Matchmaking-System (5–10 Punkte)

- Lobby vor Spielstart
- Anzeige der verbundenen Spieler
- Ready-System
- Gemeinsamer Szenenwechsel
 - z. B. auf Basis von:
<https://github.com/Patrick-van-Halm/FishNetworking-LobbySystem>

B) Power-Ups & Spielmechaniken (5–10 Punkte)

- Shields, Laser, Bomben, Slow-Motion
- Unterschiedliche Spielerklassen (Tank / Schnell / Schütze)

C) Komplexere Bullet-Patterns (5–10 Punkte)

- Spiralen, Ringe, Rotations- oder Orbit-Bullets

D) Erweiterte Persistenz (5–10 Punkte)

- Highscore-Listen
- JSON-basierte Speicherung
- Mehrere gespeicherte Statistiken

E) Koop- / Versus-Elemente (5–10 Punkte)

- Gemeinsamer Spezialangriff
- Combo-System
- Freund-/Feind-Schüsse

F) Visuelle & technische Verbesserungen (5–10 Punkte)

- Prediction / Interpolation
- Synchronisierte VFX & SFX
- Kameraeffekte (Zoom, Screenshake, Post-Processing)

README.md (Pflichtbestandteil im GitHub-Repository)

Die `README.md` im Root des GitHub-Repositories **ist verpflichtend** und muss folgende Punkte enthalten:

- **Kurzbeschreibung des Spiels**
- **Anleitung zum Starten von Host und Client**
- **Technischer Überblick**
 - verwendete RPCs
 - verwendete SyncVars
 - Bullet-Logik
 - Gegner-Logik
- **Beschreibung der Persistenz**
z. B. PHP/SQL, JSON, PlayerPrefs
- **Übersicht der umgesetzten Bonusfeatures**
- **Bekannte Bugs oder Einschränkungen**

GitHub-Abgabe

- Ein **öffentliches GitHub-Repository**
- Repository-Name: frei wählbar, aber eindeutig
- **Link zum Repository** wird zur Abgabe eingereicht
- Das Repository muss enthalten:
 - Unity-Projekt (ohne unnötige Build-Artefakte)
 - Eine vollständige `README.md` im Root

Hinweis:

Eine saubere Ordnerstruktur, verständliche Commit-Nachrichten und regelmäßige Commits fließen positiv in die Bewertung ein.

Bewertung

Kriterium	Beschreibung	Punkte
Multiplayer-Basis (FishNet Setup)	Funktionierende Host/Client-Verbindung, korrekt konfigurierter NetworkManager	10
Player-Steuerung & Synchronisation	Top-Down-Movement, Ownership korrekt, SyncVars, saubere Synchronisation	15
Bullet-Hell Shooting	Mind. 2 Schussmuster, Projektilnetzwerk, Trefferlogik, Cooldowns	20
Gegner, Waves & Boss	Mind. 2 Gegnertypen, Server-Spawn, Boss oder Waves	15
Leben & Gameflow	HP-System, Schaden, Game Over / Victory-Flow	10
HUD & Score-Anzeige	HP/Score sichtbar, synchronisiert, nachvollziehbar	10
Frei wählbare Erweiterungen	<u>Dokumentierte</u> Zusatzfeatures	20
Gesamtpunktzahl		100