Thank you for your purchase! The most recent documentation can be found online. If you have any questions feel free to post on the forums or email support@opsive.com.

Attack

Moves to the closest target and starts attacking as soon as the agent is within distance.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

Charge

Charges towards the target. The agents will start attacking when they are done charging.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

agentsPerRow
The number of agents that should be in a row

separation
The separation between agents

attackDistance
The distance to stop charging and start attacking

Marching Fire

Move towards the target. The agents will start attacking when they are within distance.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

agentsPerRow
The number of agents that should be in a row

separation
The separation between agents

attackDistance
The distance to stop marching fire and continue attacking

Flank

Flanks the target from the left and right.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag

The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

dualFlank
Should the agents flank from both the left and right side?

attackDelay
The amount of time the left and right groups should wait after the center group has started to attack

approachDistance
Optionally set an extra distance that the agents should first move towards. This will prevent the agents from crossing in front of the enemies

separation
The distance that the agents should be separated while attacking

Ambush

Wait for the group of targets to pass before attacking.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

attackDelay
The number of seconds to wait after the enemies have passed before the agents start attacking

minAmbushDistance
The minimum distance that the agents can attack

Shoot and Scoot

Attacks the target and moves position after a short amount of time.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

agentsPerRow
The number of agents that should be in a row

separation
The separation between agents

timeStationary
The amount of time that should elapse before moving to the next attack point

minMoveAngle
When moving positions the agents will move based on a new random angle. The mimium move angle specifies the minimum random angle

maxMoveAngle
When moving positions the agents will move based on a new random angle. The maximum move angle specifies the maximum random angle

minAttackRadius
When moving positions the agents will move based on a new random radius. The minimum attack radius specifies the minimum radius

maxAttackRadius
When moving positions the agents will move based on a new random radius. The maximum attack radius specifies the maximum radius

Leapfrog

Search for the target by forming two groups and leapfrogging each other. Both groups will start attacking as soon as the target is within sight.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

separation
The horizontal separation between agents within the group

groupSeparation
The horizontal separation between the two groups

leapDistance
The distance of one leap

Surround

Surrounds the enemy and starts to attack after all agents are in position

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

radius
The radius of the agents that should surround the target

Retreat

Retreats in the opposite direction of the target

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

safeDistance
The distance away from the targets that is considered safe

Defend

Defends the object within a defend radius. Will seek and attack a target within a specified radius.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

defendObject

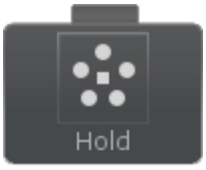
The object to defend

radius
The radius around the defend object to position the agents

defendRadius
The radius around the defend object to defend

maxDistance
The maximum distance that the agents can defend from the defend object

Hold



Defends the object within a defend radius. Will seek and attack a target for as long as it takes.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

defendObject
The object to defend

radius
The radius around the defend object to position the agents

defendRadius
The radius around the defend object to defend

Request Reinforcements

Requests reinforcements.

Reinforcements Response

Responds to a reinforcement request. Will move towards the requesting agent and start attacking as soon as the target is within distance.

targetGroup
The objects to attack. If blank the targetTag will be used

targetTag
The tag of the objects to attack. Will be used if targetGroup has no elements

waitTime
The amount of time to wait until the group starts to form

independent
Does this agent act independently?

listenForReinforcements
A list of agents that may call for reinforcements

Follow Orders

Tells the leader that the current agent is ready to follow its orders

leader
The leader to follow

Interfaces

One of the initial design decisions that we had to make with the Tactical Pack was to define what it means to attack and take damage. Attacking and taking damage means different things to different games. For example, attack could mean to shoot a gun or throw a melee punch. Our goal with the Tactical Pack is to make the code as generic as possible. To solve this we added two interfaces, IAttackAgent and IDamagable. This allows you to define exactly what it means to attack or take damage for your game while still being able to use the Tactical Pack.

Included with the demo scene is one implementation of IAttackAgent and IDamagable. When the agent

attacks they will instantiate a bullet prefab and that bullet will travel in the direction of the target. When the bullet hits the target it will call the IDamagable implementation to do the actual damage. It is expected that you will implement your own IAttackAgent and IDamagable components that fit your game. By structuring the Tactical Pack this way it succeeds in having a generic pack that works with any type of game.

The following methods need to be implemented with IAttackAgent:

```
// Returns the furthest distance that the agent is able to attack from.
float AttackDistance();

// Can the agent attack?
bool CanAttack();

// Returns the maximum angle that the agent can attack from.
float AttackAngle();

// Does the actual attack.
void Attack(Vector3 targetPosition);
```

The following methods need to be implemented with IDamagable:

```
// Take damage by the specified amount.
void Damage(float amout);

// Is the object currently alive?
bool IsAlive();
```

A* Pathfinding ProjectIntegration

The Tactical Pack tasks are integrated with the A* Pathfinding Project. The integration files are located on the integrations page because the Tactical Pack does not require the A* Pathfinding Project to work. This integration requires version 4.1.10 of the A* Pathfinding Project or later, and uses the IAStarAI interface provided by the A* Pathfinding Project.

Apex Path Integration

The Tactical Pack tasks are integrated with Apex Path. The Apex Path files are located on the integrations page because the Tactical Pack does not require Apex Path to work. To use the Apex Path tasks, you must first add all of the Apex Path components to your agent. This can be added via the Components -> Apex -> QuickStarts -> Navigating Unit menu option.

Playmaker Integration

The Tactical Pack tasks can use Playmaker to attack and receive damage. The Playmaker integration can be downloaded from the integrations page. To have an agent attack using Playmaker, add the AttackBridge component to your agent. This component allows you to specify which Playmaker event should be triggered when an attack should take place. Optionally a Vector3 may also be specified which indicates the direction that the agent should attack.

Receiving damage with Playmaker is similar to attacking. Add the DamagableBridge component to any GameObject that can receive damage, and the Damage Event will be triggered when the GameObject should

take damage. The Damage Amount and Health Variable Names are required. These variables map to a Playmaker variable which specify how much damage was received and the total health that the GameObject has left.

Third Person Controller Integration

The Tactical Pack is integrated with the Third Person Controller to allow the agents attack and receive damage using the framework. The integration files can be downloaded from the integrations page

As of version 1.2.1 of the Third Person Controller no separate pathfinding integration is required - the components are included in the Third Person Controller package. This includes Unity's NavMesh, A* Pathfinding Project, and Apex Path.

Support

We are here to help! If you have any questions/problems/suggestions please don't hesitate to ask. You can email us at support@opsive.com or post on the forum.