



MIGHT & MAGIC
HEROES VI

Storytelling Tools Guide

Version 1.0
October 26, 2011

© 2011 Ubisoft Entertainment. Developed by Black Hole Entertainment.

Introduction

The present guide will teach you how to create a story-driven Single Player Map.

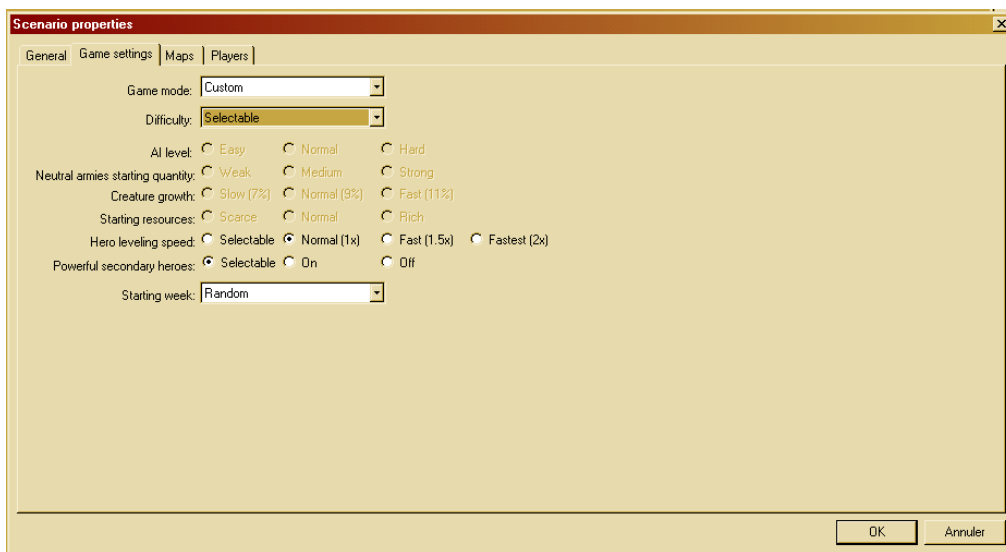
1. Getting Started

You can find the Scenario Editor in your Start menu, in *Ubisoft > Might & Magic Heroes VI > Start the Heroes VI Scenario Editor*.

Refer to the steps presented in the Quickstart Guide to create the layout of your map.

Click on *Scenario > Properties*.

In **Game Settings** you can set the game mode. Set the game mode as Custom since we are not doing a multiplayer map. In-game, your map will appear in the Scenario category.



2. Heroes and Enemies

In a story-driven map, you will want to create powerful enemies and reward the player by giving him bonuses to his stats, skills and experience.

You can change the Heroes' starting armies by opening their Properties (click on the Hero and press "Enter").

To edit their attributes and skills, open the Trigger Editor.

Under **Event trigger**, create a new trigger. Call your trigger *Heroes starting stats* or something similar. Leave **pCondition** as *Nothing* since you want it to take effect right from the beginning of the map. In **pAction**, select *Array* to be able to set a chain of actions.

Hint:

If you forgot to select *Array* and already set an action, don't worry. If you change **pAction** to *Array*, the action you previously created won't be deleted and will instead become the first item of the Array. It is recommended to always set **pActions** as *Array*.



For each of the following actions, you'll need to pick up the Hero to which they apply. You'll see an item called **pEntity** with a field "eid" (for "Entity's ID"). Click on the field in order to select the Hero directly on the map.

Attributes

Use the trigger called **Scenario_ModifyHeroAttributes** to change the attributes of the Hero. If you fill out the "add" part, the selected number will be added to the selected attribute. If you fill out the "multiply" part, the selected number will be multiplied with the Hero's actual stat.

Experience

The **Scenario_AddXP** trigger can be used to add experience to a Hero.

Level

The **Scenario_SetLevel** trigger can be used to set the level of a Hero. Only higher levels affect the Hero, lowering the level of a Hero with this trigger is not possible.

Skills

The **Scenario_SetSkill** trigger can be used to add a skill to a Hero. The trigger does not check magic schools, requirements or affinity, so Implosion (tier 3 Prime) can be added to a stronghold might Hero for example.

Important: Only use abilities located in the folders "Avatar", "PassiveAdventureAbilities" and "PassiveCombatAbilities".

Once you're done, don't forget to click on OK in the bottom-right corner of the Trigger Editor's window to save your changes.

Advanced Technique:

Sometimes you'll want to have a particular Hero stay at a given location to guard the path.

You can use the *Scenario > AdventureAI > AdventureAI_ToggleSkirmishAI* trigger to disable a player. This disables all Heroes for the selected player.

Unfortunately you cannot disable the AI for only one Hero of a given player. If you want to create a stationary Hero, you can put him/her under the control of a separate player and disable the Skirmish AI for this player.

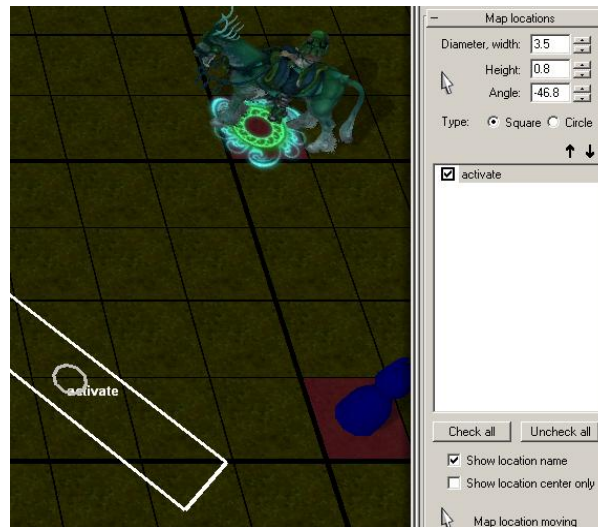
You can use Triggers to directly control the behaviour of each Hero, but that will be the topic of a future guide.

3. Events

The most basic storytelling tool at your disposal is called an Event. These are things that happen when the Hero enters a certain area of the map (Map Events), or after a certain number of turns (Timed Events).

a. Map Events

In order to create a Map Event, you first need to set the area in which the Event will take place. It is called a Map Location.



To create a Map Location, go to the **Selection panel** and scroll down until you find the **Map locations drawer**. Select the Map location tool and hover above the terrain with a Square type location. The size of the location can be changed by holding down the middle mouse button and dragging while the Shift key is pressed.

Click on the map where you want the Event to happen to place a rectangular location. Click Space to deactivate the Map location tool and return to the default select cursor.

To change the location's name select it from the location list on the sidebar, then after a moment click it again (or press F2 when the location is selected), then edit its name.

Hint:

Selecting View/Always show map locations from the menu will keep the locations visible even when the Map locations tool is not active.

Already-placed locations can be moved, resized and rotated by selecting the Map location tool, then grabbing the locations center with the cursor and then:

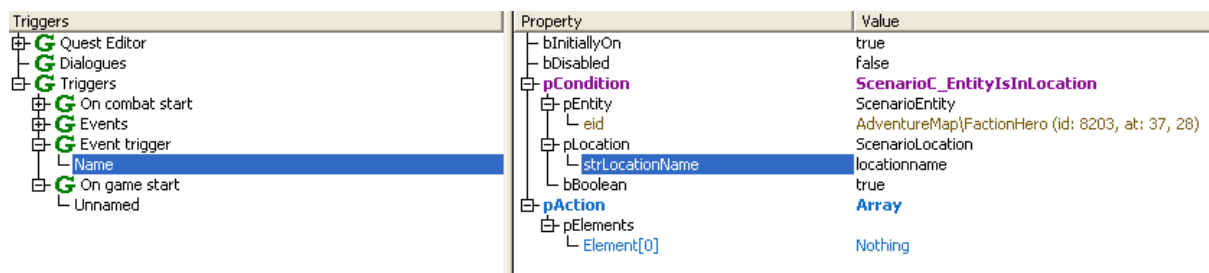
- moving it by dragging with the left mouse button,
- resizing it by Shift + left mouse button dragging,
- rotating it by Ctrl + left mouse button dragging.

Deleting a location can be done by pressing delete when the location is selected with the tool above the map, or by selecting the location from the list and pressing delete.

We want to display a classic Text Box when the player enters the Map Location. To do this you need to open the Trigger Editor.

Add a new trigger to the Event trigger branch on the left side. To fill it out:

1. give it a name,
2. leave **bInitiallyOn** on *true* and **bDisabled** on *false*,
3. set **pCondition** as *Scenario > Conditions > ScenarioC_EntityIsInLocation*,
4. fill out the condition's first parameter (**pEntity/eid**) by clicking "Nothing" next to it. The trigger editor disappears to allow entity selection and then reappears when an entity is left clicked. Select the chess piece symbolising the player's main Hero (*FactionHero*)
 Note: The *pEntity/eid* is the entity that will trigger the execution of this scenario trigger when it steps into the location. You can make sure the event will only trigger for a specific Hero.
5. fill out the condition's 2nd *strLocationName* parameter with the name of the location you placed earlier.



Select the "Scenario_DisplayText" trigger for the action's first element and set the text in the test editor by clicking on the empty line next to the "localizableText" field.

Set the player parameter as well, according to whom you want the text displayed (so you can set an Event so that the player receives a warning or resources when an enemy Hero enters a certain location).



b. Timed Events

Timed Events occur X turns after the beginning of the map, or X turns after another event. The “*Scenario_RestartTrigger*” can be used to start or restart a turned off (“*bInitiallyOn*”= *false*) trigger X days later. Set the name of the trigger you want to restart and set the “*DayDelay*” parameter as the delay.

Example A:

- To start an Event 3 days after the beginning of the map, create your Event under **Event Trigger**. Name it ExampleA and turn it off (“*bInitiallyOn*”= *false*).

Then create another Trigger and set *pAction* as *Scenario > Scenario_RestartTrigger* with *strTriggerName* = ExampleA and *DayDelay* = 3. This means that at the beginning of Day 3, your Event “ExampleA” will start.

Important note:

When located under **On game start**, Triggers cannot be restarted or turned on. They automatically start at the beginning of the game.

Example B:

- To start an Event 3 days after the defeat of an enemy Hero, create your Event under **Event Trigger**. Name it ExampleB and turn it off (“*bInitiallyOn*”= *false*).

Then create another Trigger under **On game start** and set *pCondition* as *Scenario > Conditions > ScenarioC_EntityKilled* by Player. Set *pEntity / eid* as the Hero the player needs to defeat and leave *bBoolean* as *true*. Set *pAction* as *Scenario > Scenario_RestartTrigger* with *strTriggerName* = ExampleB and *DayDelay* = 3. That means 3 days after the Hero’s defeat, your Event “ExampleB” will start.

The “*ScenarioC_IsDayOfTheWeek*” trigger checks the number of the day of the week. The following picture shows a trigger that is set up to add 1000 gold to player 1 every third day.

Triggers	Property	Value
Quest Editor	bInitiallyOn	true
Dialogues	bDisabled	false
Triggers	pCondition	ScenarioC_IsDayOfTheWeek
On combat start	Day	3
Events	bBoolean	true
Event trigger	pAction	Array
Name	pElements	
Trigger2	Element[0]	Scenario_AddResource
Lost	dwNum	1000 (VarNumeric)
DayofWeek	f	1000
On game start	strName	Gold
Unnamed	Resource	Player 1
	Player	Player 1
	Element[1]	Scenario_RestartTrigger
	strTriggerName	DayofWeek
	DayDelay	1 (VarNumeric)
	f	1
	strName	

Important: Note that the trigger restarts itself 1 day later so it will wait for the 3rd day of the next week. If the DayDelay parameter had been set to 0, then the trigger would end up in an infinite cycle, so be very cautious with restart actions and self-restarting triggers.

Hint:

While you're using the Trigger Editor, you can use Ctrl+F to open the search engine. It will help you find specific Triggers or values.

4. Advanced Events

Sometimes you'll want Events to be more than text. Here are some common examples of Events you can use in your maps:

a. Starting a battle

Let's say you want the player to fall into an ambush when he enters a Map Location. Place the army you want the player to fight somewhere on the map, in a place the player cannot reach. Customize the army by editing its Properties.

Create a new Event Trigger. Select *Scenario > Scenario_StartCombat* as the action. This will initiate combat between two entities on a combat map set. Select the desired combat map set, then set the attacker and the defender.

Triggers	Property	Value
Quest Editor	bInitiallyOn	false
Dialogues	bDisabled	false
Triggers	pCondition	ScenarioC_IsAllWindowsHidden
On combat start	bBoolean	true
Events	pAction	Array
Event trigger	pElements	
Name	Element[0]	Scenario_StartCombat
Trigger2	CombatMap	Jungle_OrcForestRuins
On game start	pAttacker	ScenarioEntity
Unnamed	eid	Sanctuary\Units\Kappa (id: 8606, at: 39, 31)
	pAttacked	ScenarioEntity
	eid	AdventureMap\FactionHero (id: 8203, at: 37, 28)

b. Gain/loss of resource

Sometimes you'll want the player to gain or lose some resources. Maybe he received some money from his king to help his war efforts, or on the contrary was robbed during his sleep?

You can use the action *Scenario > Scenario_AddResource* trigger to add or remove resources from the player.

The picture below shows a trigger that adds 5000 gold to player 1 when the Kappa stack dies. You can use negative number to remove resources from the player (when the Hero dies for example).

Triggers	Property	Value
Quest Editor	bInitiallyOn	true
Dialogues	bDisabled	false
Triggers	pCondition	ScenarioC_IsDead
On combat start	pHero	ScenarioEntity
Events	eid	Sanctuary\Units\Kappa (id: 8606, at: 39, 31)
Event trigger	bBoolean	true
Name	pAction	Array
Trigger2	pElements	
Lost	Element[0]	Scenario_AddResource
On game start	dwNum	5000 (VarNumeric)
Unnamed	f	5000
	strName	Gold
	Resource	Gold
	Player	Player 1

Advanced Technique:

Sometimes you'll want an Event to take place right after another event. In the case of an ambush, you might want to display a text explaining what happened before the battle starts.

Carry out the steps explained in 3.a. to create a Map Event. Then add a new element to the action, after the DisplayText action: right-click on the element and select "insert Element as last". Select "Scenario_RestartTriggerMs" as the second element and set its name parameter to "Trigger2" (this is the name of the new trigger that will be activated) and its ms parameter to 100.

Triggers	Property	Value
Quest Editor	bInitiallyOn	true
Dialogues	bDisabled	false
Triggers	pCondition	ScenarioC_EntityIsInLocation
On combat start	pEntity	ScenarioEntity
Events	eid	AdventureMap\FactionHero (id: 8203, at: 37, 28)
Event trigger	pLocation	ScenarioLocation
Name	strLocationName	locationname
On game start	bBoolean	true
Unnamed	pAction	Array
	pElements	
	Element[0]	Scenario_DisplayText
	localizableText	Text
	Identifier	Trigger.Text.27
	Player	Player 1
	Element[1]	Scenario_RestartTriggerMs
	strTriggerName	Trigger2
	ms	100 (VarNumeric)
	f	100
	strName	

Create a new trigger named "Trigger2" in the Event trigger folder on the left side. Set its **bInitiallyOn** parameter to *false*. This means the trigger will not run unless it is activated or restarted. Note: the difference between activation and restart is that a trigger can be activated one time only, but can be restarted multiple times. *RestartTrigger* and *RestartTriggerMs* activate the trigger as well.

Set the new trigger's condition to *Scenario > Condition > ScenarioC_IsAllWindowsHidden*.

This condition makes the trigger wait until all windows close for the actions to run. In this case this window is the text box appearing by the DisplayText action.

5. Dialogue Scenes

Dialogues are conversations between the player and an NPC. A dialogue consists of two parts: the **Dialogue Conversation window** and the **Dialogue Options window** after it.

The Dialogue Conversation window shows the Player a conversation between two (or more) participants and it does not need Player input – however, it can be skipped. The Dialogue Options window appears after the Dialogue Conversation window and it contains the pictures and names of the participants, all spoken texts of the dialogue Conversation, options and their rewards and buttons for the Player's input (answer or choose options).

A dialogue can be created in the trigger editor; it is not used by any NPC by default. The dialogue can be attached to any entity on the Adventure Map; it can be set in the property window of the entity. One entity can have only one dialogue, but a dialogue can be attached to several entities.

There are trigger functions to set a different dialogue for an entity or to remove a dialogue from an entity. When the dialogue is removed, the entity functions as usual (attack, trade, etc.). There is also a trigger action for starting a dialogue, so the Player does not always need to order his Hero to interact with an NPC.



a. **Creating and setting up the dialogue**

Right click on the “Dialogues” element on the left side of the trigger editor. Select “Add dialogue”. Add 2 talkers and a “talk”, and inside the “talk” element add 2 “lines”. These can be done by right-clicking on the talkers, talks and lines elements respectively and selecting the add option.

Property	Value
Dialogue Starts when	Entering Guarded Area
Talkers	
Talker	Unnamed
Type	Visited
Color	Default
Head slot	Left
Specific entity	GetEntityVariable
strVariableName	
Unique id	31
Display priority	No
Talker	Unnamed
Type	Visited
Color	Default
Head slot	Left
Specific entity	Nothing
Unique id	32
Display priority	No
Talks	
Talk	
Dialogue windows	Both
Lines	
Line	
Talker	No talker
Talker head slot	Default
Talk text	Unedited
Talk text sound	No sound
Unique id	34
Duration	0
Line	
Talker	No talker
Talker head slot	Default
Talk text	Unedited
Talk text sound	No sound
Unique id	35
Duration	0
Options	
Onclose	Nothing
Unique id	28
Music	None

Start naming the talkers: click on the “Unnamed” field next to the “Talker” elements and set their names. This will be used in the Lines’ Talker parameter to determine who will say that particular line. Set each type to Specific. You can also set the colour of talkers: when kept on default, the player colour will be used. The head slot determines where the talker appears in the dialogue: on the left side or the right. When someone says his line where a previous talker spoke, the portrait of the old talker will be replaced with the new one.

The “**Specific entity**” parameter can be set in 3 different ways:

- Select *Scenario > ScenarioEntity* and then you can select an entity on the adventure map.
- Select *Variable > VarEntityType* and then you can select any entity from the entity library (to be able to make someone speak in the dialogue who is not on the map).
- Select *Scenario > Variable > GetEntityVariable* and then you can enter a name of an entity that is stored in an Entity Variable. Use *Scenario > Variable > SetEntityVariable* trigger to save an entity into a variable. Dead entities can speak this way in dialogues.

Talk / Dialogue Windows

This setting determines which parts of the Dialogue are shown to the Player. “Both” means the Dialogue Conversation window starts, and when it ends or when the player skips it, the Dialogue Options window is shown.

If “Conversation only” is selected, only the Dialogue Conversation window is shown, and when it ends or when the player skips it the Player can continue playing on the Adventure Map.

If “Options only” is selected, the Dialogue Conversation window is skipped and only the Dialogue Options window is shown.

Let’s keep it on “Both” to be able to demonstrate the dialogue options.

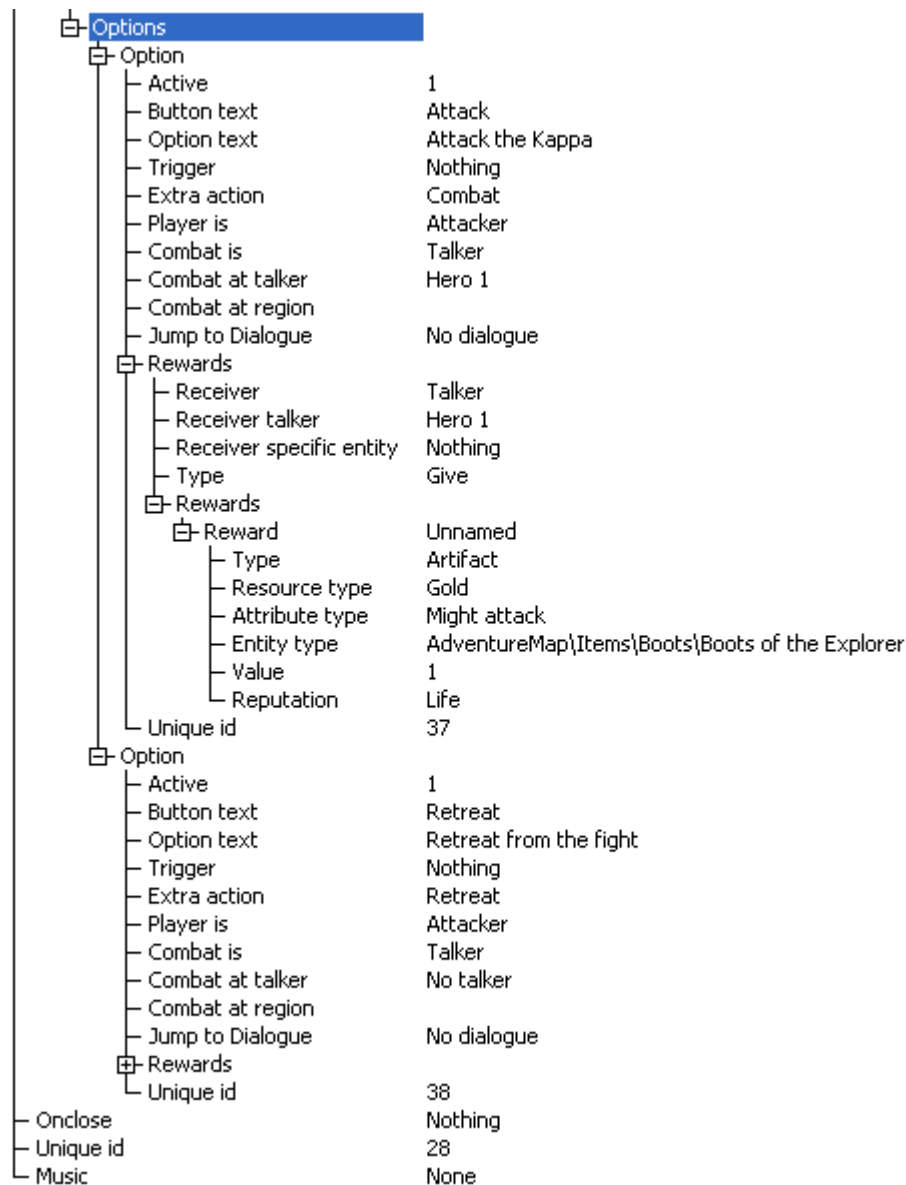
Talk / Lines

Set up the two lines to make both talkers say something. See the following picture for example:

Triggers	Property	Value
Quest Editor	Dialogue Starts when	Entering Guarded Area
Dialogues	Talkers	
Dialogue1	Talker	Hero 1
Triggers	Type	Specific
On combat start	Color	Blue
Events	Head slot	Left
Event trigger	Specific entity	ScenarioEntity
Name	eid	AdventureMap\FactionHero (id: 8203, at: 37, 28)
Trigger2	Unique id	31
Lost	Display priority	No
DayofWeek	Talker	Kappa
On game start	Type	Specific
Unnamed	Color	Green
	Head slot	Right
	Specific entity	ScenarioEntity
	eid	Sanctuary\Units\Kappa (id: 8606, at: 39, 31)
	Unique id	32
	Display priority	No
	Talks	
	Talk	
	Dialogue windows	Both
	Lines	
	Line	
	Talker	Kappa
	Talker head slot	Default
	Talk text	I am a kappa and I will attack you!
	Talk text sound	No sound
	Unique id	34
	Duration	0
	Line	
	Talker	Hero 1
	Talker head slot	Default
	Talk text	I am prepared to die!
	Talk text sound	No sound
	Unique id	35
	Duration	0
	Options	
	Onclose	Nothing
	Unique id	28
	Music	None

Now let's create the options after the conversation.

Add 2 options by right-clicking on the Options element inside the Talk element and select Add item. Set up the parameters as shown on the following picture:



There will be two buttons after the dialogue: one with an “attack” option (which automatically starts the fight between the Hero and the Kappa with the Hero as the attacker), and one with a “retreat” option. The attack option has an artifact reward that will be given to the Hero when he presses the button. If you just want to show the reward (and give it after the fight is won), select the “Show” option for the “Type” parameter of the Reward.

Finishing steps

Use the *Scenario_StartDialogue* trigger action to start the conversation:

Triggers	Property	Value
Quest Editor	bInitiallyOn	true
Dialogues	bDisabled	false
Dialogue 1	pCondition	ScenarioC_EntityIsInLocation
Triggers	pEntity	ScenarioEntity
On combat start	eid	AdventureMap\FactionHero (id: 8203, at: 37, 28)
Events	pLocation	ScenarioLocation
Event trigger	strLocationName	locationname
Name	bBoolean	true
Trigger2	pAction	Array
Lost	pElements	Scenario_StartDialogue
DayofWeek	Element[0]	Dialogue1
Dialogue	DialogueId	ScenarioEntity
On game start	pVisitor	ScenarioEntity
Unnamed	eid	Nothing
	pVisited	ScenarioEntity
	eid	Nothing

Important note:

Currently Dialogue Scenes cannot be used in Multiplayer maps.

Quests

In Heroes VI, a Quest is divided into four elements.

- i. **Quest name**
The name of the quest in the Quest Log.
- ii. **Quest description**
A short text explaining what the Quest is all about.
- iii. **Objectives**
The various steps needed to complete the Quest.
- iv. **Reward**
What the player gains by completing the Quest.



a. Quest parameters

Quests are the goals of the map, and are comprised of objectives. Every quest is considered either primary or secondary: usually failing a primary quest leads to defeat, completing all primary quests lead to victory and secondary quests are optional goals to gain some kind of bonus.

Quests can have rewards. A reward is given to the player when he completes all objectives. By default, artifact, experience, attribute, reputation and creature rewards are given to the Main Hero.

Quests have the following parameters:

- Name – This is the name of the quest in the scenario editor. The name is used to refer to the quest in trigger functions.
- Title – This is the title of the quest as it will appear in the game itself.
- Description – This is the description of the quest, which appears in the game.
- Type – This is the type of the quest. Can be Primary or Secondary (default: Primary).
- Objectives – Here, the user can add multiple objectives that they already created in the quest editor. These objectives will be the starting objectives of the quest. Later, additional objectives can be added to the quest by trigger functions.
- Rewards – Here, the user can add a maximum of 3 rewards to the quest. The parameters of a reward are the following:
 - Type – It can be set to Artifact, Creature, Resource, Attribute, Reputation or Experience.
 - Value – A non-negative integer that determines the value/quantity of the reward (for example 20 Vampires).
 - Give / Show Reward – This parameter determines whether the reward will be given to the Avatar automatically upon quest completion. If this parameter is set to Show Reward, the rewards won't be given to the Avatars, but the icons of the rewards will appear in the quest window; the user must grant them separately through trigger functions to apply them to the player. This course of action makes it possible to give the rewards to an entity other than the Avatar (for example creatures to a town).

b. Objectives parameters

An objective is a part of the quest that can be considered a "step". Quests can have several objectives. If any objective is failed, the whole quest is failed. When all objectives are completed, the quest becomes completed. It is not possible to "salvage" a failed objective or "botch" a completed quest; an objective cannot be changed from completed to failed or from failed to completed.

Objectives have the following parameters:

- Name – This is the name of the objective. The name is used in the editor only, to refer to the objective in trigger functions or in quests.
- Title – This is the title of the objective as it will appear in the game itself.
- End Game Statistics Value – This is a non-negative integer (default: 1) that is used to determine the value of the objective for the completion part of the end game statistics

window (the player's "score"). If an objective is completed, player(s) who have finished the quest with the objective will get this value.

- Quest Helper Regions (Trigger) – Here, the user can add several regions for the objective. A region can be set as "Preset" (regions created on the map) or "Variable" (regions saved into a variable). These regions serve as a guide for the player. Each region has a name that is used to refer to it in the trigger editor.
- Objective Progress – This is where the user can make the objective progressive. A progressive objective is for example the "Capture 3 Towns" objective. The parameters are the following:
 - Make Objective Progressive – This is a 0/1 option (default: 0). If it is set to 1, the other parameters have effect.
 - Progress Variable – This is a numeric variable that indicates the progress value. Level designers can increase this variable with other trigger functions.
 - Goal Value – This is a non-negative integer (default: 3). This is the end value of the progression.
 - Progression Update Text – This is a string. For example "Towns captured".

If the objective is set to be progressive, then a new line appears in the quest window below the title of the objective with the following syntax:

"Starting Value" / "Goal Value" "Progression Update String" (for example: "0/3 Towns captured"). If the value of the Progress Variable changes, then the progress value in the quest window is updated automatically.

When the progression value reaches the goal value and the Auto complete parameter is set to 1, the objective becomes completed automatically (and also, an adventure map log event is generated). The progression value cannot exceed the Goal value, and it cannot be less than 0.

c. Activating a Quest

Use the AddQuest trigger and choose which quest you want to give to the player.

Important note:

Currently Quests cannot be used in Multiplayer maps. This means you can only activate Quests for the human player (Player 1 by default).

Appendix: Example of a typical Quest

Check the following pictures for a typical quest setup:

Triggers	Property	Value
Quest Editor	Title	Title of objective 1
Quests	End game result	1
Quest	Trigger	ScenarioLocation
Objectives	strLocationName	questhelper_location
Objective1	Make progress	1
Objective2	Progress starting trigger	GetNumericVariable
Dialogues	strVariableName	progress
Triggers	Progress goal value	3
On combat start	Progress update text	towns captured
Events	Auto show objective progress	1
Event trigger	Auto complete	1
AddQuest	Unique id	42
CompleteObjective1		
CompleteObjective2		
On game start		

This is the first objective. It has quest progress and quest helper location.

Triggers	Property	Value
Quest Editor	Title	Title of objective 2
Quests	End game result	1
Quest	Trigger	ScenarioLocation
Objectives	strLocationName	questhelper_location_2
Objective1	Make progress	0
Objective2	Progress starting trigger	Nothing
Dialogues	Progress goal value	0
Triggers	Progress update text	
On combat start	Auto show objective progress	0
Events	Auto complete	0
Event trigger	Unique id	46
AddQuest		
CompleteObjective1		
CompleteObjective2		
On game start		

This is the second objective. It is a regular objective with quest helper location, but without progress.

Triggers	Property	Value
Quest Editor	Title	Title of Quest
Quests	Text	Longer description of quest...
Quest	Objective type	Primary
Objectives	Objectives	
Objective1	Objective	Objective1
Objective2	Rewards	
Dialogues	Reward	Unnamed
Triggers	Type	Attribute
On combat start	Resource type	Gold
Events	Attribute type	Destiny
Event trigger	Entity type	
AddQuest	Value	2
CompleteObjective1	Reputation	Life
CompleteObjective2	Give or show	Give
On game start	Unique id	41

This is the quest that has objective 1 initially. Objective 2 will be added later. The quest has a +2 Destiny reward.

Triggers	Property	Value
<ul style="list-style-type: none"> Quest Editor <ul style="list-style-type: none"> Quests <ul style="list-style-type: none"> Quest Objectives <ul style="list-style-type: none"> Objective1 Objective2 Dialogues Triggers <ul style="list-style-type: none"> On combat start Events Event trigger <ul style="list-style-type: none"> AddQuest CompleteObjective1 CompleteObjective2 On game start 	<ul style="list-style-type: none"> bInitiallyOn: true bDisabled: false pCondition: Nothing pAction: Array <ul style="list-style-type: none"> pElements: <ul style="list-style-type: none"> Element[0]: SetNumericVariable <ul style="list-style-type: none"> strVariableName: progress pVariable: 0 (VarNumeric) f: 0 Element[1]: Scenario_AddQuest <ul style="list-style-type: none"> QuestId: Quest 	

This is the first trigger. It sets up a numeric variable to 0. This variable is used as the first objective's progress value, so it will be 0/3. The second action adds the quest to the player's quest log, so the player will have this quest with 1 objective at the start of the map (because there are no conditions in the trigger).

Triggers	Property	Value
<ul style="list-style-type: none"> Quest Editor <ul style="list-style-type: none"> Quests <ul style="list-style-type: none"> Quest Objectives <ul style="list-style-type: none"> Objective1 Objective2 Dialogues Triggers <ul style="list-style-type: none"> On combat start Events Event trigger <ul style="list-style-type: none"> AddQuest CompleteObjective1 CompleteObjective2 On game start 	<ul style="list-style-type: none"> bInitiallyOn: true bDisabled: false pCondition: ScenarioC_EntityIsInLocation <ul style="list-style-type: none"> pEntity: ScenarioEntity <ul style="list-style-type: none"> eid: AdventureMap\FactionHero (id: 8203, at: 37, 28) pLocation: ScenarioLocation <ul style="list-style-type: none"> strLocationName: location1 bBoolean: true pAction: Array <ul style="list-style-type: none"> pElements: <ul style="list-style-type: none"> Element[0]: Scenario_AddObjective <ul style="list-style-type: none"> QuestId: Quest ObjectiveId: Objective2 Element[1]: SetNumericVariable <ul style="list-style-type: none"> strVariableName: progress pVariable: Compare <ul style="list-style-type: none"> pOp1: GetNumericVariable <ul style="list-style-type: none"> strVariableName: progress dwOp: add pOp2: 3 (VarNumeric) f: 3 strName: 	

The second trigger (after the player enters the first location) adds the second objective to the quest and then completes the first objective by increasing the variable's value by 3. The objective is added before completing the second objective because of the quest completion rule: when all objectives of a quest is completed, then the quest itself becomes completed too.

Increasing the value of a variable can be done in the following way: Select the *SetNumericVariable* action inside Scenario/Variable folder. Enter the name of the variable. Then set the **pVariable** parameter to "Compare" that can be found in the *_Obsolete/Condition* folder. Set the first operand (pOp1) to *GetNumericVariable*, set its name to the same name as the variable above. Set the second operand (pOp2) to *VarNumeric* (found in the Variable folder) and set its "f" parameter to 3. Set the operator (dwOp) to "add".

This second action increases the progress variable by 3, so the objective's goal value is reached. Because the automatic completion is true (1), the objective will be completed with this action.

Triggers	Property	Value
Quest Editor	bInitiallyOn	true
Quests	bDisabled	false
Quest	pCondition	ScenarioC_EntityIsInLocation
Objective1	pEntity	ScenarioEntity
Objective2	eid	AdventureMap\FactionHero (id: 8203, at: 37, 28)
Dialogues	pLocation	ScenarioLocation
Triggers	strLocationName	location2
On combat start	bBoolean	true
Events	pAction	Array
Event trigger	pElements	
AddQuest	Element[0]	Scenario_SetObjectiveState
CompleteObjective1	ObjectiveId	Objective2
CompleteObjective2	ObjectiveState	Accomplished
On game start		

This third trigger makes the second objective completed when the Hero enters the second location. The quest becomes completed automatically, because all its objectives are completed.