

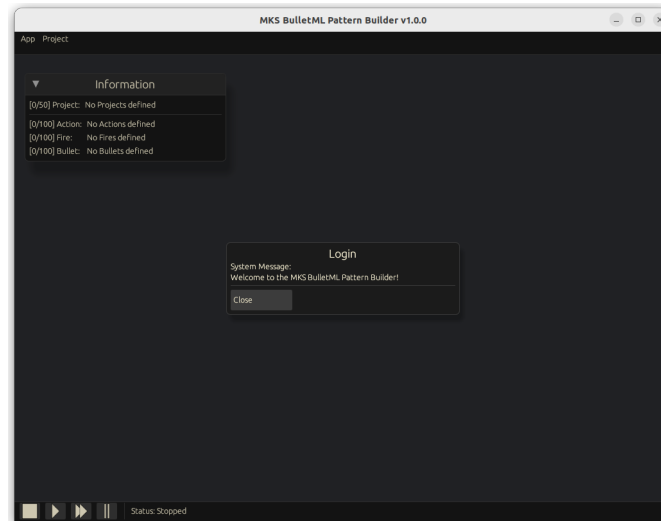
**MKS**

**BulletML Pattern Builder**

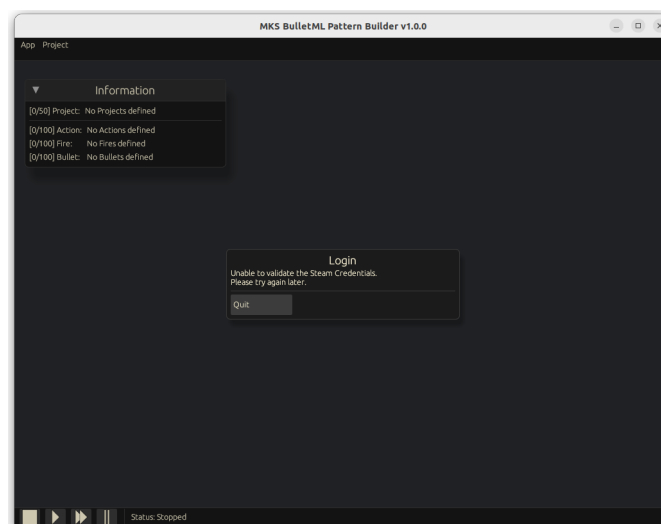
**Basic Tutorial**

# Getting Started

As an introduction, the goal here is to show the quick workflow setting up bullet patterns and previewing the results playing in the Playfield of the editor. On starting the application, the main user interface shows the application after a successful login and providing any system information that may be important for working with the tool. In the case shown below, there is no specific message available and the login has been successful.

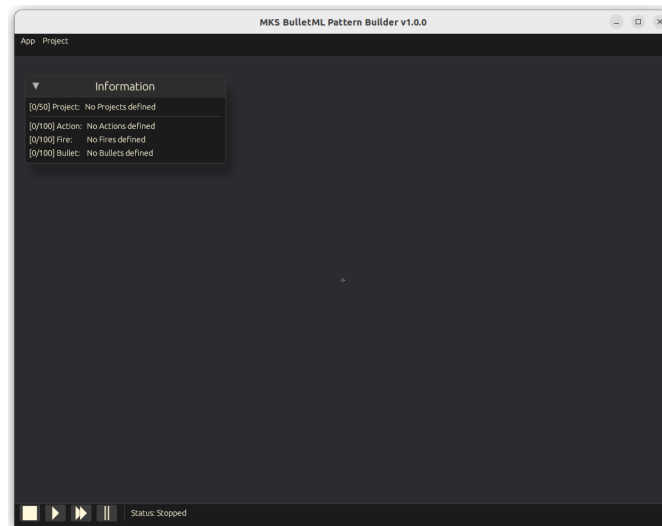


The application is launched through the Steam Launcher and requires a valid purchase on the user's Steam account as well as the execution of the Steam Launcher at all times for validation of the access rights. If the Steam Launcher is not able to verify the correct credentials and purchase or the application account is not in good standing, the Pattern Builder tool will indicate the following on start-up:



Please contact our [support](#) per email or come into our [Discord Server](#) if you encounter any issues here that we can help with.

After confirmation of the start-up message dialog, the following screen is presented. As we haven't created any projects yet, the menu shows only relevant next steps.

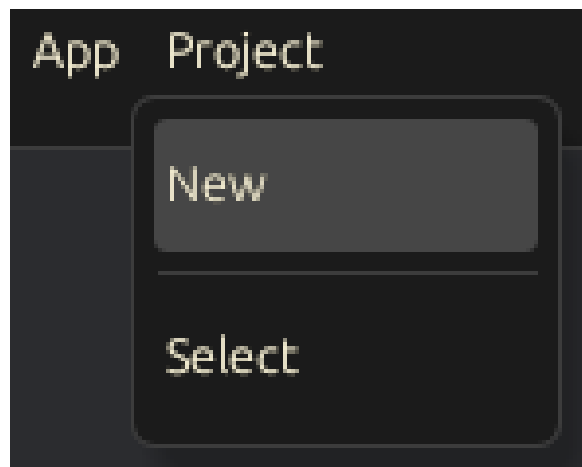


At this stage, we can go through the introductions as described below. Note that the Information Window can be moved per mouse drag in the window title or toggled visible using the F2 key or the sub-menu option “Show/Hide UI Overlays” under the App menu.

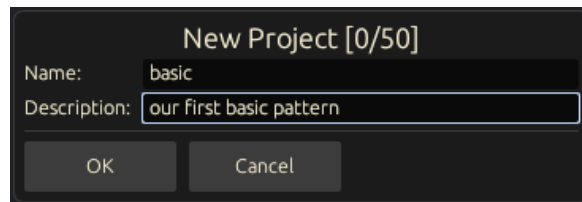
## Basic Pattern

For our basic pattern, let’s start by creating a project. To do this, select the Project menu and click on the “New” menu item. This brings up a dialog where the project name and description can be edited.

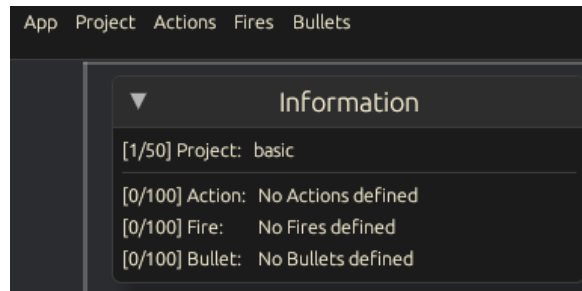
Entering a good name and description is strongly urged as the list of projects can get quite large and the data entered here greatly helps differentiate between projects later on.



For this introduction, enter “basic” as project name and “our first basic pattern” as the description.

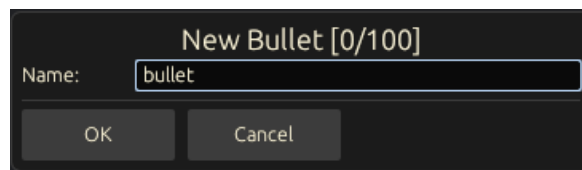


Press “OK” to close the dialog. The menu bar at the top of the application has now expanded fully and will allow us to create the asset content required to see some bullets flying.



As Actions depend on Fires and Fires depend on Bullets, we will create them in reverse order here.

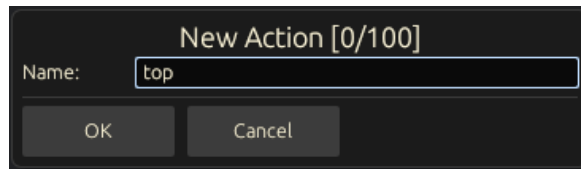
First, select the “Bullets/New” menu option and we will define a Bullet named “bullet” here as shown below.



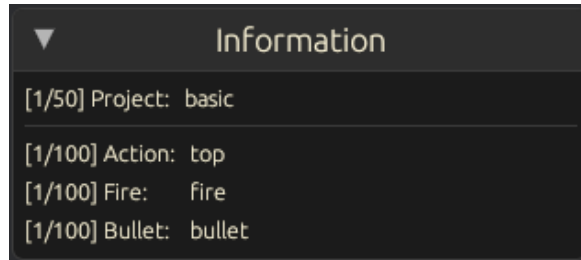
Second, select the Menu “Fires/New” and enter the following data to create a Fire named “fire”. The Fire is used to emit a Bullet when triggered.



Press “OK” here to close the dialog and select the “Actions/New” menu option. As per naming convention, we will name the first Action, which is also where the playback always starts at, “top”.

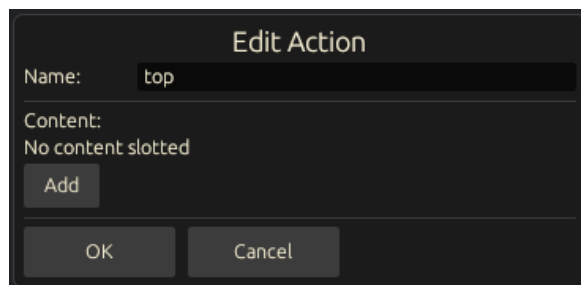


The Information Window now reflects what we have created up until now. We do need to customize the assets though to ensure playback is as we intended.



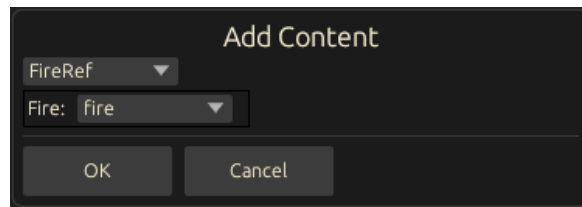
We have used 1 of 50 Projects on our account as stored in the Cloud. In this project, we have created our main Action “top” and a Fire named “fire” to emit a Bullet named “bullet”. We have used up 1 slot of each asset type but could create a further 99 in this project’s budget. These assets shown are also the currently activated elements for any editing we will do. As you switch between multiple assets, the Information window always displays the currently active set of assets.

We can now customize the Action to trigger the Fire in an interval. Select the menu “Actions/Edit”. As the currently active Action asset is “top”, we can now edit it.

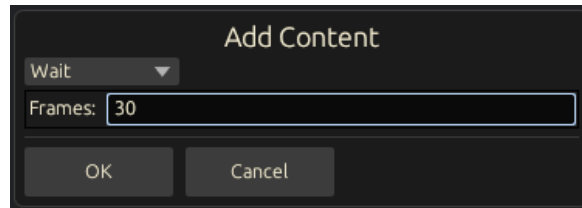


We want to create a Fire emission and then wait 30 frames, which will amount to half a second, until this Action repeats. We will have no further logic in this example.

In the Edit Action dialog, click on “Add” to create a new FireRef content entry from the drop-down widget. As “fire” is the only available Fire asset, it has been preselected.



Press “OK” here and see that the “Edit Action” dialog now contains the entry we just created. Click the “Add” button again and create a Wait content entry with a 30 frame wait period.



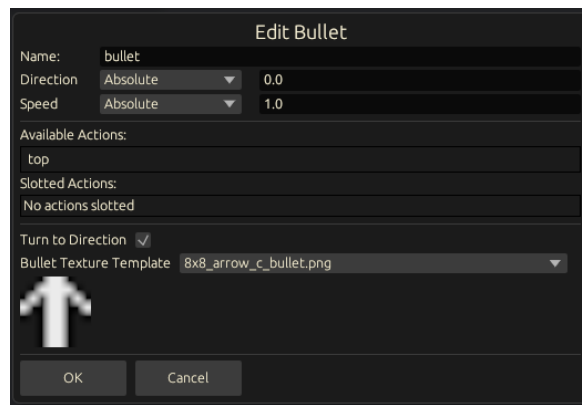
The “Edit Action” dialog should now read as displayed below. We can then move on to the customization of the Fire “fire”.



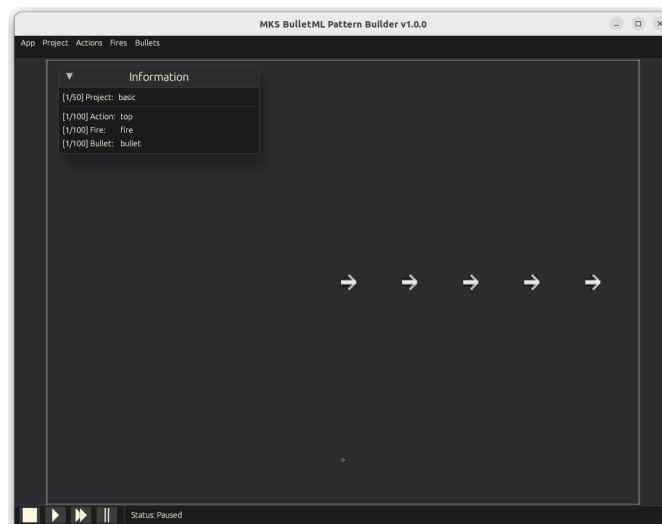
Press OK here and select “Fires/Edit”. As the currently active Fire is “fire”, we now see the dialog allowing us to customize that specific Fire asset. In this case, we will leave everything as per default and the previously created Bullet “bullet” has already been preset. The direction and speed control the emission velocity of bullets and the parameters customize the behavior of the bullets as they are emitted. Though no parameters are used in this basic example, they can be used to enable parametric inputs to the Bullet velocity at runtime.



To finish this setup, let's edit the Bullet "bullet" by selecting the menu "Bullets/Edit".



Set all options as they are shown above, and make sure to select the Bullet Texture Template option "8x8\_arrow\_c\_bullet.png". Press OK to close this dialog and let's see it all in action.



Press the "Play" button in the status bar of the application. If all went well, you should see a steady stream of bullets being emitted horizontally and leaving the Playfield on the right

side, as the default setup is configured for a horizontal shooter type game which would scroll towards the right side.

This project has been saved in the Cloud and can be accessed at anytime later and from any application running this Steam account on the list of supported platforms as described in the introduction.

This concludes the basic introduction to working with this tool. A more involved walk-through follows below.

15 <sup>th</sup> August, 2025	v1.0 - Initial version.