

Kurzanleitung Animationsframework

Sebastian Stelter
Hamburg BitBots

12. Juni 2018

1 Setup

1.1

Zum Starten des Animationsframeworks ist eigentlich gar nicht viel vorzubereiten. Wenn du das Framework mit einem Roboter starten willst, verbinde dich zuerst mit `ssh robotername@roboterdomain` mit dem Roboter, für unseren Minibot z.B. `ssh bitbots@odroid03`. Als nächstes muss das cm730-Board auf dem Roboter laufen. Dazu wechselst du mit `roscd bitbots_cm730` in den entsprechenden Ordner und startest mit `roslaunch launch/cm730_standalone.launch robotertyp:=true` das entsprechende Launchscript.

Wichtig ist auch, dass du die `ROS_MASTER_URI` auf deinem Computer festlegst. Für unseren Minibot gibst du dafür in einem neuen Terminal

```
export ROS_MASTER_URI="http://odroid03:11311" ein.
```

Nun sind alle nötigen Vorkehrungen getroffen und wir führen den Befehl `rqt` aus. Unter Plugins können wir nun Animation Framework auswählen und mit dem Animieren beginnen.

Beim Starten des Plugins öffnet sich zunächst ein Fenster, in dem nach dem Pfad gefragt wird, in dem die Animationen gespeichert werden sollen. Dieser Pfad ist dann relevant, wenn wir eine Animation auf dem Roboter abspielen wollen, denn dafür muss die Animation zunächst auf den Roboter kopiert werden. Dieser Pfad muss dementsprechend mit einem Pfad übereinstimmen, aus dem der Animation Server auf dem Roboter versucht, die Animationen zu lesen. Für einen Minibot ist bereits ein defaultwert angegeben, der Pfad für unsere anderen Roboter sieht ähnlich aus.

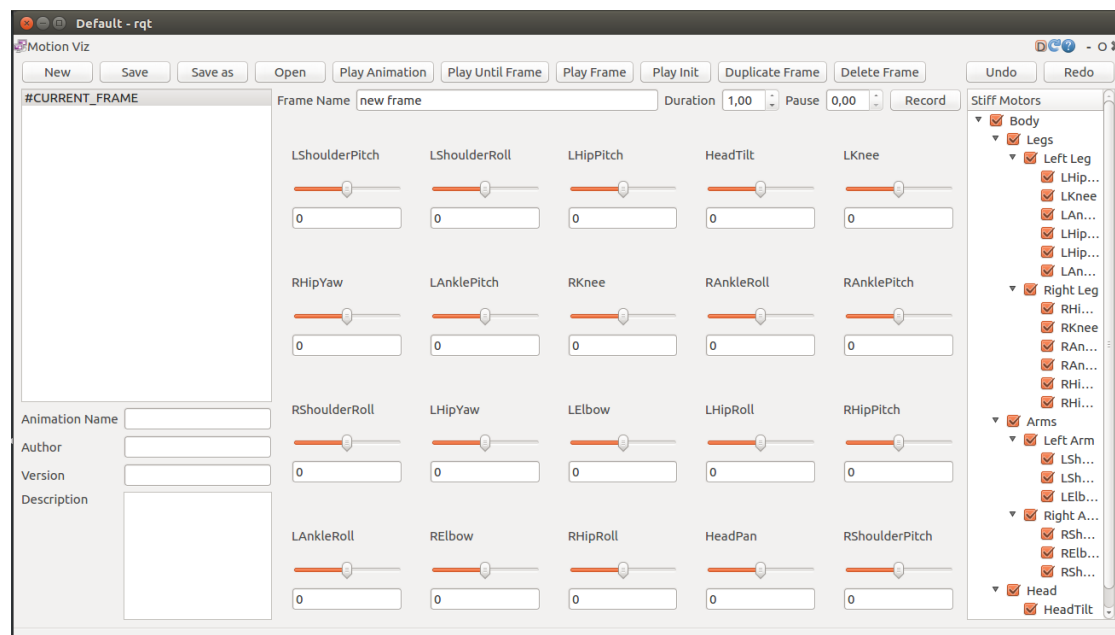
1.2

Falls du grade keinen Roboter zur Hand hast, musst du einige andere Vorkehrungen treffen, um das Framework zu starten. Da wir nun keinen Roboter haben, müssen wir zunächst einmal `roscore` starten. Unser Programm erwartet neben einem funktionierenden ROSCore auch Daten von den Motoren. Deshalb müssen wir jetzt in einem neuen Terminal ein ROSBag starten, das Motorenwerte enthält. dazu führen wir in einem neuen Terminal

`roslaunch bitbots_meta/bitbots_tools/bitbots_recordui_rqt/init.bag` aus.

In einem dritten Terminal starten wir mit `rqt` unser Framework und wählen unter Plugins Animation Framework aus. Wenn du die recordUI auf diese Weise startest kannst du natürlich keine Animationen abspielen. Der Animationspfad, der am Anfang abgefragt wird spielt hier daher auch keine Rolle.

2 Benutzeroberfläche



Auf den ersten Blick sieht die GUI vielleicht etwas unübersichtlich aus, im Grunde ist die Handhabung aber relativ einfach.

2.1



Ganz oben befindet sich zunächst eine Leiste mit einigen Knöpfen. Ganz links befindet sich *New*. Dieser Knopf setzt alles auf die ursprünglichen Einstellungen zurück und löscht alle geladenen Keyframes (außer `#CURRENT_FRAME`). Direkt daneben befindet sich *Save*, um eine Animation zu speichern. Damit das Speichern funktioniert, muss links in dem Feld *Animation Name* vorher ein Name eingegeben werden. Falls die Animation schon einmal gespeichert wurde, überschreibt dieser Knopf die vorherige Version der Animation. *Save As* speichert die Animation unter einem neuen Namen. Rechts davon befindet sich *Open*, um eine bereits existierende Animation zu öffnen. Bereits geladene Keyframes werden dadurch verworfen.

Der nächste Knopf, *Play Animation*, spielt alle geladenen Keyframes der Reihe nach ab. *Play Until Frame* spielt die Animation ab dem ersten Frame ab und stoppt beim aktuell ausgewählten Frame. *Play Frame* spielt nur den aktuell ausgewählten Frame ab. *Play Init* bewegt den Roboter in die Init-Position.

Rechts davon befindet sich *Duplicate Frame*, was eine Kopie des ausgewählten Frames erzeugt, sowie *Delete Frame*, um den ausgewählten Frame zu löschen. Die letzten beiden Knöpfe sind *Undo* und *Redo*, um Schritte rückgängig zu machen oder wiederherzustellen.

A horizontal control bar with a light gray background. It contains four main elements: a text input field labeled 'Frame Name' with the text 'new frame' inside; a 'Duration' label followed by a numeric input field showing '1,00' and a small downward arrow; a 'Pause' label followed by a numeric input field showing '0,00' and a small downward arrow; and a button labeled 'Record'.

Unter dieser Leiste befindet sich eine weitere Leiste mit Knöpfen. In dem Feld *Frame Name* wird der Name des jeweiligen Frames angegeben, unter dem er gespeichert werden soll. Jeder Frame muss einen einzigartigen Namen haben. In dem Feld *Duration* wird angegeben, wie lange der Roboter für die Bewegung vom vorherigen Keyframe zu diesem Keyframe brauchen soll. In dem Feld *Pause* wird angegeben, wie lange der Roboter nach Erreichen des Keyframes in diesem Frame verweilen soll, bevor er versucht, den nächsten Keyframe zu erreichen. Mit dem Button *Record* wird ein Frame aufgenommen. Der neue Frame taucht dann auf der linken Seite auf.

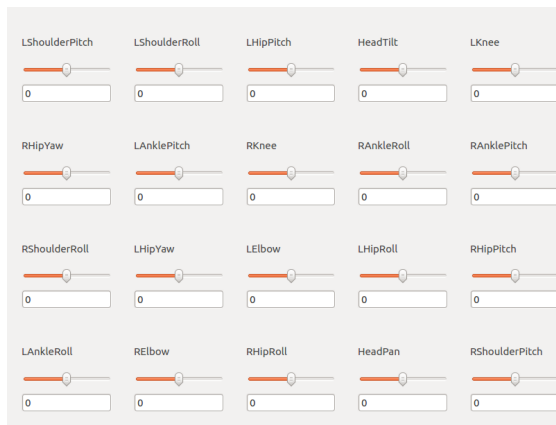
2.2

A vertical panel with a light gray background. The top part is a large, empty rectangular area with a thin border, labeled '#CURRENT_FRAME' in the top-left corner. Below this area is a sidebar with a darker gray background. The sidebar contains four labels with corresponding input fields: 'Animation Name' with a single-line text input; 'Author' with a single-line text input; 'Version' with a single-line text input; and 'Description' with a multi-line text input area.

Auf der linken Seite befindet sich oben die Liste aller Frames. Per drag-and-drop kann

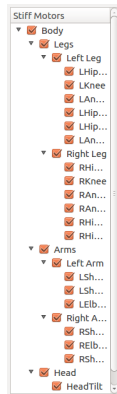
die Reihenfolge der Frames verändert werden. Hier wird auch ausgewählt, welcher Frame mit *Play Frame*, bzw bis zu welchem Frame mit *Play Until Frame* abgespielt wird. Darunter kann in dem Feld *Animation Name* der Name der Animation angegeben werden. Dies ist der Name, unter dem die Animation später gespeichert wird. Darunter können zusätzlich der Author, die Version und eine kurze Beschreibung angegeben werden. Alles außer dem Namen der Animation sind dabei optional.

2.3



In der Mitte befinden sich viele Schieberegler. Damit werden für jeden Frame die Positionen der Motoren eingestellt. Welcher Motor sich wo befindet, findest du weiter unten.

2.4



Ganz rechts befindet sich noch eine Liste der Motoren. Wenn du eines der Kästchen anklickst, wird der entsprechende Motor, bzw. die entsprechenden Motoren von steif auf locker, bzw. von locker auf steif gestellt. Ein Motor, der locker gestellt ist, also rechts nicht angekreuzt ist, wird bei den Schiebereglern in der Mitte ausgegraut und kann keinen Wert annehmen.

3 Unsere erste Animation

Jetzt, da wir unsere Umgebung verstanden haben, möchten wir unsere erste eigene Animation erstellen. In diesem Tutorial erstellen wir eine Animation, in der der Roboter den linken Arm hebt und winkt.

Als erstes brauchen wir einen Ursprungsframe, in dem der Roboter auf dem Boden steht. Wenn wir den Roboter starten, sollte er sich bereits in dieser Position befinden, deshalb nehmen wir genau diese Position als Frame auf. Als nächstes müssen wir den Arm des Roboters heben. Dazu wählen wir links wieder `CURRENT_FRAME` aus und erhöhen an den Reglern den Wert von `RShoulderPitch` auf ca 80. Mit Play Frame können wir jetzt immer wieder ausprobieren, ob uns der Frame gefällt. Wenn er so aussieht, wie wir das möchten, nehmen wir den Frame auf.

Jetzt wollen wir, dass der Roboter winkt. Dazu müssen wir die Position von `RElbow` ändern. Wir setzen `RElbow` beispielsweise auf -25 und nehmen den Frame auf. Das gleiche machen wir jetzt noch ein mal in die andere Richtung: wir setzen `RElbow` auf 25 und nehmen auch diesen Frame auf. Diese Animation können wir jetzt schon testen.

Wir möchten jetzt, dass der Roboter erst nach links und dann nach rechts winkt. Dazu tauschen wir einfach die Reihenfolge von Frame 3 und 4.

Außerdem möchten wir, dass der Roboter zwei mal winkt, anstatt nur einmal. Dazu erstellen wir mit Duplicate Frame eine Kopie von Frame 3 und 4 und fügen sie am Ende an. Nach dem Winken soll der Roboter sich natürlich auch wieder in die Ausgangsfunktion zurückbewegen, dazu kopieren wir mit Duplicate Frame den ersten Frame und setzen ihn ganz ans Ende unserer Animation.

Nachdem wir unsere Animation getestet haben und vollkommen damit zufrieden sind, geben wir der Animation unten links einen Namen, z.B. "minibot_winken". Wir geben unseren Namen als Author an, vielleicht Version 1.0 und eine kurze Beschreibung, was unsere Animation tut. Zum Schluss speichern wir unsere Animation mit Save As.

4 Hilfe, alles kaputt!

Wenn irgendetwas nicht funktionieren sollte, schau einfach kurz in die folgende Liste hinein und guck, ob dein Problem irgendwo genannt ist:

Could not find ROS master.

Prüfe, ob das Launchscript läuft und die `ROS_MASTER_URI` gesetzt wurde, bzw. lokal in einem deiner Terminals ein `roscore` läuft.

Das Plugin öffnet sich nicht, im Terminal steht wait

Es läuft kein Rosbag oder es laufen nicht alle nötigen nodes auf dem Roboter.

Das Launchscript gibt ganz viele gelbe Zeilen mit CID missing zurück

Irgendein Motorenkabel steckt nicht richtig im Roboter.

Ich erreiche den Roboter mit ssh nicht

Roboter anschalten, Kabel rein stecken, gucken, ob dein Laptop im BitBots-WLAN ist.

Wenn ich X tue, stürzt das Programm ab oder verursacht komisches Verhalten

Öffne ein Issue im Gogs und sag mir(Sebastian) bescheid, ich kümmer mich darum.

A Shortcuts

New	Ctrl + N
Save	Ctrl + S
Save As	Ctrl + Shift + S
Open	Ctrl + O
Play Frame	Ctrl + P
Play Animation	Ctrl + Shift + P
Play Until Frame	Ctrl + Alt + P
Play Init	Ctrl + I
Duplicate Frame	'+'
Delete Frame	'-'
Record	Space
Undo	Ctrl + Z
Redo	Ctrl + Y