2.5 Conditional actions

Sometimes actions needs to be performed on complex conditions for which there is no event associated. In this cases, the developer needs to write a special condition under which, the action will be performed. And the game engine has to be instructed to check the condition at every possible moment. When the condition is met, the action is performed. This offers great flexibility to the developer. However, conditional actions are in the end actions that need to be linked to an event (this is how GameMaker works). If the condition needs to be checked every time, then the event is naturally, the "step" event. And the action is simply a piece of program that additionally checks for the condition.

Let's get to know more about conditional actions with an example: if the player enters a deadly zone and there are no extra lives, then the game should end, and inform the user with the dreaded "GameOver" message. There is no event for this complex condition. The way this happens is by executing actions only when the some condition is met. In this example, the actions would be:

1. Freeze the game
2. Show the "You entered a deadly area. Game over" image or text
3. Wait for some seconds to allow the user see and understand the message, or expect he clicks some "ok" button, accepting the fate.
4. Change to the initial menu or level

The condition is composed by comparing values to variables or to functions. Variables are names that hold values inside. For example the number of remaining lives of the player could be held by a variable called "lives". The programmer has to decide where and how does a variable change its value. The value can be used whenever it's needed.

Also, the vertical position of the player in the room is held by a variable called "y".

In the example the statement for the condition in the object handling function could be something like:

```
( lives == 0 )  and  (  y <100 )
```

This is read as "is the number of lives is 0 and simultaneously, the position of the player is below some border?" Whereas the above is just the statement of the condition, the whole instruction where it is binded to the action is performed by the "if" statement, which includes the condition and the action:

```
if  ( ( lives == 0 )  &&  (  y <100 ) ) {
     msg = show_message_async( "You entered a deadly area.
           Game over");
     room_goto(room_first);
}
```

Here, the question or statement of condition is written inside the parenthesis, and the braces "{ }" are used to group the sequence of statements that will be executed if the condition happens to be true. The above is the general way to include conditional actions into games.

Game Maker "*events and actions*" is a special case of conditional actions. While general conditions are able to verify almost any conceivable condition and could implement events, the latter is preferable as it makes the development simpler, and reviewing the development is more straightforward.