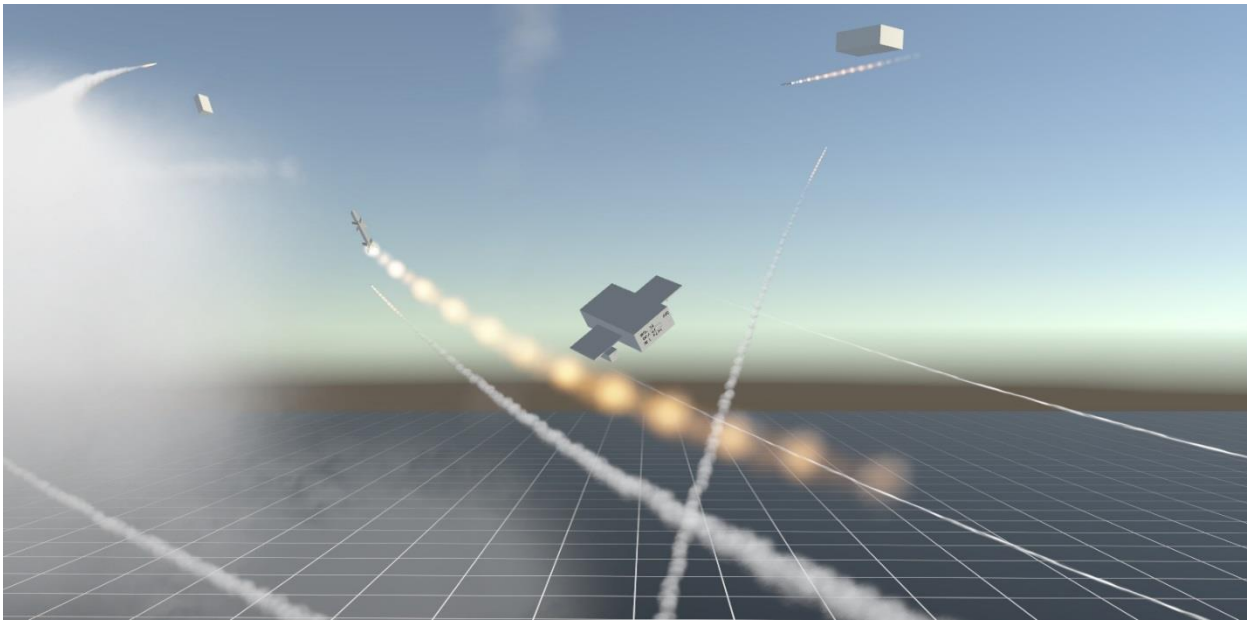
Ace Arcade Missiles v1.0



1 DESCRIPTION

Ace Arcade Missiles is a complete code and effects package for creating arcade style missiles.

- Detailed documentation for reference
- Two kinds of guidance: Pure pursuit and Lead
- Support for both Update and FixedUpdate based projects
- Many tweakable missile parameters such as seeker cone, motor acceleration, and turn rate
- Missiles can be set with a delayed activation to allow dropping before firing
- Effects manager that handles both ribbon trails and particle system trails, as well as explosions
- Spawned effects (trails and explosions) automatically clean themselves up
- Audio is handled automatically when assigned an audio clip with adjustable parameters
- Two types of missile launchers: Pod and Hardpoint
- Hardpoints can carry missiles externally, while pods launch from user defined tubes with reloadable magazines
- Pack includes example effects for particle and ribbon trails, basic sound effects, and explosions

2 CONTENTS

3 MAJOR COMPONENT DETAILS

This section details the specific settings of each parameter on the user facing included components and anything worth noting to ensure proper operation.

3.1 AA MISSILE

The core of the asset pack, this controls both missile flight and guidance. Missiles can be fired manually if they are already in the scene by calling the "Launch" function on them, but they were primarily designed to be fired from launchers.

3.1.1 Required Components

Rigidbody: A missile is required to have a rigidbody, but in most cases it can be ignored as the missile will override certain Rigidbody parameters on start.

This is used for collision detection, and requires that your target have a collider of some kind. If the missile starts to travel fast enough that it clips through targets, you might want to experiment with setting the collision detection to Continuous.

The mass, drag, and angularDrag of the Rigidbody are only used when the missile is dropping.

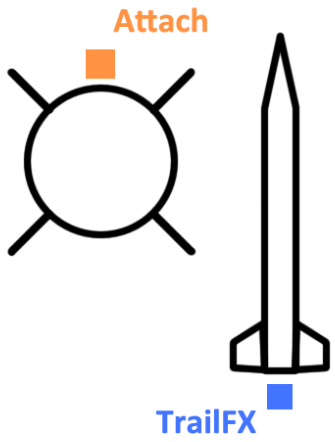
Capsule Collider: Missiles require a capsule collider. It's best to adjust the size and orientation of this yourself to match the missile, as the automatically created one will be default to a 1 meter sphere. Because the capsule collider covers the entire missile, it's best to not have any other colliders on the missile game object.

3.1.2 AA Missile Parameters

Movement Update Cycle: Run missile movement code in fixed update versus update. If you notice jittery movement by the missile during play, try changing this. Update is for projects where all movement is handled through the Update loop. Fixed Update is for physics based projects that rely on rigid bodies, forces, and prominently feature updates through the FixedUpdate loop.

Target Update Cycle: Runs guidance code in the update versus fixed update. Same as above, but for the target's movement. If set incorrectly, lead guidance will likely be thrown off and jitter.

Own Ship: This is typically assigned by the launcher and only needs to be assigned if manually launching a missile already in the scene. Set this to the launching game object (airplane, missile turret, etc.) to prevent the missile from colliding with the launching object. Only necessary to set



Attach Point: This is a Transform that the missiles uses as reference for where to attach to a hardpoint launcher. Not necessary for rocket launchers, but should be assigned regardless for good practice. If not assigned manually through the Inspector, you can alternatively create a GameObject called "Attach" and the missile will automatically use that. If neither is done, the missile will use its origin as an attach point.

Guidance Type: Either pure pursuit or lead. Pure pursuit will fly the missile directly towards the target. This is simpler and less expensive, but less effective. Lead on the other hand will have the missile intercept the target, reaching it faster and while being much more difficult to avoid.

Seeker Cone: How far off-boresight the missile can track a target. A wider angle means the missile is more likely to keep lock. If a target maneuvers outside of the seeker code, the missile will go dumb and fly straight until it either hits something or times out. A missile won't re-acquire its target if the target is lost. A larger seeker cone also allows the missile to make more dramatic maneuvers when leading the target as it'll be able to make bigger turns while still keeping the target in sight.

Seeker Distance: How far the missile can track a target. If the target falls outside this range, the missile will go dumb and fly straight until it either hits something or times out.

Override Initial Speed: When true, initial speed will be taken from either the velocity passed into the Launch function, or from the forward velocity of the missile after a drop launch if a drop delay is used. This is useful for missiles that you want to inherit their start speed from their launchers.

Initial Speed: How fast the missile is when it launches. Note that this does not inherit any velocity that the missile might have had before launch.

Motor Lifetime: When nonzero, allows the missile to accelerate for the time specified. Once the motor lifetime is up, the missile will continue on at whatever speed it was when the motor finished. Use this for missiles that you want to accelerate over time.

Acceleration: How much speed per second the missile gains while the motor is active.

Turn Rate: Degrees per second the missile can turn. The higher this value, the faster the missile can turn and the more difficult it is to evade.

Time to Live: How long the missile goes before self-destruct and the game object gets removed the scene.

Drop Delay: If greater than 0, missile will free fall for this many seconds and then activate after this many seconds have elapsed.

Eject Velocity: How fast the missile will be ejected downwards from its launch point.

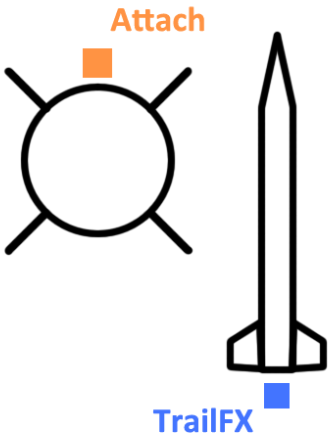
Gravity: Whether or not the missile should have gravity when dropping.

3.2 AA Missile Effects

Handles all the special effects for the missile. This includes trails, explosions, and sound effects.

3.2.1 AA Missile Effect Parameters

Trail Always On: When true, the missile trail effect will play continuously as long as the missile is active. When false, the trail effects stop playing when the missile's motor burns out. This is to emulate the way that real missiles only leave trails as long as their motor is burning, but for aesthetic reasons most arcade games ignore this. However, it's good to have choices so if you'd like to do it "realistically" the option is available.



Trail Fx Point: A Transform that is used as the reference for where the trail effect should play. If not manually assigned through the Inspector, the missile will search for a game object named "TrailFx" and use that. In the case neither is available, the trail effect will play from the missile's origin.

Trail Prefab: If the missile is using a TrailRenderer based trail, assign it here. It's not necessary the effect be a prefab, but for organizational purposes, it should be. If you want to mix a trail and particle system based trail, use separate effects and assign them separately. An AARemoveEffect component will be automatically attached to this trail so that the trail can remove itself smartly.

Particle Trail Prefab: If the missile is using a ParticleSystem based trail, assign it here. It's not necessary the effect be a prefab, but for organizational purposes, it should be. If you want to mix a trail and particle system based trail, use separate effects and assign them separately. An AARemoveEffect component will be automatically attached to this trail so that the trail can remove itself smartly.

Explosion FX Prefab: Effect to play when the missile impacts a target or, optionally, self-destructs. This effect should have "Play on awake" set to true. An AARemoveEffect component will be automatically attached to the effect so that it can remove itself smartly.

Play Explosion on Self-Destruct: When true, the explosion effect will play when the missile's TimeToLive expires. This is purely an aesthetic choice.

Mixer Group: Associates the generated audio sources with the specified mixer group.

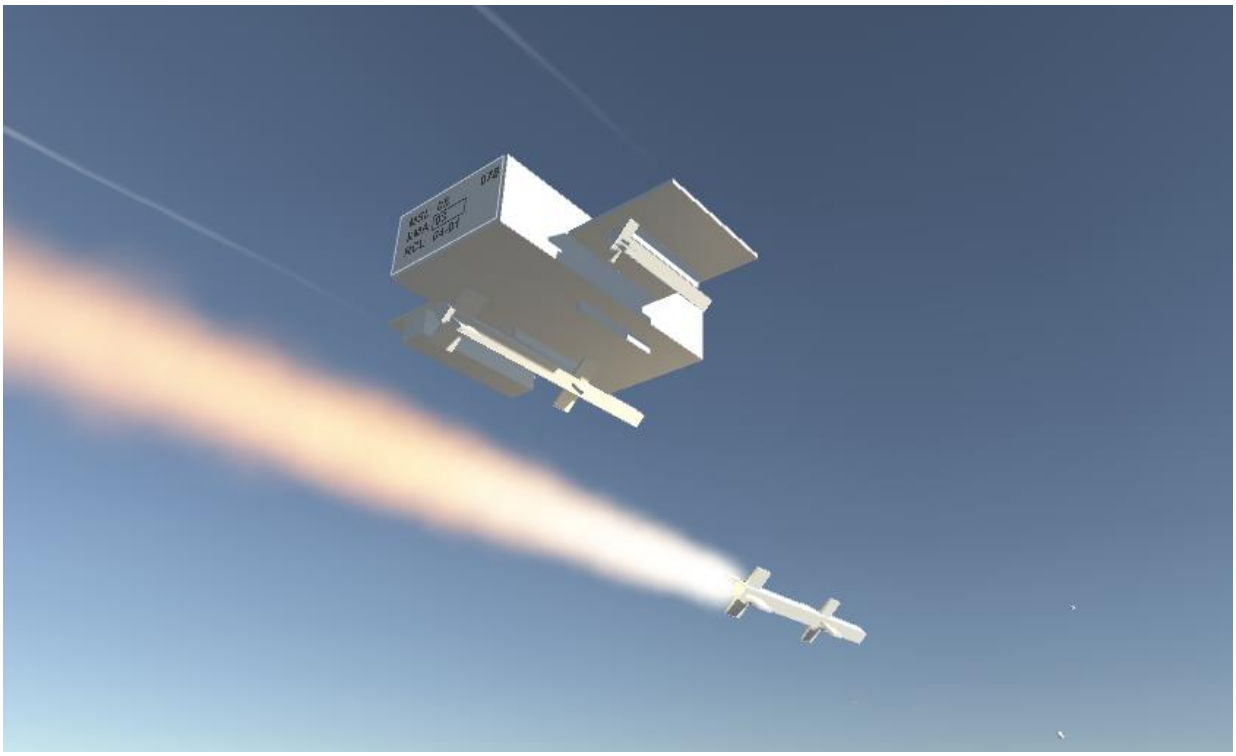
Fire Clip: Sound that gets played when the missile is launched. Will automatically create an AudioSource at runtime with the following values. Note that pitch is a random value between 0.8 and 1.5. This is to create variety in the fire sound for repeated firings.

- Output: **Taken from Mixer Group**
- Mute : False
- Bypass Effects: False
- Bypass Listener Effects: False
- Play On Awake: True
- Loop: False
- Priority: 128
- Volume: **Taken from Fire Volume**
- Pitch: *Random value between 0.9 and 1.3*
- Stereo Pan: 0
- Spatial Blend: 1
- Reverb Zone Mix: 1
- Doppler Level: 0
- Spread: 0
- Volume Rolloff: Logarithmic Rolloff
- Min Distance: **Taken from Fire Min Distance**
- Max Distance: **Taken from Fire Max Distance**

Loop Clip: Sound that the missile plays while flying. It's a constant noise versus the one time sound that the fire clip is. Both sounds are optional. Will automatically create an AudioSource at runtime with the following values:

- Output: **Taken from Mixer Group**
- Mute : False
- Bypass Effects: False
- Bypass Listener Effects: False
- Play On Awake: True
- Loop: True
- Priority: 128
- Volume: **Taken from Loop Volume**
- Pitch: 1
- Stereo Pan: 0
- Spatial Blend: 1
- Reverb Zone Mix: 1
- Doppler Level: 1
- Spread: 0
- Volume Rolloff: Logarithmic Rolloff
- Min Distance: **Taken from Loop Min Distance**
- Max Distance: **Taken from Loop Max Distance**

3.3 AA HARDPOINT



One of the two launchers that can be used to fire missiles. This one is designed for external carry and display of missile. The launcher will spawn missiles of the assigned prefab at given launch points. After a missile fires, a new missile will appear after a given reload time. To fire a missile from a launcher, simply call the Launch function on it.

AA Hardpoint and AA Pod both inherit from abstract class AA Launcher, so most parameters are shared between them.

3.3.1 AA Hardpoint Parameters

Own Ship: Assign this to prevent the launcher and missile from colliding with whatever launched/owns it.

Missile Prefab to Launch: The missile game object that will be instantiated by the launcher. While this parameter can be assigned to a missile already on the scene (and thus instantiating a copy of something already in the scene) this can cause issues with duplication of the trail effect. Only use missile prefabs.

Missile Count: Ammo count of the launcher. When it reaches 0, the launcher won't spawn a new missile on the Launch point.

Fire Delay: How long it takes for the missile on the launcher to respawn.

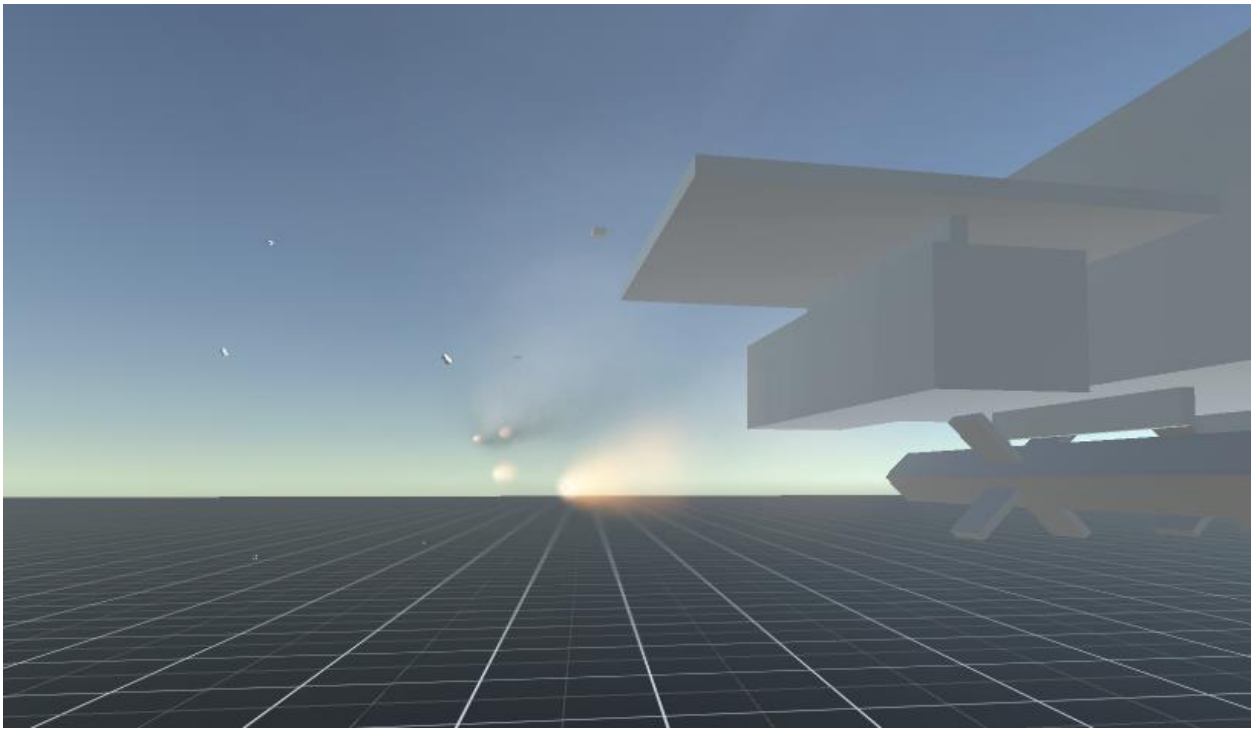
Launch Points: Array of Transforms used as reference positions for the missile to launch. If unassigned, list will be auto populated with child Game Objects that follow the naming scheme "Hp1", "Hp2", "Hp3", etc. If none are found, a single launch point will automatically be created at the launcher's center.

Mixer Group: Associates the generated audio source with the specified mixer group.

Fire Clip: Sound that plays when the hardpoint is fired. This will automatically create an AudioSource with the following values. Note that pitch is a random value between 0.8 and 1.5. This is to create variety in the fire sound for repeated firings.

- Output: **Taken from Mixer Group**
- Mute : False
- Bypass Effects: False
- Bypass Listener Effects: False
- Play On Awake: True
- Loop: False
- Priority: 128
- Volume: **Taken from Fire Volume**
- Pitch: *Random value between 0.8 and 1.5*
- Stereo Pan: 0
- Spatial Blend: 1
- Reverb Zone Mix: 1
- Doppler Level: 0
- Spread: 0
- Volume Rolloff: Logarithmic Rolloff
- Min Distance: **Taken from Fire Min Distance**
- Max Distance: **Taken from Fire Max Distance**

3.4 AA POD



The AA Pod simulates a rocket pod style launcher. Rather than the missile physically being placed on a hardpoint, they are dynamically spawned as needed. Pods take advantage of multiple launch points by cycling through each point on each successive firing to simulate different rocket tubes. Pods are unique in having “magazines” with their own size, count, and reload times. Magazine reloading is triggered automatically when all missiles are fired from the pod. Reloading can be called manually on the pod, but all missiles currently in the pod will be lost.

The AA Pod shares most of the same parameters as the AA Hardpoint. Listed below are options specific to the AA Pod.

3.4.1 AA Pod Parameters

Dispersion Angle: Randomized angle at which the missile comes out of the pod at. The higher the number, the more spread out the missiles will be.

Missile Count: For the AA Pod, this doubles as how many missiles can be fired before a reload is required.

Magazine Count: Number of times the pod can additionally reload after all missiles have been fired from the first salvo or a manual reload is triggered.

Magazine Reload Time: Time to reload a missile pod magazine. All missiles must be depleted to start a reload. Alternatively, you can manually call the "ReloadPod" function.

4 Minor Component Details

This section briefly describes the smaller components used by the Ace Arcade Missile Asset Pack to assist other things such as particle effects.

4.1 AA Enable Light On Effect Play

On the included trail particle system prefabs, these turn a light on when the effect is playing. It's expected to be used on a child Game Object of the trail effect. It must be specified if the parent Game Object is a TrailRenderer or PartcleSystem.

4.2 AA Jitter Direction

Used on particle systems to jitter the emitter in a smooth way. This allows for smoke trails that jitter in direction rather than being perfectly uniform. It approximates the look of minute course adjustments from the missile and variable wind along the path of the missile.

4.3 AA Remove Effect

Very basic script that removes a particle system once it's finished emitting. This is used to clean up trails after the missile has impacted, the particle system has been detached from the missile, and it stops emitting new particles.

4.4 AA Scrolling Trail

Used on TrailRenderer based missile trails to scroll the texture.

5 Scripting Reference

There are many publicly accessible functions in the scripts.

5.1 AA Missile

5.1.1 public void Launch(Transform newTarget)

Launch the missile at the given target. If no target is given, the missile will dumbfire. If the missile has a drop delay, use the Launch function with inherited velocity for the correct drop behavior.

5.1.2 public void Launch(Transform newTarget, Vector3 inheritedVelocity)

Launch the missile at the given target with an inherited velocity for correct drop behavior. If no target is given, the missile will dumbfire. It's recommended to use this function in general as it will work for missiles with and without drop delays.

5.2 AA Missile Effects

5.2.1 public void Explode()
Trigger the explosion effect and automatically detach any spawned trail effects.

5.3 AA Hardpoint

5.3.1 public override void Launch(Transform target)
Launches a spawned missile at the given target if possible. If no target is given, the missile will fire without guidance.

5.3.2 public override void Launch(Transform target, Vector3 velocity)
Launches a spawned missile at the given target. If no target is given, the missile will fire without guidance. Velocity is used to give a missile with a drop delay an initial velocity. Typical use case would be passing in the velocity of the launching platform.

5.3.3 public override void ResetLauncher()
If the missile prefab has changed, calling this function will delete the current loaded missiles and replace them with the new ones. Resets the ammo count of the launcher as well.

5.4 AA Pod

5.4.1 public override void Launch(Transform target)
Launches a spawned missile at the given target if possible. If no target is given, the missile will fire without guidance.

5.4.2 public override void Launch(Transform target, Vector3 velocity)
Launches a spawned missile at the given target. If no target is given, the missile will fire without guidance. Velocity is used to give a missile with a drop delay an initial velocity. Typical use case would be passing in the velocity of the launching platform.

5.4.3 public override void ResetLauncher()
Used to reset the ammo on a rocket pod. Intended to be used when the missile prefab is changed. Also resets the launcher's cooldown, magazine reload cooldown, missile count, and magazine count.

6 LICENSE