

Spell Creator

Concept

First Concept

A visual spell creator made up of events and actions. This tool is meant for more complex spells like 'A burst of fireballs that explode into small bouncing orbs dealing damage over time'

Targeted at game designers, the tool should include many pre-made nodes and ways to add custom particles, models, etc. With programming knowledge, the tool should be easily expandable.

Final Concept

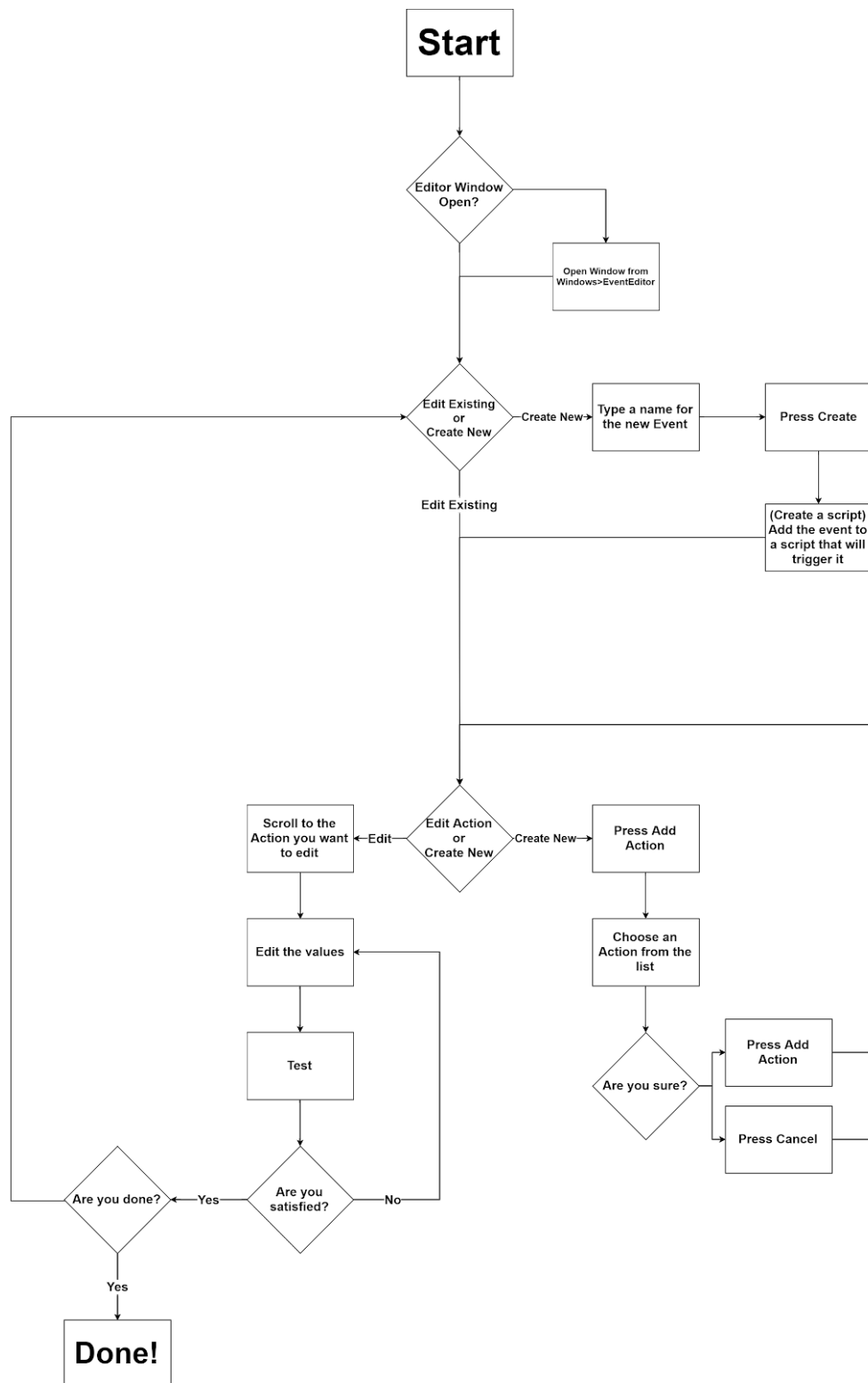
The concept has changed during the project into a more general visual scripting tool. This still uses the idea of Events that can be triggered via simple code. The event contains Actions that are preprogrammed functions the user can add and modify.

I have mostly worked on the tool itself rather than the actions for it and in my opinion, the tool seems promising but is still missing features that would make it good:

- Loops in actions
- Actions returning variables available for other actions
- Events returning variables available for actions
- Possibility to use any variable of any object as a parameter (using reflection)
- Modifiers (either as decorator pattern or just as optional variables that need to be enabled)
- A big library of useful actions
- Separate save locations for events and actions

User Flow

My tool could have infinite used so this user flow will depict the general steps needed to create or edit something.



My code in short

Event.cs

This ScriptableObject contains a list of all it's Actions and has functions to add and remove them. This is also where all actions are executed in a Coroutine triggered by other scripts.

Action.cs

This ScriptableObject is a base class for all Actions to inherit

EventSaver.cs

This static class is used for saving and loading the events. This script also contains functions to write the Events to an XML file. I don't use them anymore because I now save everything as ScriptableObjects.

EventEditor.cs

This Editor script handles the drawing and functionality of the editor window.

ActionTracker.cs

This static class's only purpose is to get all Actions in the Assembly. It is only used by the editor.

Modifier

The Modifier script is not used. This is because I prioritized other features.

Repertoire Research

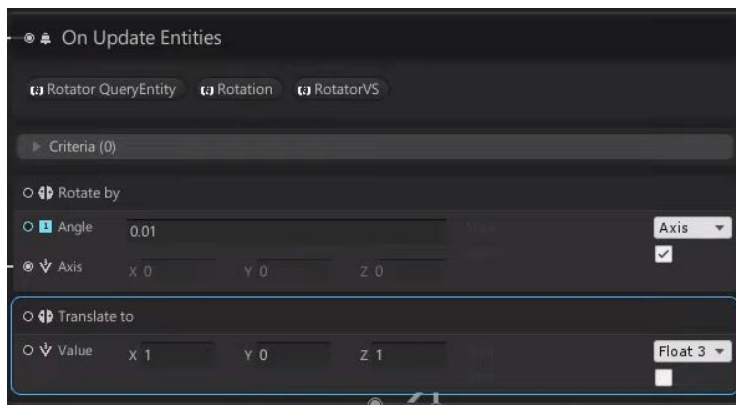
There are surprisingly few similar tools I could find. Most 'tools' I found were very limited and only used some inspector fields.

1. [weapon system](#)
Very Limited, Paid
2. [easy weapons](#)
Somewhat flexible for standard guns
3. <https://github.com/bevren/Unity-Spell-Maker>
Very Limited, looks to be only for 3rd person RPG

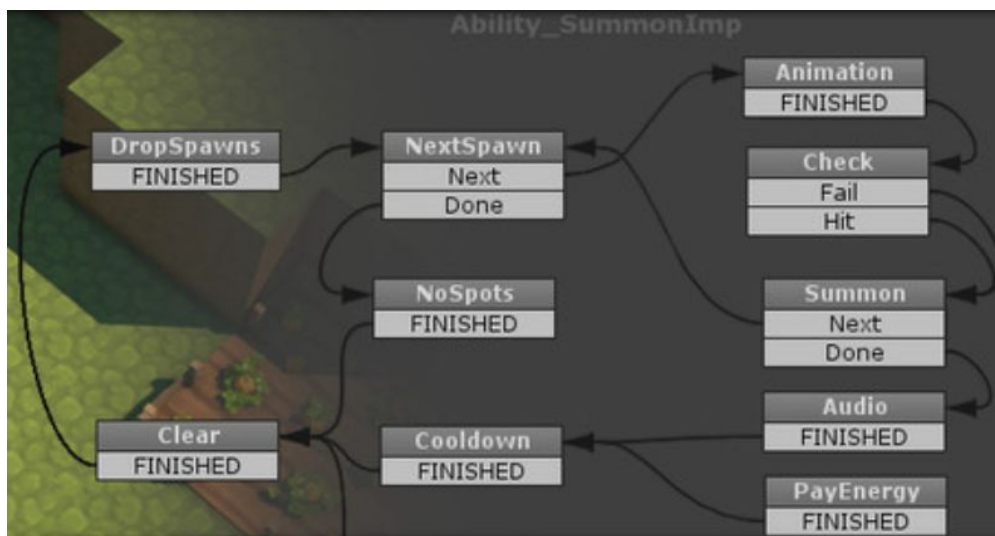
Researched later:

Most Visual scripting tools use a full node system, I decided to keep it simple.

4. Unity Visual Scripting for DOTS (in development)
I used a similar strategy they use for the OnUpdateEntities of a list of actions.



5. PlayMaker (Unity Store)
It looks like a powerful tool, but I think the node editor itself misses some information that makes it less clear what exactly would happen, this is only visible in the inspector. (What is checked at a Check node)



6. Bolt (Unity Store)

It seems to be an almost perfect visual scripting tool, but I expect that some quite simple actions will have to be built by the user itself (example: adding a force relative to a transform's direction)

7. uScript (Unity Store)

It looks like it uses a lot of logic nodes, which means that although someone wouldn't need much programming knowledge, it still requires a lot of logic and thinking.

Troubleshooting for users

1. Check if there is any message in the console that would suggest the problem (E.g. "GameObject does not have a rigidbody")
2. If you can't add an action:
Check in the Data/Tool folder in the installation location of that action still has a script reference. If so delete the object and add it again with
RightClick>Create>Actions>[ActionName]
3. If an Event has lost actions you could try to find the action assets in Data/Saved and add them to the list of the Event asset
4. Cry for help

Testing

I created a very basic example scene and asked a game designer to create something similar.

Result

Observations:

It was easy to use the editor itself but getting to know how each action works was more difficult. The tool is very limited at the moment. Actions could have tooltips / info to explain their behaviour. Only small thing that really should be fixed as soon as possible is that when an new event is created it isn't selected and thus seems not to work

User FeedBack:

It was very easy to use the editor to create a simple TNT that would explode after some time after landing, coding this would have taken more time. (Would have to search up required functions).

Strong points would be: Being able to make something quickly without having to worry about code. Great for prototypes.

Risks: The tool will be very dependant on the amount of actions that are included, for more complicated use cases it might still be better to write your own actions.

Could be improved: Tooltip on the more vague actions, did not know what "relative" meant but after a short explanation it made sense.

Reflection

As mentioned before, I am quite pleased with the result but it needs more work to be really awesome. I unfortunately did not have the time to add as much as I wanted, everything also took more time that i expected. My code is in my opinion quite good and clean but not finished. I also switched from XML to ScriptableObjects because it made it easier to draw an editor for all action variables.