

[Combat Module] [Tank War] Shooting Logic

THIẾT KẾ LOGIC BẮN (2.5D-aware, Unity Kinematic)

Version: 1.1

Người viết:  Kent (QuocTA)

Ngày cập nhật: 17 - 09 - 2025

Phiên bản	Ngày	Mô tả	Người viết	Người review	Duyệt?
v1.0	08 - 09 - 2025	Xong nội dung của file	 Kent		<input type="checkbox"/>
v1.1	17 - 09 - 2025	Format lại file	 phucyth12		<input type="checkbox"/>

1. Mục lục

2. Mục tiêu & Phạm vi

1.1 Mục tiêu UX

1.2 Phạm vi

1.3 Vấn đề hiện tại

3. Nguyên tắc thiết kế

4. Thuật ngữ & Định nghĩa

5. Tổng quan kiến trúc hệ thống

5.1 Lớp A — 2.5D Collision Band (MẶC ĐỊNH ON)

5.2 Lớp B — Multi-Height Sweep (OPTIONAL)

5.3 Lớp C — Smart Pitch Assist (OPTIONAL)

6. Yêu cầu (Request)

6.1 Yêu cầu chức năng (FR)

6.2 Yêu cầu phi chức năng (NFR)

7. Cấu hình dữ liệu (không code)

7.1 BulletProfile (đề xuất trường)

7.2 ClassProfile (band preset theo class)

7.3 Blocker Category (cho Level/Art/Dev cấu hình)

8. Quy trình bản (Flow — không code)

2. Mục tiêu & Phạm vi

1.1 Mục tiêu UX

- Đạn **không** “lướt qua” vật cản thấp khi người chơi kỳ vọng bị chặn.
- Giữ **đạn kinematic** (không dùng Rigidbody/forces) để dễ kiểm soát & nhẹ CPU.
- Tăng **tính “công bằng trực quan”** trong bắn ngang (mặt phẳng XZ) nhưng vẫn tôn trọng chênh lệch cao độ (Y).
- Cấu hình linh hoạt theo **tank class, bản đồ, mode**.

1.2 Phạm vi

- Áp dụng cho **đạn thẳng (hitscan-like kinematic projectile)** của mọi class.
- Không bao gồm tên lửa đạn đạo/đạn cong/đạn bám mục tiêu (sẽ có GDD riêng).

1.3 Vấn đề hiện tại

- Vì muzzle (nòng súng) cao, đạn bay thẳng ở cao độ Y cố định → **vật cản thấp** (thùng, tường lưng, lan can) bị **lướt qua đỉnh** → UX cảm giác “**đạn xuyên vật cản**” .

3. Nguyên tắc thiết kế

- **2.5D trước, 3D sau**: Quyết định va chạm ưu tiên theo **mặt phẳng XZ** (nhận thức của người chơi), sau đó kiểm tra **chồng lấn theo Y** để xác thực.
- **Tha lỗi có kiểm soát**: Cho phép **yForgiveness** nhỏ để giảm các trường hợp “suýt trúng” .
- **Data-driven**: Toàn bộ tham số có thể cấu hình per-class, per-mode.
- **Hiệu năng**: Tối đa 1 sphere/capsule cast mỗi frame/đạn (mặc định); các sweep bổ sung có thể bật/tắt theo map.
- **Khả dụng đa bộ phận**: Dev đọc → implement; Art đọc → sản xuất VFX/SFX/UI; Level Design đọc → quy ước chiều cao; QA đọc → test.

4. Thuật ngữ & Định nghĩa

- **Muzzle**: điểm phát bắn trên nòng súng.
- **Attack Band (Y-band của đạn)**: khoảng cao độ mà đạn được phép coi là chạm va chạm. Xác định bởi `[y - bandDown, y + bandUp]` .
- **Hit Band (Y-band của vật cản)**: khoảng cao độ “thân” của vật cản/đối tượng được tính va chạm, lấy từ `collider.bounds` và **HitBandPad**.
- **yForgiveness**: biên độ “tha lỗi” khi hai band gần chạm nhau.
- **Multi-Height Sweep**: bó ray/cast song song tại nhiều cao độ mẫu từ mặt đất đến gần muzzle để phát hiện vật cản thấp ở gần.
- **Smart Pitch Assist**: dịch nhẹ **góc pitch** của hướng bắn ($\pm 1^\circ \sim 3^\circ$) trước khi spawn đạn để giảm các tình huống hụt vô lý.

- **Blocker Category:** phân loại vật cản (Wall_Low/Wall_High/Fence_Thin/Vehicle/Shield/Vegetation...).
-

5. Tổng quan kiến trúc hệ thống

Hệ thống gồm **3 lớp**; có thể bật/tắt độc lập theo mode/map:

5.1 Lớp A — 2.5D Collision Band (MẶC ĐỊNH ON)

- Đạn di chuyển kinematic; mỗi frame thực hiện **Sphere/Capsule Cast** theo hướng **bắn** → nếu trúng collider, kiểm tra **overlap Y-band** giữa **Attack Band** (đạn) và **Hit Band** (đối tượng).
- Nếu **overlap** (kể cả trong ngưỡng **yForgiveness**) → tính là **HIT**.
- Nếu **không overlap** → bỏ qua va chạm (tiếp tục bay).

5.2 Lớp B — Multi-Height Sweep (OPTIONAL)

- Khi bóp cò, thực hiện thêm 2–3 **sweep** tại các cao độ mẫu (`groundY+0.3`, `+0.6`, `+0.9` m).
- Nếu bất kỳ sweep nào báo va chạm **sớm hơn** đường thẳng từ muzzle → ưu tiên vị trí đó để **chặn đạn**.
- Dừng khi map có **nhiều vật cản lùn/khe hở**.

5.3 Lớp C — Smart Pitch Assist (OPTIONAL)

- Trước khi spawn đạn, điều chỉnh **nhẹ** pitch trong phạm vi $\pm 3^\circ$ để “ghé xuống” nếu trước mặt có blocker thấp.
- Không phải auto-aim; chỉ giảm case “bay sượt đỉnh” khó chịu.

[Inference] Multiplayer có thể cần xử lý hit **server-authoritative**. Khi đó, logic 3 lớp cần **phản chiếu** tương đương ở server hoặc dùng seed/lockstep để tránh lệch.

6. Yêu cầu (Request)

6.1 Yêu cầu chức năng (FR)

- **FR-1:** Đạn bị **block** khi đường bay giao XZ với collider thuộc **Blocker Category** và **Attack Band** overlap **Hit Band** của đối tượng.

- **FR-2: yForgiveness** áp dụng cả hai phía (đạn & vật cản).
- **FR-3:** Cho phép **bật/tắt** Multi-Height Sweep và Pitch Assist theo **map/mode**.
- **FR-4:** Cho phép **preset per-class** (Scout/Assault/Heavy) cho bandUp/bandDown.
- **FR-5:** Tôn trọng **team filtering** (không hit đồng đội nếu thiết kế yêu cầu).
- **FR-6: End-of-Range:** đạn nổ/biến mất khi đạt range tối đa, có VFX khác biệt so với hit.
- **FR-7:** Bề mặt khác nhau tạo **VFX vật liệu** khác nhau (kim loại/đất/đá/wood/shield).

6.2 Yêu cầu phi chức năng (NFR)

- **NFR-Perf:** Mặc định ≤ 1 **shape cast/frame/bullet** (Lớp A). Nếu bật Lớp B, tổng ≤ 3 **cast thêm ở khởi phát**.
- **NFR-Data:** Mọi tham số có thể chỉnh via **ScriptableObject** (BulletProfile, ClassProfile, MapProfile).
- **NFR-UX:** Tỷ lệ “**khi người chơi nghĩ trúng mà game báo hụt**” phải $< 5\%$ trong playtest.
- **NFR-QA:** Có **ma trận test** kèm tiêu chí pass/fail.

7. Cấu hình dữ liệu (không code)

7.1 BulletProfile (đề xuất trường)

Trường	Kiểu	Mô tả	Default [Baseline]
Speed	float	Tốc độ đạn (m/s)	40
Range	float	Tầm bắn tối đa (m)	Theo vũ khí
Radius	float	Bán kính cast (m)	0.1
BandUp	float	Độ dày band phía trên (m)	0.5
BandDown	float	Độ dày band phía dưới (m)	0.8
YForgiveness	float	Tha lỗi chênh Y (m)	0.15
UseMultiHeight Sweep	bool	Bật lớp B	Map-dependent
SweepHeights	float[]	Các cao độ mẫu (m)	[0.3, 0.6, 0.9]
UsePitchAssist	bool	Bật lớp C	Off (mặc định)
PitchAssistMax Deg	float	Biên pitch tối đa (°)	3
HitMask	mask	Layer blocker/target	Theo dự án

7.2 ClassProfile (band preset theo class)

Class	BandDown	BandUp	Ghi chú
Scout	0.9	0.4	“áp sần” hơn → dễ bị blocker lùn chặn (fair)
Assault	0.8	0.5	Cân bằng
Heavy	0.7	0.6	Muzzle cao, band trên rộng hơn chút

| Toàn bộ con số là **[Baseline]** để playtest/tuning.

7.3 Blocker Category (cho Level/Art/Dev cấu hình)

Category	Mô tả	Height gợi ý (m)	HitBandPad (m)	VFX vật liệu
Wall_Low	Tường lửng, thùng cao	0.7–1.1	0.05	đá/xi măng/kim loại
Wall_High	Tường cao, cột lớn	>1.2	0	đá/xi măng/kim loại
Fence_Thin	Hàng rào mỏng	0.9–1.2	0.08	tia lửa kim loại nhỏ
Vehicle	Xe/chướng ngại lớn	>1.0	0.05	kim loại nặng
Shield	Khiên năng lượng	0.5–∞	0	năng lượng
Vegetation	Cây bụi	0.3–1.2	−∞ (không block)	lá/bụi (chỉ FX xuyên)

Vegetation mặc định **không block** (trừ khi có thiết kế đặc biệt).

8. Quy trình bắn (Flow — không code)

Pre-Fire

- Lấy `ClassProfile` → set bandUp/bandDown.
- (Nếu bật) `Smart Pitch Assist`: đánh giá blocker thấp trong 2–6m trước mặt → áp $\text{pitch offset} \leq \text{PitchAssistMaxDeg}$.
- (Nếu bật) `Multi-Height Sweep`: cast tại `SweepHeights` (theo groundY). Nếu phát hiện hit **sớm hơn** đường thẳng từ muzzle → lưu **HitCandidate**.

Flight

- Mỗi frame: **Sphere/Capsule Cast** theo hướng bay trong khoảng `Speed * Δt`.
- Nếu trúng collider thuộc `HitMask` → tính **overlap Y-band** (Attack vs Hit).
- Nếu overlap (kể cả trong `YForgiveness`) → **HIT** (ưu tiên **HitCandidate** nếu có).
- Nếu không có hit & đã bay hết `Range` → **End-of-Range**.

Impact

- Tạo **VFX vật liệu**, **SFX**, **decal** (nếu bề mặt cho phép); gửi sự kiện damage; sinh camera shake nhỏ (xem [mục 9](#)).
- Ghi Telemetry (xem [mục 9](#)).

9. Tương tác & Quy tắc đặc biệt

- **Đồng đội:** mặc định **không block** (friendly-fire off). Nếu bật friendly-block, thêm **Outline/UI** cảnh báo.
 - **Shield:** luôn block, hiển thị **impact FX energy** đặc trưng; xuyên shield cần đạn/skill đặc biệt.
 - **Vật thể rất mỏng** (cọc/thanh): ưu tiên **CapsuleCast** hoặc tăng **Radius** ở bullet nhanh.
 - **Mặt dốc/ramps:** Multi-Height Sweep tính **groundY** tại XZ để không sai lệch.
 - **Tốc độ đạn rất cao:** giữ **shape cast** (không dùng ray đơn) để tránh tunneling.
-

10. Art / VFX / SFX / UI (Asset Bible rút gọn)

10.1 VFX

- **Hit Blocker (vật liệu):** 3–5 preset theo vật liệu (đá/kim loại/gỗ/đất/năng lượng).
- **Hit Shield:** flare năng lượng + ripple.
- **End-of-Range:** pop nhẹ, khác biệt với Hit Blocker.
- **Tracer:** chiều cao bám đường đạn; có thể **khóa Y** theo bullet để nhất quán.
- **Debris:** bụi/đá vụn/tia lửa nhỏ; scale theo tốc độ va chạm.

10.2 SFX

- **Impact theo vật liệu; Shield** riêng.
- **End-of-Range:** âm “tắt” nhẹ, không nhầm với trúng mục tiêu.
- **Ducking** ngắn khi hit gần camera.

10.3 UI/Camera Feedback

- **Crosshair State:**
 - Idle, Firing, **Blocked-Soon** (nếu Sweep báo blocker ở 2–3m trước mặt → viền crosshair mờ cảnh báo 150ms).
- **Camera Shake:**
 - Hit Blocker: 0.15–0.2 amplitude, 0.1s.

- Hit Shield: 0.1 amplitude, 0.12s.
- End-of-Range: không rung.
- **Aim Assist Indicator** (nếu Pitch Assist on): icon mờ 100ms để truyền thông minh bạch.

10.4 Yêu cầu Naming & Foldering

- VFX/Bullet/Hit_Metal, VFX/Bullet/Hit_Stone, VFX/Bullet/Hit_Shield ...
- SFX/Bullet/Impact_Metal, ...
- UI/Crosshair/*

11. Level Design Guideline (Chiều cao & Tags)

- Quy ước đơn vị: 1 Unity unit = 1 m.
- Dải chiều cao khuyến nghị:
 - **Blocker thấp:** 0.7–1.1 m → tag `Wall_Low`.
 - **Blocker cao:** >1.2 m → tag `Wall_High`.
 - **Lan can/hàng rào mỏng:** 0.9–1.2 m → tag `Fence_Thin`.
- **Collider:** ưu tiên **Box/Convex Mesh**, tránh concave phức tạp.
- **HitBandPad:** theo bảng 7.3; có thể override ở prefab.
- **Vegetation:** mặc định **không block** (trừ prefab đặc biệt có tag `Blocker`).
- **Scene Test:** mỗi map cần **khu test** gồm: tường 0.8/1.0/1.2m, lan can mỏng, dốc 15°, bồn cây.

12. QA — Ma trận Test & Tiêu chí Pass

Case	Thiết lập	Kỳ vọng	Pass nếu
T1	Bắn vào tường 0.8 m ở cự ly 5 m	Đạn bị chặn	Tỷ lệ chặn \geq 99%/100 phát
T2	Tường 1.2 m, mọi class	Chặn với mọi class	100%
T3	Lan can mỏng 1.0 m	Chặn ; VFX kim loại mỏng	99% + đúng VFX
T4	Bụi cây 1.0 m	Không chặn (FX lá mờ nếu có)	100%
T5	Blocker 0.9 m rất gần (2 m)	Chặn (Sweep/Pitch Assist xử lý)	98%
T6	Dốc 15°	Kết quả nhất quán	\geq 98%
T7	Đạn tốc độ cao	Không tunneling	100%
T8	Friendly trước mặt	Không block (ff off) hoặc block theo rule	Đúng rule

Telemetry đối chiếu: ghi `PerceivedHitMismatch` (người chơi ngấm blocker nhưng game cho xuyên) mục tiêu < 5%.

13. Telemetry & Tuning

Sự kiện

- `Bullet.Fire` {class, map, pitchAssistUsed}
- `Bullet.HitBlocker` {blockerCategory, distance, normalY, bandOverlapDelta}
- `Bullet.EndOfRange` {distance}
- `UX.PerceivedHitMismatch` {context} (đánh dấu khi người chơi ngấm XZ trúng blocker nhưng không hit; thu thập qua debug tool/QA)

Dashboard đề xuất

- Tỉ lệ HitBlocker theo **blockerCategory** & **class**.
- Phân bố `bandOverlapDelta` để điều chỉnh `YForgiveness`.
- Tỉ lệ `pitchAssistUsed` theo map → cân nhắc bật/tắt mặc định.

14. Rủi ro & Kế hoạch triển khai

- **Rủi ro:** Over-assist → cảm giác “dính ảo” .
Mitigation: Giới hạn `PitchAssistMaxDeg ≤ 3°` ; review clip QA.
- **Rủi ro:** Hiệu năng giảm khi bật Sweep toàn map.
Mitigation: Chỉ bật ở map nhiều blocker thấp; cache groundY.
- **Triển khai từng bước:**
 - Lớp A (mặc định) → 2) Lớp B (map-specific) → 3) Lớp C (A/B test).
- **Fallback:** Cho phép tắt Lớp B/C bằng config runtime.

15. Tích hợp hệ thống liên quan

- **Damage & Combat Log:** sự kiện `OnBulletHit` gọi damage, ghi Combat Score.
- **VFX Tiers (3/6/9/10):** kế thừa hiệu ứng theo **ATK/FireRate**; Hit FX material tách riêng.
- **Audio Mix:** route Impact → bus SFX_Impact; Shield → SFX_Energy.
- **UI:** Crosshair state + icon PitchAssist (nếu bật) 100–150 ms.

16. Phụ lục

16.1 Layer & Tag (đề xuất)

- **Layers:** `Obstacle_Blocker` , `Unit_Tank` , `Shield` , `Vegetation` , `Projectile` .
- **Tags:** `Wall_Low` , `Wall_High` , `Fence_Thin` , `Vehicle` , `Shield` , `NonBlock_Vegetation` .

16.2 Tham số gợi ý theo mode

Mode	UseMultiHeightSweep	UsePitchAssist	Ghi chú
Death Match	ON (map có nhiều thùng)	OFF	Ưu tiên cảm giác “đã” nhưng minh bạch
Base Capture	ON selective	ON ($\leq 2^\circ$)	Tránh spam “sượt đỉnh” khi thủ base

16.3 Glossary

- **2.5D**: logic ưu tiên XZ, kiểm chứng Y.
- **Band**: khoảng chiều cao chấp nhận và chạm.
- **Sweep**: loạt kiểm tra song song ở nhiều cao độ.