FPS – Things to work on

Enemies must wait until being within a minimum and maximum range before attacking with a particular weapon

Range stats should obviously be stored in the attack itself, because an enemy might want to stay in a particular positional range, but choose to attack/not attack with their weapon based on another range

Variables

Aiming stats while aiming

Aiming stats while telegraphing

Aiming stats while firing

FIGURE OUT: How do I emulate the inaccuracy and difficulty aiming that people can have when trying to aim at very short range? Using RotateTowards has no overshoot.

* Maybe I can have a ‘rotation speed and velocity’ quaternion that accelerates towards ‘the rotation required to rotate towards the target’ but somehow normalised and multiplied by the aim speed?

I have an AIAttack class that stores data on a particular enemy attack, and an AimAtTarget state machine behaviour that the enemy uses for actually aiming and performing the attack. I’m beginning to find these classes are too closely intertwined, and it’s a huge pain figuring out which variables and functions should be in one class versus the other. I also can’t think of any scenarios where I would want to use one of those classes without the other, plus the fact that the specific behaviour for aiming and timing attacks might potentially change depending on the nature of the attack. Do you guys think I should just combine them together into a single class?

Finish implementing AIAttackPhase as an action with conditions, abort state and completed state

Change AimAtTarget to just continuously track gaze towards target

Bugs to fix

|  |  |  |
| --- | --- | --- |
|  | Easy | Difficult |
| High priority | * Weapon selector will not update when the player first opens it, only once they switch to something new. * Fix melee attack and game menu buttons appearing despite not doing anything * Remove redundant UI bindings in input action asset | * If a reload is cancelled, the weapon needs to return to its ready state before being fired * Fix dodging code to prevent the player from accelerating ridiculously fast and slamming into a wall |
| Low priority |  | * Enemy health meter doesn’t move correctly when aiming through the rifle scope * Figure out math to show correct aspect ratio in options menu |

Features to implement

|  |  |  |
| --- | --- | --- |
|  | Easy | Difficult |
| High priority | * Separate hit marker for critical damage * Separate hit marker for kills * Looping animation that plays when health is low * Serves the same function as the ‘strawberry jam’ effect but without obscuring vision, and animated so the player notices it more easily | * Make reload animations more intuitive and polished so player can properly cancel out of the shotgun loading animation after only loading a small number of shells * Melee attack * Tweak combat feel to be slower paced and allow for more defensive play and tactical thinking * Make enemies more cautious to advance, and reduce player movement speed * Allow player to stagger enemies * Have enemies attempt to avoid attacks * Have enemies attempt to stay close to cover * Make more complex environments with more options to conceal yourself from enemies * Try having wait periods before enemies try different strategies and behaviours * Stunning enemies * Enemy attack avoidance * Explosions and explosive weapons * Button remapping menu * More weapons * Grenade launcher * Hand grenade/trigger bomb * Taser slugs * Enemies * Objective system |
| Low priority |  | * Experiment with having player jump forward in the direction they’re aiming, for improved precision platforming. * Tweak interact function so it auto-targets the interactable closest to the reticle * Experiment with button sequence-based reloading, for more gameplay ‘meat’ compared to pressing one button and watching an animation play out |

Resolution and refresh rate settings

Deathloop – two separate options for resolution and framerate, plus settings for adaptive FPS

Titanfall 2 – no framerate options except for adaptive target

ADS study

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Game | Ironsights |  | Red dot | Scope |
| Far Cry 3 | Seems to be exactly the same as what I have. |  | Red dot appears and disappears in the sight picture when aiming vs. hipfiring. Reticle is positioned forward relative to the actual sight glass to mimic the real life effect of the ADS moving relative to the glass to represent the actual aim direction.  Zoom is just accomplished by altering the player’s FOV instead of any extra camera trickery. | After a certain point in the animation cycle, the gun model will disappear from view and the overlay will appear. Similar to what I had before, except the overlay will shift around when the player moves. Not sure how they ensure the aim still lines up, but I would assume it to involve the aim overlay rotating around the aim and camera axis, so it moves but does not appear distorted relative to the camera.  The scope is just a tube with transparent material for the lenses on the front and back. The inside of the scope is just black. |
| Titanfall 2 |  |  | The set of illuminated sights closer to the camera enables when the player activates ADS. The centre dot is at the far end of the scope. | Does the similar scope thing of having a rapid switch animation, disabling the weapon model and enabling a HUD overlay.  The overlay tilts on the Z axis when the player adjusts their aim from side to side, to provide a pleasant cosmetic effect without altering the reticle relative to everything else. This might be a very good idea. |
|  |  |  |  |  |

Personal idea:

For the scope, have the glass material use a custom shader with a 0-1 variable that shifts between opaque and translucent enough to see through based on the ADS status

Or have it be based on the angle of the scope relative to the player camera.

Or alternatively, do just have it appear instantly based on the scope angle. I was looking at the actual rifle scope I happen to have, and that’s kind of what it looks like.

Ideas for specific weapons

Pistol

1. Semi-auto 9mm pistol that takes Glock magazines
2. Glock 18

Rifle

1. Civilian bolt-action hunting rifle
2. Semi-auto AR-15
3. F90 assault rifle with underbarrel grenade launcher

Shotgun

1. Sawn-off over-under double-barrelled shotgun
2. Shortened Adler A110
3. M26 in standalone configuration

Things to fix

* EventSystem transitions are messed up, and will switch to options in hidden windows.
* Movement controller cosmetic tilting will jitter when moving forwards or backwards
* Gun ADS on handgun will fail to trigger if the player starts in the target range level, leaves and then loads into it again. This might not be worth fixing since in the final build the player will always start off in the main menu

Alternate ammunition type ideas

Shotgun

* Buckshot
* Slug
* Taser
* Beanbag
* Rock salt
* Flechettes
* Flares
* Flashbangs
* Tiny shells
* More capacity, less damage
* Only makes sense with a tube fed design, though
* Bolo rounds

Grenade launcher

* HEDP
* Airburst
* Gigantic shotgun shell
* Smoke
* Tear gas

Ideas for extra tools for the player

* Binoculars – View far-away enemies, in multiple vision modes (regular, thermal and something else)
* Hacking/EMP thing – Aim at an electronic device and fire to jam it.

HL2 weapons

1. Crowbar
2. Gravity gun
3. Pistol
4. Revolver
5. SMG
6. SMG grenades
7. OSIPR
8. OSIPR energy ball
9. Shotgun – single barrel
10. Shotgun – both barrels
11. Crossbow
12. Grenades
13. Rocket launcher
14. Bugbait

Weapons I plan on having in my game (to make sure I have an actually impressive arsenal)

I think 12 will be a nice round number. All of the FPS basics plus a few extra exotic/utility weapons.

Kick (activated with a dedicated button)

1. Pistol – semi-auto
2. Pistol – full-auto
3. Assault rifle – full-auto
4. Assault rifle – semi-auto
5. Shotgun – buckshot
6. Shotgun – Taser slugs
7. Grenade launcher – HEDP rounds
8. Hand grenades (or flashbangs)
9. .
10. .
11. .
12. .