

RPG Template (lite-c)

Made by Dennis van den Broek (aka Helghast)

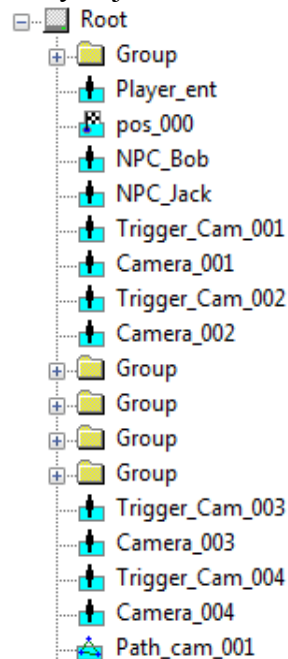
This document is made to help understand how to setup the actions from the actions.c file. It is in no way a full documentation on the whole code. I personally feel however the code is well documented, and experienced users shouldn't have a problem reading them. If however you are still trying to grasp the basics of coding, then I'd suggest to read through the AUM's first and try to get the hang of that, this Template is build modular, and contains lots of memory allocating code, which can be quite difficult to understand.

I hope I did not scare anyone away by this Disclaimer, I hope you enjoy the work I put in it so far!

General Setup

Before we dive into the scripts part, it is important that you name your entities inside WED properly. Not only is this a good habit if you are working within a team, so they know which things do what, it is also easier for yourself, especially if you are looking back at your level after some months, and need to learn things all over again ;)

Here is an example of how I named my objects in the test level of the Template:



You can open testlevel.wmp from the Levels folder, and check the rest out for yourself.

Keep in mind, some functions of the template rely on writing the right name of an object into the “strings” field, therefor I can't urge you enough to write down sensible accurate names for entites.

Actions Setup

I went through and setup the actions, when applied in WED, the skill names show up (and a default value is filled in), with what each of them do. However, Conitec didn't implement this feature for Strings, so to clarify all that, this document is written. Most of it though should be straight forward out of the box, and if not, you can always refer back here.

Actions.c Humanoids

action name:
 ACT_player

what it does:
 Makes the current entity the playable character (only 1 player at the same time is supported)
skills used:
 Skill 2; Set's up the gravity, this makes the player fall faster/slower (default 15)

action name:
 ACT_NPC

what it does:
 Makes the current entity the a static NPC (non playable character)
skills used:
 Skill 2; Set's up the gravity, this makes the player fall faster/slower (default 15)
 Skill 3; Set's how "wide" the NPC is (diameter), this is used for collision detection. (default 40)
 String 1; Path to the file containing the dialogue.
 String 2; Name the Character Dialogue should start

action name:
 ACT_NPC_Walk

what it does:
 Makes the current entity the a walking NPC (non playable character)
skills used:
 Skill 1; The speed at which the NPC walks (default 10)
 Skill 2; Set's up the gravity, this makes the player fall faster/slower (default 15)
 Skill 3; Set's how "wide" the NPC is (diameter), this is used for collision detection. (default 40)
 String 1; Path to the file containing the dialogue.
 String 2; Name the Character Dialogue should start

action name:
 ACT_player

what it does:
 Makes the current entity the playable character (only 1 player at the same time is supported)
skills used:
 Skill 2; Set's up the gravity, this makes the player fall faster/slower (default 15)

Camera's

Every trigger that has a camera action applied, needs to have another entity in the level that serves as a camera. It also uses the WED entity name inside the trigger for setup. Again, look at the testlevel to see how it is intended to work.

Also, make sure never to set Flag8 on any of the triggers, this is used on internal code.

action name:

ACT_Trig_fade

what it does:

Fades the camera to the new position by creating a new view, and fading that one over the other. (This feature works on Commercial and up only!)

skills used:

String 1; Entity name the camera takes the position and rotation over from.

action name:

ACT_Trig_instant

what it does:

Moves the camera to the new position/rotation instantly, used for hard camera cuts.

skills used:

String 1; Entity name the camera takes the position and rotation over from.

action name:

ACT_Trig_smooth

what it does:

Moves the camera to the new position/rotation smoothly, lerps it to the new position, from current position.

skills used:

String 1; Entity name the camera takes the position and rotation over from.
Skill 1; the speed the camera uses to lerp to new position (default 1.1)

action name:

ACT_Trig_path

what it does:

Moves the camera to the new position over a predefined path. Make sure the path is attached to the camera used, not to the trigger. Besides that, this action requires you setup speeds on the camera itself.

skills used on Trigger:

String 1; Entity name the camera takes the position and rotation over from.

skills used on Camera:

Skill 1; speed the camera moves over the path (default 175)
Skill 2; speed the camera Lerps over the path (default 2)
Skill 3; speed the camera's rotation Lerp's over the path (default 4)

action name:

ACT_Trig_pathFollow

what it does:

Moves the camera to the new position smoothly, lerps it over a path, while looking at an entity. Make sure the path is attached to the camera used, not to the trigger. Besides that, this action requires you setup speeds on the camera itself.

skills used:

String 1; Entity name the camera takes the position and rotation over from.

(If you want to make the camera look at the player, dont put in the WED name, but type "player" [without quotes] instead, code handles so it uses the handle to the player to look at)

Skill 1; the speed the camera uses to lerp to new position (default 1.1)

skills used on Camera:

Skill 1; speed the camera moves over the path (default 5)

Skill 2; speed the camera Lerps looking at the look at entity (default 5)

Triggers

action name:

ACT_Event_Trig

what it does:

Makes it possible to run functions you make yourself not included in this Template, whenever the player runs into a trigger with this action applied

skills used:

String 1; exact name of the function you want to run.

Lookat

action name:

ACT_Lookat_object

what it does:

Makes entities using the lookat code be able to look at this object (usually NPC's and Player)

skills used:

None

Collision

action name:

ACT_Collision_object

what it does:

Sets this entity to work with the Template's written collision detection code.

skills used:

Skill 1; Diameter of the collision object

Dialogue

action name:

ACT_Dialogue_object

what it does:

Starts a new dialogue when in range, and player pressed the action key.

skills used:

Skill 1; Range player needs to be in before dialogue can be triggered

String 1; Path to the file containing the dialogue.

String 2; Name the Character Dialogue should start

External Data

These are all the actions currently in the Template. Another thing to keep in mind is that the NPC , Dialogue, Key Setup and Player settings are all read from external files.

These files are in the Data folder and all it's subfolders in there.

These files have different extensions (.ds .char .ini), but can all be opened using notepad.

The formatting in these files are done by hand, and are supposed to work exactly like that with the code's. I wanted to write editors that would set all that up for you, but due to lack of time, and money, I never got round doing that.

Credits

Joozey – for learning me how Structs work

HeelX – for his Rudi project, there is a function in this Template used from that.

George – for making the AUM (and with that, this Template) available to everyone for free!

The community – you – for making use of this template (hopefully).