Melee Combat: Design and Tech
Val Gorbunov
Who I Am

Val Gorbunov Khanovich
Table of Contents

● Industry vs Roblox:
  • Shooter vs Melee Tech

● Melee Combat:
  • What makes a good system

● Design a Game:
  • Pick a set of features we want
  • Design tech and implementation
  • Test demo and feedback

● Alternative designs, tech considerations
Industry vs Roblox
Shooters in Industry

- Arcade, Milsim, Arena
- Standards:
  - Favor Shooter
  - Ballistic vs Hitscan
- Multiplayer Tech:
  - One Directional
Shooters in Roblox

- Standards match industry
- Ballistics in some games
Melee in Industry

- More varied than shooters
- Unique Tech
- Bi-directional interaction
Melee in Roblox

- Most Cases
  - Designed much like shooters
  - No sense of "back and forth"
  - No real feedback
- Exception
  - Legendary
Shooter vs Melee Design (Industry)
Making a Good Melee Combat system
Melee Ideal – The Perfect Duel

- Think samurai movies/anime

- Defending + Looking for opening to Strike

- Tricking an enemy into creating an opening

(Masaki Kobayashi’s *Samurai Rebellion* (1967))
Attacks – Universal Concepts

- Windup Time
- Interruptible
- Stagger
- Damage
- Chain/combos
Defense – Universal Concepts

- **Dodge**
  - Positioning Based

- **Block**
  - *Safe* defense without extra benefit

- **Parry**
  - *Risky* counter that rewards precision
Defense vs Offense

- Force action, provide FEEDBACK
- Main takeaway: **Stagger + Parry**
- Punish single-minded focus
- Blocking should not be sustainable
- Punish offensive spam
- Parry should be risky but rewarding
Block + Stagger
Why parries are awesome
Design a Game
Example in Roblox
Let’s Design a Game

- Stamina System
- Directional Attack
- Block, Parry, Dodge
- Multiplayer/PVP – 1 vs 1
Anatomy of an *Attack*

- **Attacker**
  - Does Damage
  - Staggers

- **Defender**
  - Can Block, Parry, Dodge
  - Can take Damage
  - Can also Attack

- **Multiplayer:**
  - Latency vs Attack Speed
New Attack == Data, not Code

```lua
-- RIGHT ATTACKS --
-- EFFECTDATA
createEffect( offsetTime, effectDuration, effectType, ... )
-- MOVEMENTDATA
createMovement(offsetTime, moveDuration, moveDirection, moveDis)
-- ACTIONDATA
createAction(length, priority, animation, reach, damage,
local r1Movement = ActionManager.createMovement( 0.2, 0.4, Vector3.new(0, 0, 1), 5)
local rightAction_1 = ActionManager.createAction( 1.0, 1, Animations.RightSwing1, 4, 20,
table.insert(rightActions, rightAction_1)

local r2Movement = ActionManager.createMovement( 0.2, 0.2, Vector3.new(1, 0, 1), 2)
local rightAction_2 = ActionManager.createAction( 1.2, 1, Animations.RightSwing2, 4, 30,
table.insert(rightActions, rightAction_2)

-- LEFT ATTACKS --
local l1Movement = ActionManager.createMovement( 0.2, 0.4, Vector3.new(0, 0, 1), 5)
local leftAction_1 = ActionManager.createAction( 1.0, 1, Animations.LeftSwing1, 4, 20,
table.insert(leftActions, leftAction_1)

local l2Movement = ActionManager.createMovement( 0.2, 0.2, Vector3.new(-1, 0, 1), 2)
local leftAction_2 = ActionManager.createAction( 1.2, 1, Animations.LeftSwing2, 4, 30,
table.insert(leftActions, leftAction_2)
```
Anatomy of an Attack

Player 1
- Attack Start
- Attack Hit
- Get Parried

Server
- Forward Event
- Forward Event

Player 2
- Display Warning
- Check Dodge
- Check Block
- Check Parry
- HP Damage
- Stam Damage

Events:
- Parry Success
- Parry Fail
- Block Success
Anatomy of an *Attack* – Parry Timing
Roblox - Stagger
Designing for Multiplayer

- **Latency**
  - Slower attacks easier to deal with latency

- **Hit Detection**
  - Mix of Collision (Unlocked) + Guaranteed when locked

- **Client - Driven**
  - Favors Active (attack or defense)
  - Out of sync
Roblox – Raw Gameplay
Postmortem
Design Feedback Loop – Tune for Fun

- Important to Tune animations
  - Damage on visual hit
  - When does sword move fastest?
  - Tweak details

- Movement complements animations
  - Attacks should push the enemy back on successful hit
Feedback Follow-up – Attack and Stagger
Alternative Designs

- **Server Authoritative System**
  - Security vs Responsiveness
  - Predictive logic on Client

- "Paired" animation system
  - More realistic sword deflections
  - Requires even more Engine-level code
Conclusion

- Roblox market niche with room for expansion
- Requires custom "Lua Melee Engine" writing
- Art intensive
- Just like in industry - there are less "rules" to designing Melee Combat Systems vs Shooter Combat Systems
- Good feedback for Roblox Engineers
External Material

- Game Maker's Toolkit - What Makes a Good Combat System
  - [https://www.youtube.com/watch?v=8X4fx-YncqA](https://www.youtube.com/watch?v=8X4fx-YncqA)
External Gameplay Sources

- Destiny 2 - Bungie
- Battlefield 4 - DICE
- Player Unknown's Battlegrounds - PUBG Corp
- Dark Souls III - Fromsoftware
- Final Fantasy XV - Square Enix
- For Honor - Ubisoft
- Guild Wars 2 – Arenanet
Roblox Gameplay Sources

- Polyguns – Mailbox Games
- Phantom Forces – StyLiS Studios
- Counter Blox: Roblox Offensive – ROLVe Community
- Beyond – B-b Studio
- Dragon Ball Z Final Stand - SnakeWorl
- Swordburst 2 – Swordburst 2 (team)
- Legendary – pa00