

GameMaker Guide

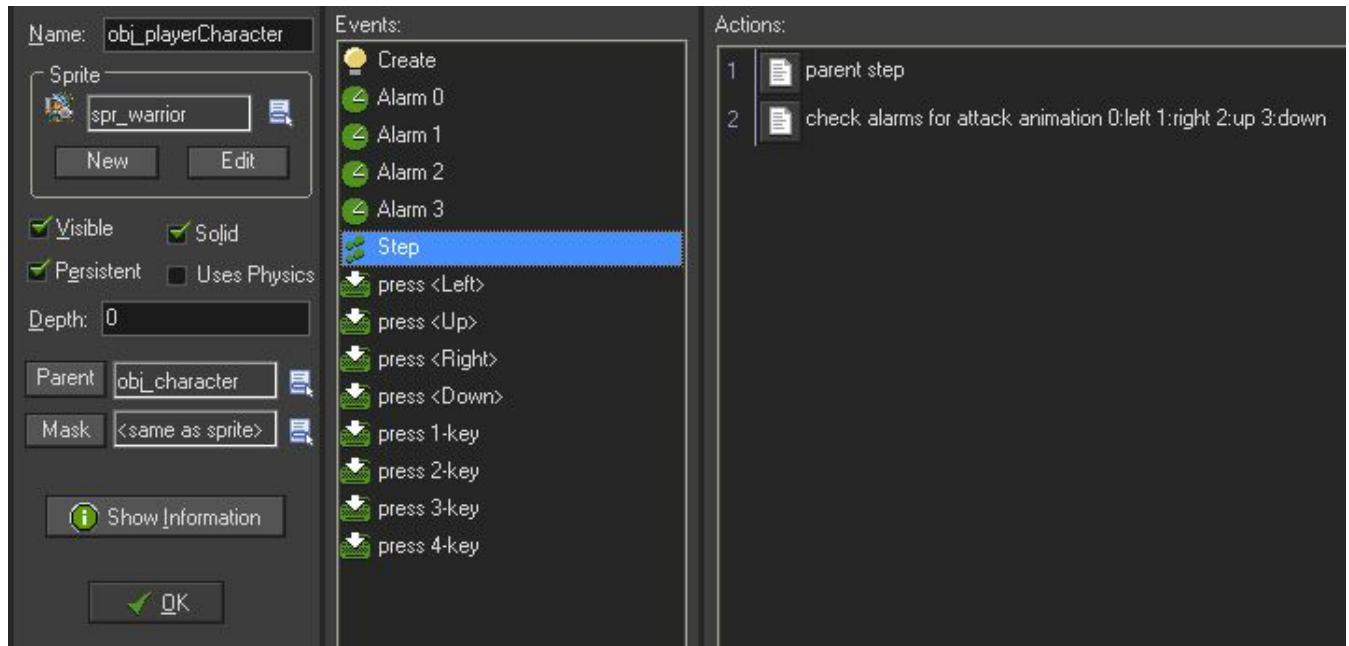
How to install GameMaker: Studios and open project:

1. Visit - <http://www.yoyogames.com/get> and click select the free GameMaker: Studios download. You will need to make an account on the website before you are able to download GameMaker: Studios.
2. Download the project from our github repository and extract it to a folder.
3. Once extracted, go to the project file and double click the “Sunsinger.project.gmx” file located in “Alpha-master\Sunsinger.gmx\” and Gamemaker: Studios should open, allowing you to view and run our project.

Gamemaker Asset Information:

1. Sprites - Representation of objects within our game. There is a sprite for each of our objects and multiple sprites for some objects. Players and enemies have a sprite for each of their animations, such as a running sprite or attacking sprite.
2. Scripts - Our scripts are written in GameMaker Language (GML) and can be called whenever needed to avoid replicating large amount of code multiple times. Our scripts perform actions such as taking keyboard/controller input, handling collision, performing an attack, etc.
3. Sounds - Free-to-use sound assets gathered from Freesound.org.
4. Objects - Objects are a special resource that we use to control aspects of a game and to do specific things. Most of the time they have a sprite associated with them so that you see them in the game rooms, but sometimes they are used as a "behind the scenes" controller to do things related to the user or for timing, etc. They can be given behaviors and they can react to certain events as well as to each other, and most of the things you see in a game are based on objects and their interactions. (Description from GameMaker's site.)

If you double-click on an object you will see more information about it, here's an example from online:



From this information you can see if the object has a parent, a list of its events, and what code file(s) will run when the event occurs. Some of the most common events you will see in our project are as follows:

Create - Create events execute when the object is created, used mainly to instantiate variables like movement speed, gravity, health points etc.

Step - Step events are executed every frame that an object exists (our game runs 60 frames-per-second), step events are used for scripts that should be constantly running, like getting user input or enemy AI.

Collision - Collision events occur when two objects collide in the game, such as when a fireball collides with an enemy or the player collides with an exit door.

Alarm - Alarm events occur when an a previously set timer has gone off. Alarm statements set in code such as “Alarm[0] = 30;” mean that the Alarm 0 event will occur after 30 frames. This is extremely useful for timing things like enemy attacks.

5. **Rooms** - Rooms are the actual level that players see when they play the game. Rooms consist of many objects like floors, enemies, traps, etc. Our rooms have a speed of 60 frames per second which means that all step event are performed 60 times a second.