

# RoAR Assault Mapping Tutorial

By: Aaron Zabudsky & Doc\_EDo

This tutorial will aid Unreal 2003 mappers interested in making maps compatible with the RoAR Assault Mod.

**Note** that this tutorial is not meant to be a complete mapping guide. You need to know how to make basic **DM** and **CTF** maps and also learn **Triggers** and how actors cooperate using **EVENT** and **TAG** properties. Please visit the [Unreal Wiki](#) or [UDN](#) for more information about Unreal mapping concepts. For the most recent Assault files and information, please visit our web site at: <http://planetunreal.com/roar>.

## CONTENTS

<a href="#">1. Getting Started</a>	Assault map recommendations.
<a href="#">2. Spawnpoints</a>	Designing Attacker and Defender starting areas.
<a href="#">3. Objectives</a>	Adding FortStandard actors that must be captured to complete the map.
<a href="#">4. Touch Objectives</a>	How to make an objective that you need to touched to disable it.
<a href="#">5. Timed Objectives</a>	Making an objective that must be held for a certain length of time.
<a href="#">6. Destroyable Objectives</a>	How to make an objective that you need to shoot to disable it
<a href="#">7. Bot Support</a>	How to use UnrealScriptedSequences to help your bots navigate the level.
<a href="#">8. Final Changes</a>	Adding the list of assault objectives and screenshots for them etc..

---

## 1. Getting started

The level design is a bit different from DM, CTF or DOM. You need to focus more on flow in AS. But, CTF and DOM maps have team-specific starting areas, level objectives and multiple routes to each objective to prevent camping / spamming that is similar to AS.

Assault maps also differ in that one team is going after the goals, while the other team must protect them. As a mapper, you must take this into consideration, balancing your map to make the gameplay fair for both teams.

Before starting in UnrealEd, you may want to draw a quick diagram on paper of how you want your level to look. Make a note of the attacker and defender spawn areas and where you want to place your objectives. Make sure that defenders spawn closer to objectives than attacker do, but don't let that difference be too big.

When you have an idea of what you want to make, start UnrealEd 3.0 and make the geometry.

---

## 2. Spawnpoints

The next step is to add PlayerStarts in your level.

If defenders or attackers will ever change spawnpoints then all of their playersstarts must be *"TriggeredPlayerStart"*, not the normal PlayerStart actor!!!

Defenders and Attackers should spawn in separate areas. Make sure that defenders spawn closer to objectives than attackers do, but don't let that difference be too big. When you open the properties window for playerstarts you have an option under *"PlayerStart->TeamNumber"* which specify what team it belongs to. Use "0" for the Defending team and "1" for the Attacking team. The option *"bEnabled"* (only in TriggeredPlayerStart) can be set false/true and then when the actor gets triggered it will become enabled/disabled. This way players will change the spawn area/location.

It's good to have at least 6 playerstarts enabled at any time for each team, since 12 players will usually play your maps.

---

## 3. Objectives

Now we add an assault objective, using an actor called FortStandard. You need one FortStandard for each objective.

OpenPackage "RoARAssault.u".

Select *NavigationPoint-> JumpDest -> JumpSpot -> GameObjective-> DestroyableObjective-> **FortStandard***

(This is the main AS Actor ; contains functionality for AS objectives, and is a NavigationPoint as well)

Place it in your level where you want something to be activated (or close to it), but always on the floor.

A bullseye icon will appear on the map indicating the location of the fortstandard. Don't move it up and down, let it be as it is vertically, but you can move it sideways. Don't change the collision radius for it either, because it doesn't get touched or shot. You need special actors for that. The important properties are explained in the following table.

## FortStandard

<b>bFinalFort</b>	Ends mission when destroyed if true, even though some other objectives may not be completed. Set this to true if you want to allow possible skipping of objectives.
<b>bTriggerOnly</b>	Set true to make the objective disabled by touching.
<b>bTimed</b>	Set true to make the objective timed (must be trigger only too).
<b>DefensePriority</b>	In which order bots should defend the objectives (higher are attacked first).
<b>DefenseTime</b>	Amount of time (minutes) players have to complete the map.
<b>DestructionMessage</b>	Message to be shown when objective is destroyed.
<b>EndCamTag</b>	Tag of the SpectatorCam to be used as end camera.
<b>ObjectiveName</b>	Name of the objective, shown before destroyed message. (Don't make it bigger than 15 characters)
<b>DamageCapacity</b>	How much damage the fortstandard can sustain. Usually not used.
<b>BeaconOffset</b>	Offset for beacons, when an objective name should be show as located on the wall or a little bit higher than the actual FortStandard actor.
<b>TargetedActorTag</b>	Actor to be destroyed if it's a destroyable objective.
<b>DefenseScriptTags</b>	AI Script that is associated with this objective. (Defense points)



You need to test, learn and understand how to set properties and achieve correct results. Also study how other maps are made.

NOTE! The **FortStandard** actor alone will not provide the whole functionality, but must be combined with one of 3 Assault-specific actors.

## 4. Touch Objectives

Touch objectives are those that should be disabled by one single touch.

To make a Touch Objective set *FortStandard->bTriggerOnly=true*.

Then add **Triggers-> AssaultTeamTrigger\_RoAR** very close to the FortStandard.

Now you gotta make this trigger's Event = *FortStandard->Tag*.

Also make sure the collision radius of this actor is always bigger than the FortStandard's.

### **AssaultTeamTrigger\_RoAR**

**AssaultTeam** Team that may touch this actor. Set ATT\_Attackers when using with an objective,

**Enabled** In case you want to make this objective protected set FALSE. Later this actor should be triggered (with its tag) and with that the touching of objective enabled.

**Event** Set so that Event = *FortStandard->Tag*.



NOTE! This actor can be replaced by a normal trigger too, but this one will give you more control.

## 5. Timed Objectives

Timed objectives are those that should be disabled by standing in one position for a few seconds. This will simulate hacking, pushing, setting up a bomb, unlocking and everything that may take some time.

To make a Timed Objective set *FortStandard->bTriggerOnly=true*. AND *FortStandard->bTimed=true*.

Then add **Triggers-> AccTrigger\_RoAR** very close to the FortStandard.

Now you gotta make this trigger's Event = *FortStandard->Tag*.

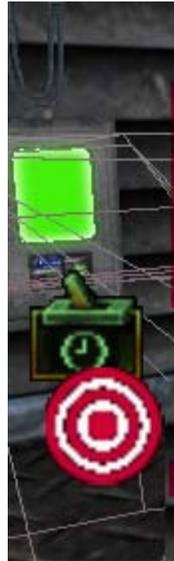
Make sure the collision radius of this actor is always bigger than the FortStandard's.

When this trigger is being touched a special warning message will appear on top of the screen.

Also set other properties.

## AccTrigger\_RoAR

<b>Enabled</b>	In case you want to make this objective protected set FALSE. Later this actor should be triggered (with its tag) and with that the touching of objective enabled.
<b>bResets</b>	When the touching is interrupted the time will reset. False recommended.
<b>bShowplayerName</b>	If true the touching player's name will show in the warning message.
<b>FailedMessage</b>	If bResets = true and touching gets interrupted this message will show.
<b>MaxAccTriggerTime</b>	Total Time needed to get this trigger to send it's event and disable an objective.
<b>WarningMessage</b>	Message to be displayed when someone is touching this actor
<b>Event</b>	Set so that Event = <i>FortStandard-&gt;Tag</i> .



AccTrigger_RoAR Properties	
- AccTrigger_RoAR	
- bEnabled	True
- bResets	False
- bShowPlayerName	True
- FailedMessage	
- MaxAccTriggerTime	2.000000
- Message	
- WarningMessage	is hacking the security panel.
+ Advanced	
+ Collision	
+ Display	
- Events	
- Event	entry1
- ExcludeTag	...
- Tag	AccTrigger_RoAR

## 6. Destroyable Objectives

Destroyable objectives are those that should be disabled by shooting.

To make a Destroyable Objective set *FortStandard*->**bTriggerOnly=false**. AND *FortStandard*->**bTimed=false**.

Then add **Mover**-> **DestroyableActor\_RoAR** (as you add any other mover) where it needs to be, but still it's own *FortStandard* should be close to it.

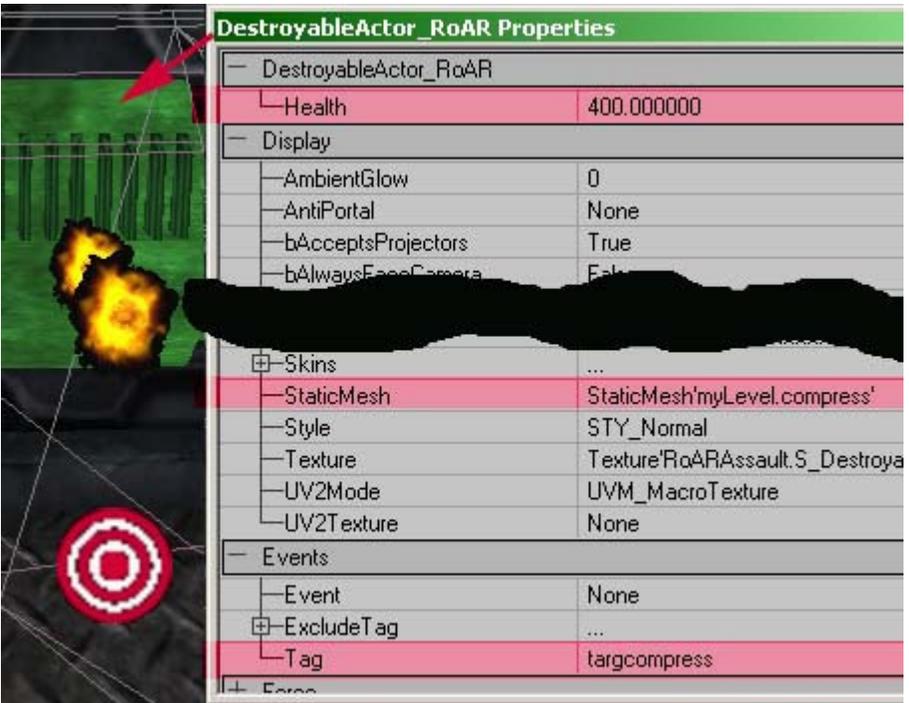
Now you gotta make this trigger's **Event** = *FortStandard*->**Tag**. You also must make **Tag** = *FortStandard*->**TargetedActorTag**.

Also set **Display**= **StaticMesh**, so that the *DestroyableActor\_RoAR* has a shape (in this case compressor), and give it some **health**.

Also set other properties.

### **DestroyableActor\_RoAR**

<b>Health</b>	How much damage this actor can take before being destroyed.
<b>StaticMesh</b>	This decides what form and shape the actor will have.
<b>Event</b>	Set so that Event = <i>FortStandard</i> -> <b>Tag</b> .
<b>Tag</b>	Set so that Tag = <i>FortStandard</i> -> <b>TargetedActorTag</b> .



DestroyableActor_RoAR	
Health	400.000000
Display	
AmbientGlow	0
AntiPortal	None
bAcceptsProjectors	True
bAlwaysFaceCamera	False
Skins	...
StaticMesh	StaticMesh'myLevel.compress'
Style	STY_Normal
Texture	Texture/RoARAssault.S_Destroya
UV2Mode	UVM_MacroTexture
UV2Texture	None
Events	
Event	None
ExcludeTag	...
Tag	targcompress

## 7. Bot Support

Bots defend/attack the fortstandard with the highest priority first and then the next in line. In this case defense priority is ignored and might result in some weird behaviour.

If bots are unable to reach the objective for some reason they will roam the level like a normal DM map and pickup items along the way.

Defending bots will defend the objective with their lives :p.

To make them stand at strategic positions you need to place  
*Keypoint* -> *AI Script* -> *ScriptedSequence* -> ***UnrealScriptedSequence***  
in that place.

You should assign it to a Objective by setting the Objective's *DefenseScriptTags* property to match the *UnrealScriptedSequence* actors' **Tag**. All *UnrealScriptedSequence*'s that serve as defense points to the same objective must have the same Tag.  
*UnrealScriptedSequence* actors that aren't assigned to a Objective are used by bots sent on freelance.

If you do not have any defense points for a FortStandard bots will stand/defend close to the FortStandard.

If you have more than one FortStandard you need to create sets of defense points for each one.

---

## 8. Final Changes

### - Sequences of Events and Effects

You will need to use *Dispatcher\_RoAR* to send events to doors and other actors. Also you will need to spawn some effect like breaking wall etc. Actors needed for that are easy to understand and easy to use. Check out other maps for reference.

### - Assault Info

To create a mission briefing you will need to place *Info*-> ***AssaultInfo*** anywhere in you level.

After that set following properties:

- *ObjShots* = Texture for screenshots

- *ObjDesc* = Description for each objective. What needs to be done?

- *ObjectiveObjectName* = Here you need to put *Object*->*Name* of a Fortstandard actor (for exaple Fortstandard12).

- *NumObjShots* = total number of objectives.

### - Spectator Camera / EndCamTag

To create the end of level view you need to add a spectator camera. The *SpectatorCam* actor is found under *Keypoint* in the actor browser. Set the *FortStandard*->*EndCamTag* to the *Tag* of the *SpectatorCam*.

This is directional so make sure it face in the correct way.

### - Level settings

Under the "LevelInfo" section, set both the "DefaultGame" and "PreCacheGame" properties to:  
*RoARAssault.xAssault*

Also set other properties like name, number of bots etc.

---

Good Luck!