



Berner Fachhochschule  
Haute école spécialisée bernoise  
Bern University of Applied Sciences



# RocketShooter

Bti7527a – Game Developer Challenge 2015

Marco Fülleemann, Julien Villiger, Daniel Inversini

# Erweiterungen

- ▶ Multiplayer
- ▶ Rockets
- ▶ Models
- ▶ GUI

# Multiplayer

- ▶ Connect onLevelLoad / Disconnect onQuit
- ▶ Spawnpoints

```
[RPC] void RequestSpawnPos(NetworkPlayer player)
{
    if (Network.isServer) {
        NetworkPlayer p;
        for (int i = 0; i < Network.connections.Length; i++) {
            p = Network.connections[i];
            if (p == player) {
                _networkView.RPC("RespondSpawnPos", RPCMode.AllBuffered, (i+1) % 3, p);
                break;
            }
        }
    }
}

[RPC] void RespondSpawnPos(int spawnIndex, NetworkPlayer player)
{
    if (Network.player == player) {
        SpawnPlayer(spawnIndex);
    }
}
```

# Multiplayer

## ► Synchronisation

```
public class PlayerNetworkBehaviour : BasicNetworkBehaviour {
```

```
    private WheelSynchronizer wfl;  
    private WheelSynchronizer wfr;  
    private WheelSynchronizer wrl;  
    private WheelSynchronizer wrr;  
    private TurretSynchronizer turretSynchronizer;
```

Erstellung Synchronizer

```
    protected override void initialize ()  
    {...}
```

Scripts deaktivieren

```
    protected override void initializeOpponent ()  
    {...}
```

Positionen & Rotationen  
interpolieren

```
    protected override void SyncedMovement(float duration)  
    {...}
```

Daten senden

```
    protected override void OnOutgoingSync(BitStream stream, NetworkMessageInfo info) {...}
```

Daten empfangen

```
    protected override void OnIncomingSync(BitStream stream, NetworkMessageInfo info) {...}
```

```
}
```

# Rockets

- ▶ Prefab (Rigidbody) instanziiieren
- ▶ Position von Turret übernehmen
- ▶ Force hinzufügen
- ▶ Sound abspielen
- ▶ Nach Timeout GameObject zerstören



# Rockets

- ▶ Gleichzeitig Explosion (Partikelsystem) instanziiieren
- ▶ Zufälliger Sound abspielen
- ▶ ExplosionForce für nahe Rigidbodies

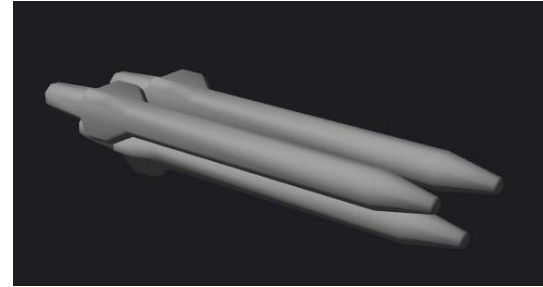
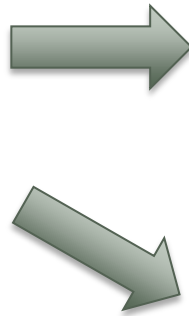
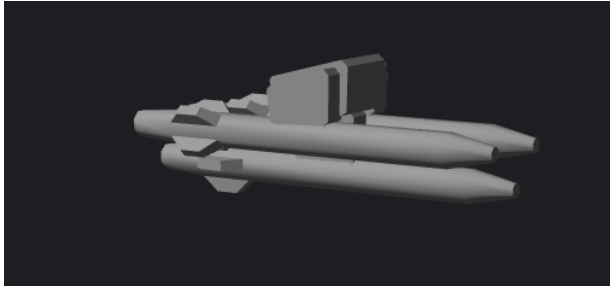


# Rockets

- ▶ Punkteberechnung nach Treffer
  1. Dictionary in Spiel-Hierarchie mit Punktestand von jedem Spieler
  2. Lineare Funktion für Punkteberechnung je nach Distanz zu Auto
  3. Erspielte Punkte per RPC übermitteln
  4. Dictionary Updaten und anzeigen
- ▶ Bei Spielende Gewinner ermitteln und Rang anzeigen
  - ▶ Zentral geführter Counter

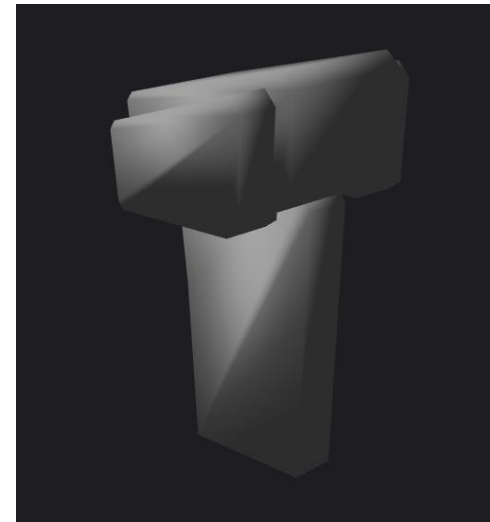
# Models

- ▶ Bearbeitung im VS



- ▶ Rotation

- ▶ Winkel





# GUI

- ▶ Punkte, Zeit
  - ▶ Crosshair
    - ▶ Scrollbar
    - ▶ Sprites
- «onGUI» Funktion mit DrawTexture