

Game Develop Tutorial

created by 4ian
translation and layout by DiSc , 10/14/11



Table of Contents

1.Preface.....	3
2.Inserting images	4
3.Scene design	6
4.Creating and positioning objects	7
5.Creating events	11
5.1.Moving the target with the mouse and rotating the spaceship	11
5.2.Moving the spaceship by pressing the up arrow key	14
5.3.Clicking and shooting	15
5.4.Creating enemies and managing collisions.....	16
6.Possible improvements.....	17

1. Preface

This tutorial shows you how to create a bare-bone game called Galaxies War with Game Develop. In this mini-game you pilot a spaceship and have to shoot down the enemies appearing at the top of the screen before they reach you.

If you have not already done so, you can read the **Getting Started** guide included in the Game Develop on-line help, and you might want to check out the examples shipped with the program as well. This way you will get a head-start in getting to grips with the tutorial.

Finally, you can do some experimenting by trying to improve the mini-game, which is far from being complete.

You can consult the Game Develop on-line help and, if you do not find the answers you are looking for, you can post a message on our forum or mail us at compilgames@gmail.com.

2. Inserting images

We are going to start by inserting some images. Start up Game Develop or create a new game. You can find your game in the [Project Manager](#) on the left side:

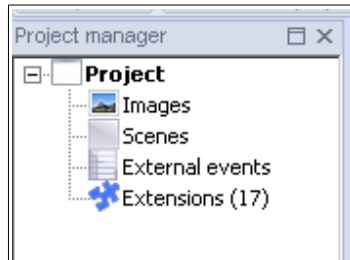


Fig. 1: A new game is automatically created at start-up time.

Double click on [Images](#) in order to open the game [Images bank](#), which will appear at the center of the center. Right-click on [All images](#) and choose [Add an image](#):

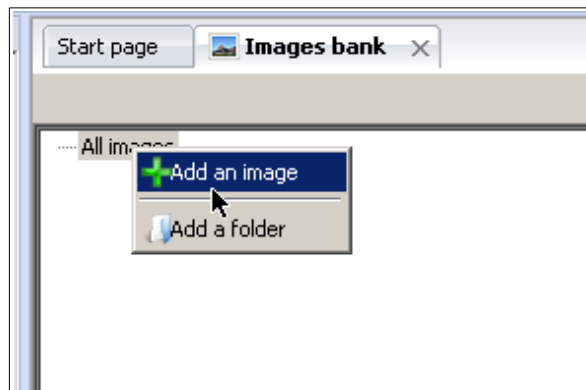


Fig. 2: Add images with the contextual menu.

You can also add images from the ribbon: click on [All images](#) in order to activate the ribbon for the [Image Editor](#), then click on the [Add an image](#) button at your screen's top-left corner:

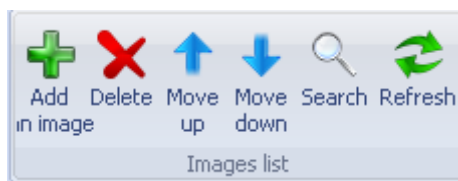


Fig. 3: You can also add images from the ribbon.

We are going to add several images at a time. Navigate to the directory containing this tutorial's images, and select [Vaisseau.png](#), [VaisseauMechant.png](#), [Viseur.png](#) and [MiniTir.png](#):

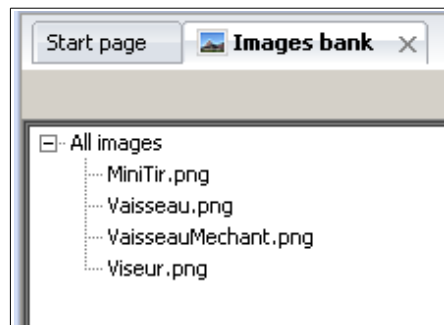


Fig. 4 The images have been inserted

NOTE: Do not be disconcerted by the file names: it is French for "vessel", "enemy vessel", "target" and "mini shot" respectively. Game Develop is a mostly French-speaking project and much of the included material is still in French.

We are going to leave the image names as they are (by default, Game Develop names the images with the corresponding file names).

3. Scene design

We are going to create a scene, i.e. a simple level.

In the **Project Manager**, right-click on **Scenes** and pick **Add a Scene**. This will add an item called **New scene** under the **Scenes** caption.

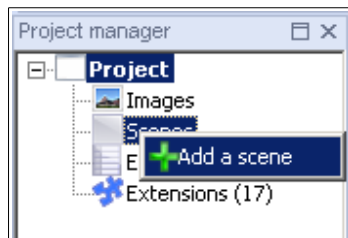


Fig. 5: Add a scene by right-clicking on Scenes.

Right click on **New scene**, click on **Change the name** and enter **Level**.

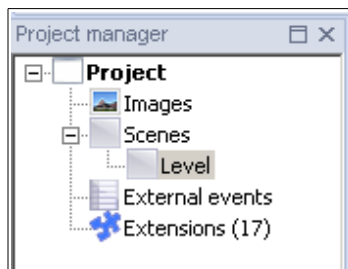


Fig. 6: A scene named Level

Let's continue by designing the scene. Right-click again on the scene named **Level** and choose **Edit the properties**. In the window, change the background color to black and type **Galaxies War** in the **Window's title** field, then click **OK** to close this window:

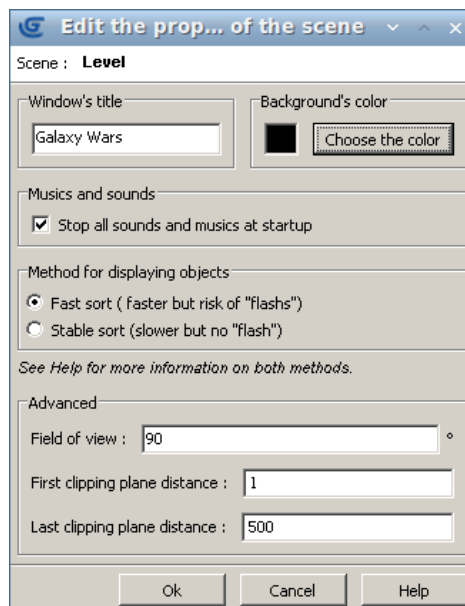


Fig. 7: The Edit the properties of the scene window

4. Creating and positioning objects

We are now going to edit our scene. Select the Level scene and double-click on it. A **Scene Editor** will open at the center of the screen, just like the Image Editor. The scene still contains no objects. We are going to create four objects: a spaceship, a target, an enemy spaceship and a shot.

In the **Object Editor**, now appearing on the right-hand pane, right-click on **All objects** and pick **Add an object**, as shown in the image below:

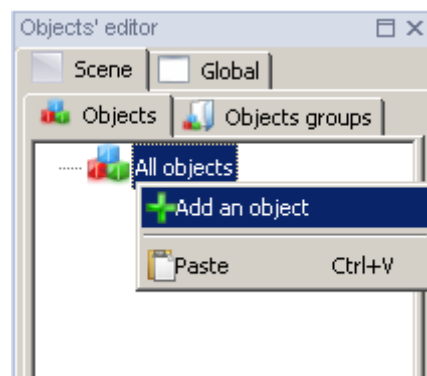


Fig. 8: Adding an object

You can also add an object by first clicking on **All objects** and then clicking on the **Add an object** icon from the ribbon as shown below:

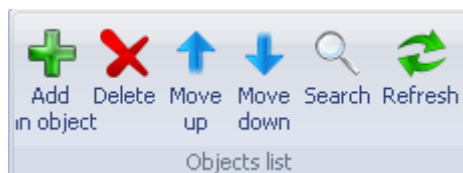


Fig. 9: You can also add an object from the ribbon

A window will appear, allowing you to choose the type of object you want to add. Every object type has different functionalities and possibilities.

By default only the **Sprite**-type is listed, and this is the type we are going to use for this tutorial. Double-click on **Sprite** and an item called **New object** will appear in the list. Select it, right-click on it and rename it **Spaceship** by typing in the name field.

Now edit the object by right-clicking on it, but this time choose **Edit the object properties**. The **Edit the Object** window will appear, as shown below. You can achieve the same result by double-clicking on the object.

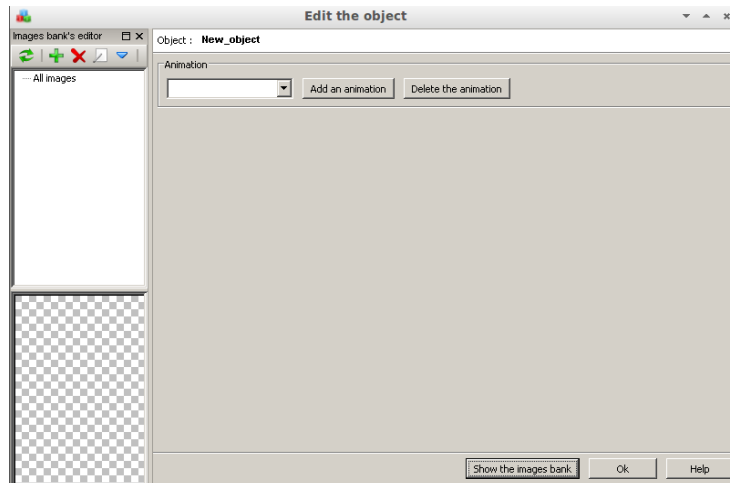


Fig. 10: The Edit the Object window

Insert an animation by clicking on [Add an animation](#), on the top-left. We are now going to add the Spaceship image. Click on [Show the images bank](#) on the window lower side to make the [Image Bank](#) appear on the left side: there you will see a list with the images we added before. Click on the drop-down menu in the Animation frame and make sure it is set to **0**, then select [Vaisseau.png](#) from the Image bank and right-click on the dark bar in the Images frame, on the right. From the menu, pick [Add selected image](#) and [...after this one](#).

You can also click on this icon  to insert the selected image.

Finally, our spaceship will need to rotate when changing direction on the screen. This feature is automatically activated by ticking the [Automatic rotation](#) box.

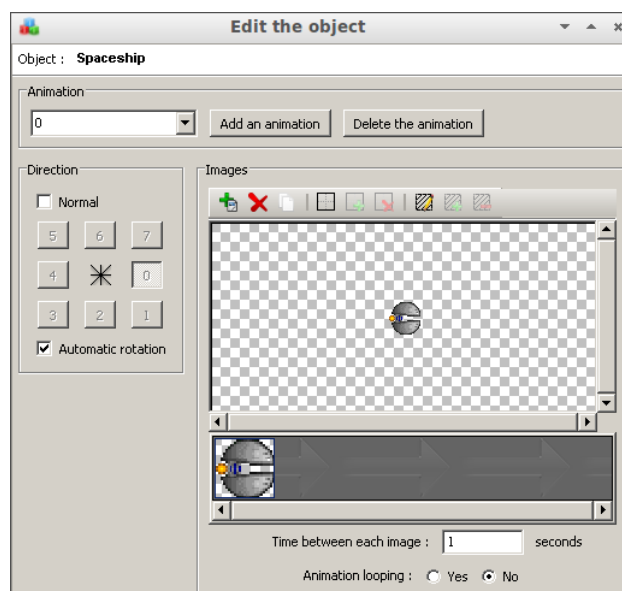


Fig. 11: Our Spaceship object is complete.

- Repeat the previous steps in order to create three more objects:
- an object called [Shot](#) with the image [MiniTir.png](#) in animation 0, direction 0;
 - an object called [Enemy](#) with the image [VaisseauMechant.png](#) in animation 0 and with [Automatic rotation](#);
 - an object called [Target](#) with the image [Viseur.png](#) in animation 0, direction 0.

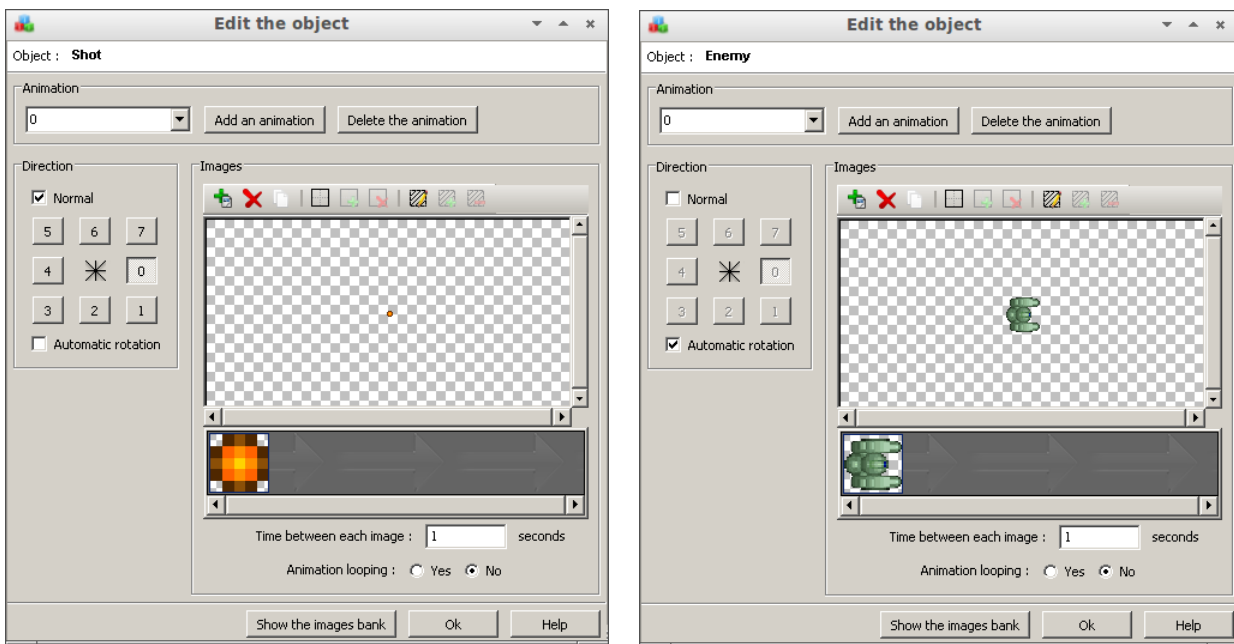


Fig. 12: The Shot and Enemy objects

Our **Object editor** should now look as follows:

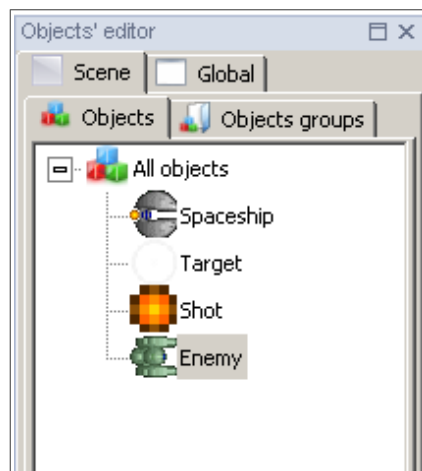


Fig. 13: The four objects in the Object editor.

Let us now position our objects on the scene: a vessel and a target. Make sure that you are in **Edit mode**. Double-click on the Level scene from the **Project Manager** on the left side: the scene black background will appear at the center of the screen. Click on the background to activate the related ribbon, then click on **Edition** in the ribbon top-left corner. You can start the game by clicking on the **Play** icon.



Fig. 14: Switch between Edit and Preview mode by clicking on these two icons.

In order to position your objects, you will need to click on the ribbon icon named **Choose an object**, shown below:



Fig. 15: You can position an object by clicking on Choose an object

Select Spaceship from the pop-up window and click on **Choose**. The pop-up window will close; now place the Spaceship object by double-clicking anywhere on the background. Do the same for the target.

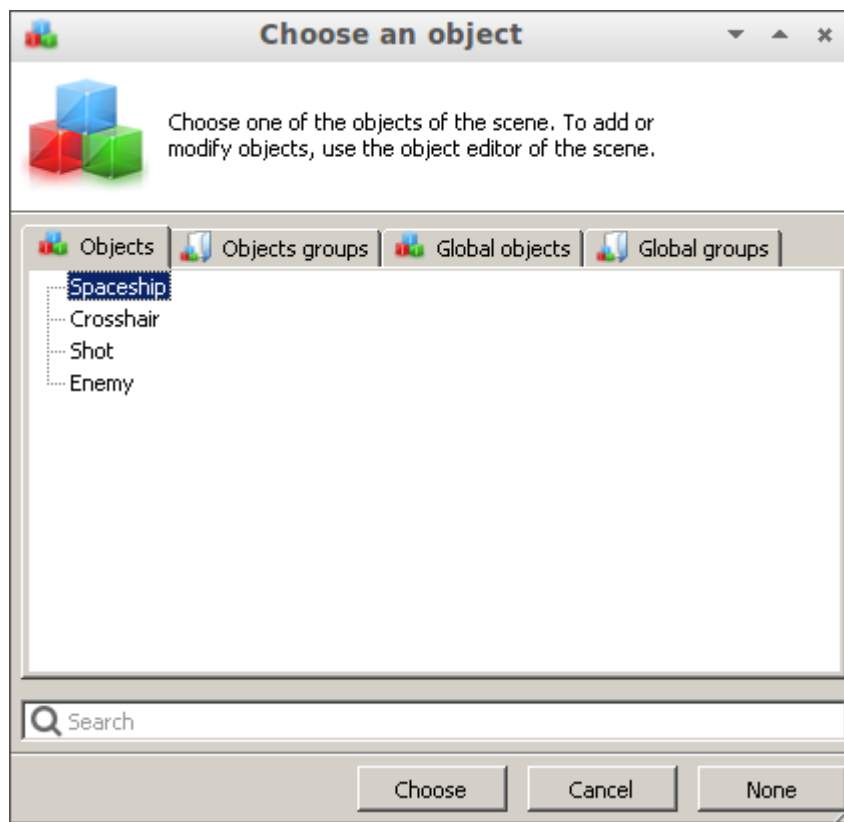


Fig. 16: Select the object you want to insert.

Alternatively, you can place objects in the scene by dragging and dropping them from the **Object editor** on the right-hand side.



Fig. 17: Our two objects have been added to the scene.

We are now finished creating and placing our objects. Let us now see how to animate a scene by using events.

5. Creating events

Creating events is probably the most complicated step, but also the most powerful it allows you to define how a scene is going to behave.

We are going to fulfill several objectives here:

- Moving the target with the mouse;
- Rotating the spaceship towards the target;
- Moving the spaceship with the up arrow key;
- Making the spaceship fire a shot when pressing the mouse right button;
- Making enemies appear and managing collisions.

Let me first point out three concepts we will be using in the following paragraphs: [events](#), [actions](#) and [conditions](#).

[Events](#) are changes in the way our game behaves, for example the spaceship on the background may start moving around the screen. An event may consist of several [actions](#): for example, in order to move the spaceship, we need to a) set its direction and b) change its position. Events are triggered by [conditions](#): the spaceship will only move when a key is pressed.

The meaning of these terms will become clearer as we develop the Galaxy Wars game.

5.1. Moving the target with the mouse and rotating the spaceship

First of all, we need to make sure that the spaceship rotates towards the target, and that the target is on the same place as the mouse.

Click on the [Events](#) button underneath the scene to open the event editor. In the ribbon, click on the button labeled [Add event](#). The following will appear:



Fig. 18: A newly inserted event.

We are then going to add two actions that will cause the target to follow the mouse. We are not going to put any condition to this event, since these actions will have to be repeated all the time.

Double-click on the field showing [No actions](#) in order to add an action in the pop-up window:

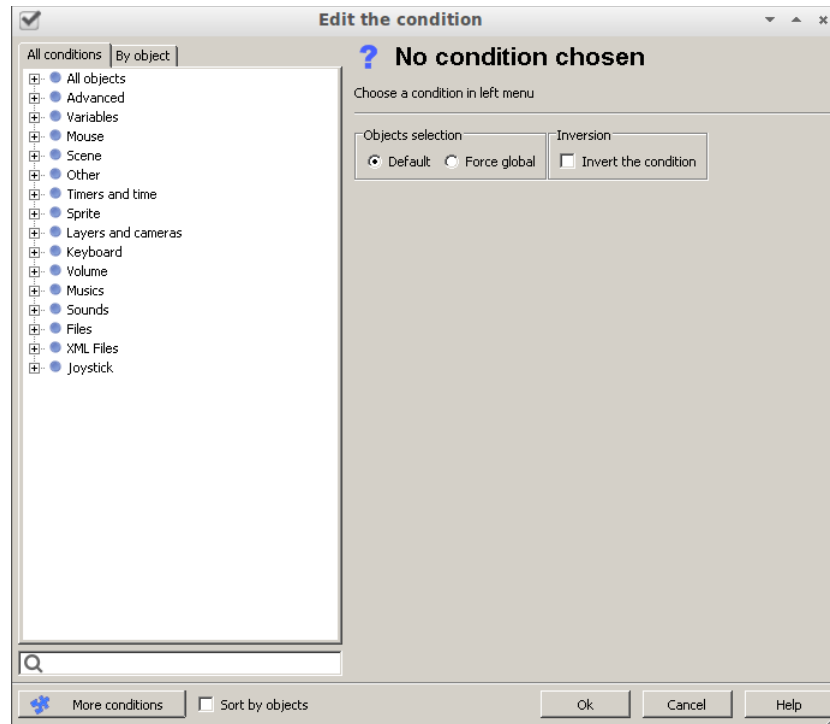


Fig. 19: The Edit the action pop-up.

We want the target to always follow the mouse pointer, and this means that the X and Y location of the target will have to be the same as the mouse.

Let us start with the X: choose **All objects** from the list on the left, then **Position** and **X position of an object**. Fill out the fields as in the picture below:

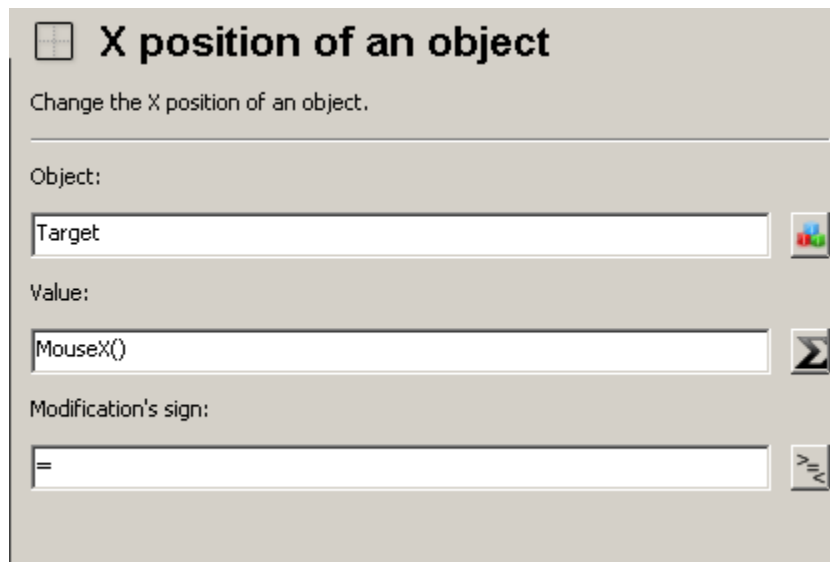



Fig. 20: The target x-position must coincide with the mouse x-position.

- The first parameter is the object name: in our case, this is "**Target**". You may also click on the button after the text field and select the target object from the list;

- The second parameter is the position. We want it be the same as the mouse x-position. Open the expression editor by clicking on the Expression button : a window will open with the title [Edit the expression](#).

Now expand the list labeled [All special values](#), click on [Mouse](#) and pick [X position of the mouse](#). Click on the [Add](#) button and then close the window by clicking on the OK button. The expression editor will have filled in the second value with [MouseX\(\)](#), which means: insert here the x-position of the mouse;

- Finally, the third parameter describes what we want to happen to the second parameter, i.e. how the object x-position must relate to the mouse x-position. We want the target position to be the same as the mouse, so we are going to fill in the equal sign [=](#).

Close the [Edit the action](#) window. In the event list, you can see an action that links the x-position of target to the x-position of the mouse:



Fig. 21: The event first action.

In order to add a second action, hover with the mouse over the first action and click on [Add an action](#) from the menu as shown below:



Fig. 22: Add a new action.

In the [Edit the action](#) window click on [Position Y](#) from the [All objects/Position](#) sublist and fill out the fields like you did for the x-position, but remember to substitute [MouseX\(\)](#) with [MouseY\(\)](#). Close the window to see the new action appear next to the first one:



Fig. 23: The two actions in our event.

Please note that you can also create this second action by clicking on the first one, copying and pasting it and editing the action afterwards. In order to copy and paste an action, select and right-click it or click on the [Copy](#) button in the ribbon. Also note that you can quickly edit an action by clicking on its parameters, without having to open the [Edit the action](#) window.

The last action will make the spaceship turn towards the mouse pointer. Insert an action by right-clicking on one of the two actions and select [Add an action](#). From the list, click on [Sprite](#), [Direction](#) and select [Rotate an object towards another](#). In the first field fill in [Spaceship](#), and in the second one fill in [Target](#).



Fig. 24: The three actions allow to position the target and rotate the spaceship.

The first event is complete. If you want to try out the scene exit the [Event editor](#), then click on the [Preview](#) button and then on the [Play](#) button in the ribbon. You can see how the spaceship turns towards the target cursor. If it does not, then go through the event actions again and make sure that the Spaceship object is in [Automatic rotation](#) mode.



Fig. 25: To preview a game, click on Edition and then on Play in the Scene ribbon.

5.2. Moving the spaceship by pressing the up arrow key

We now want to create an event that will make the spaceship move forward when we press the up arrow button.

Insert a new event by hovering on the event list and clicking on Add an event.

Let us start with the conditions. Insert the first condition as an action by double-clicking on [No conditions](#). Choose then the item [Keyboard](#) from the list, and then click on [Key pressed](#). Fill out the field by typing **Up** as key.

Let us now edit the actions.

Insert an action and edit it. Pick the item [All objects](#), [Displacement](#) and then [Add a force \(Angle\)](#) from the list.

In the first field, enter the name **Spaceship**, because that is the object we want to move.

In the second field, enter angle you want the spaceship to move at. In our case, the desired angle is the direction the spaceship is turned towards, so we need to retrieve this direction. Click on the [Expression](#) button, then expand the object properties list. Choose [Sprite](#) object and then [Direction](#), and then click on the [Add](#) button. In the pop-up, pick the Spaceship object, then close the [Edit expression](#) window.

In the third field enter for example **150**: this means that the spaceship will move at a speed of 150 pixel per second.

Finally, enter **0** in the last field. This means that the force will be multiplied by 0 when events are evaluated. In other words, the force will cease if the player does not press the up arrow button, as one would normally expect.




Fig. 26: The event managing the spaceship movement.

You can now test the scene to make sure the spaceship moves when the up arrow key is pressed.

5.3. Clicking and shooting

Let us now insert an event to allow us to shoot. The spaceship will need to shoot when we press on the left mouse button, and this every 0.25 seconds.

Create a condition and choose **Mouse** and **Mouse button** from the **All conditions** list. Type **Left** in the field labeled **Button to test:**, or click on the mouse icon  and then pick **Left button**.

Finally, in order to avoid to shoot too frequently, let us insert a new condition. Choose **Timers and time** and then **Value of a timer**. Fill out the fields on the left-side panel with **Shot** as timer name (do not forget the quotation marks) and **0.25** seconds (use the point as separator, not the comma).

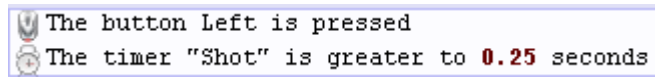



Fig. 27: The conditions of the shooting event.

Let us now look at the actions. Insert a new action in our event. From the list, pick **All objects**, **Objects** and **Create an object**.

We want to create the spaceship shot: enter **Shot** under **Object to create**, then fill out the remaining fields with the x and y-positions of the Spaceship object. To do this, open the **Expression** editor (click on the Expression button ) , then click on **All objects**, **Position** and **X position**. When asked, double-click the **Spaceship** object in the pop-up window. Do the same for the third field, but choosing **Y position** instead. The final result should look like the following:

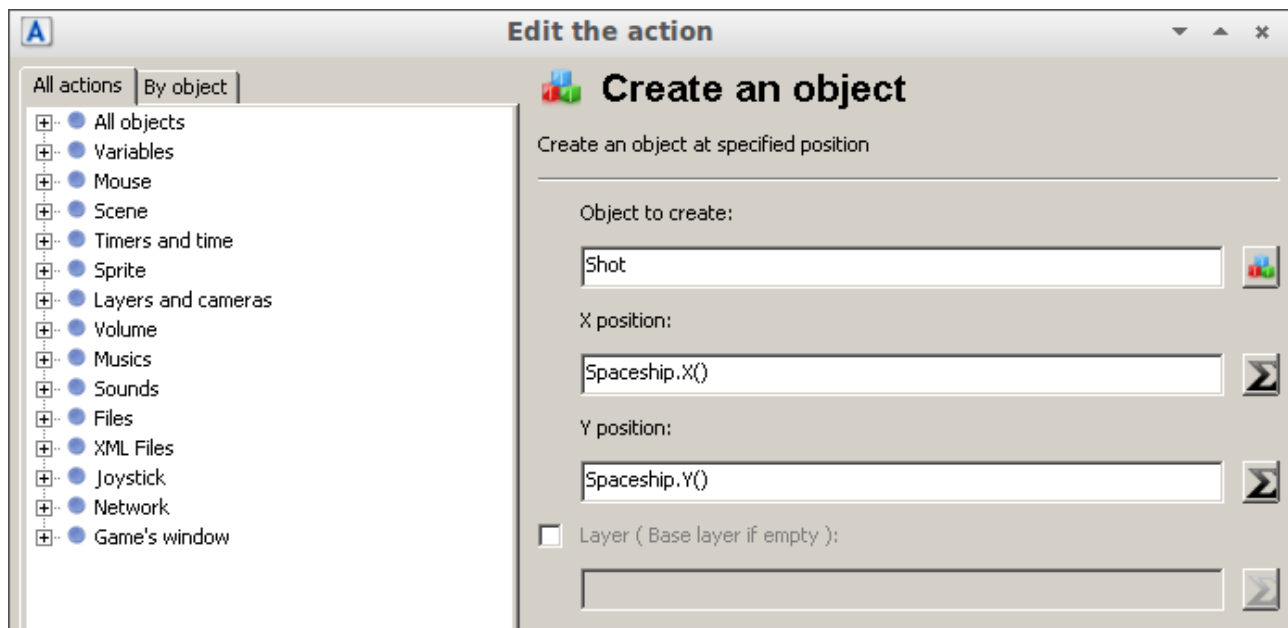


Fig. 28: The action above creates a shot in the direction of the spaceship.

We finally need to set the force acting on the Shoot object. Add a new action and click on **All objects**, **Displacement**, **Add a force (angle)**. Fill out the fields in the right-hand pane so that the

Shot object will move along the direction of the Spaceship object by **300 pixels**, and set **Dissipation** to **1**.

If you remember from the previous instructions, a dissipation value of 0 caused the Spaceship object to stop immediately when the up arrow key was not pressed anymore. A value of 1 will instead cause the Shot object to never stop, and all intermediate values will cause objects to progressively come to a halt.

Add a last action by clicking on **Timers and Time** and **Reset a timer**, then type **"Shot"** for the timer name without forgetting the quotation marks.

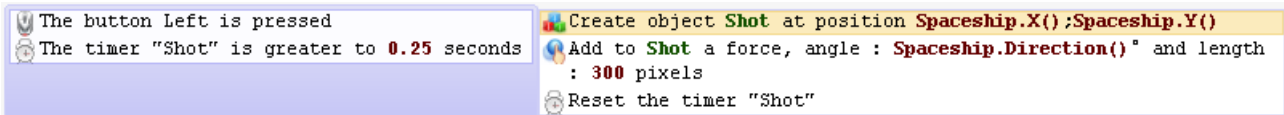


Fig. 29: The actions of the shooting event.

5.4. Creating enemies and managing collisions

We are now going to create an event similar to the previous one. This one will create Enemy objects on the top of the scene and move them towards our spaceship.

Here is a screenshot of what the event will need to look like if it is to work as desired:

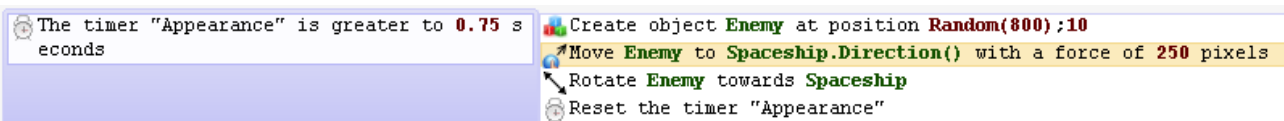


Fig. 30: The event creating the enemies and moving them towards the spaceship.

If you do not know how to do some of the above, scroll back a few pages and read again how to create events.

Tip: do not forget the quotation marks around **Appearance**!

Finally, two simple events are necessary to destroy the Enemy and Spaceship objects when they get hit:

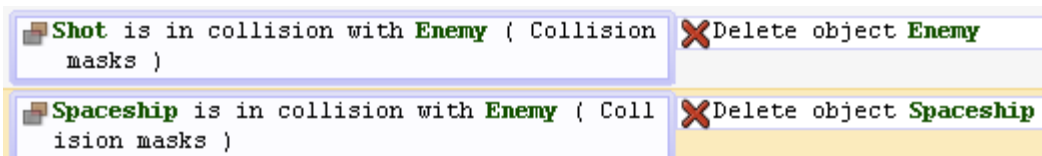


Fig. 31: The events above destroy the Enemy and Spaceship objects.

6. Possible improvements

The game is clearly far from being complete. For example, the spaceship and the enemies disappear in a very simple way to say the least. You could improve that by making an explosion object appear at a vessel's coordinates, and then making the object disappear when the animation has finished.

Next to that, you can create a Game Over scene, or show a text message if the player has died. And you can also create a Menu scene to allow the player to choose a level, quit the game, etc...

New conditions and actions will then be necessary. You should not hesitate to try them out to discover the possibilities of the program, and just check out the help if you need more information.

Happy creating with Game Develop!