Character Bodypart Tutorial

By Popcorn

INTRODUCTION

Level: Intermediate

Have you ever tried to make a platform game, but given up on the fact that you just aren't good enough with the graphics? If you have used TGF2 or MMF2 for a while, you probably have tried to make a platform game at least once before. And you probably have got most of it working also, till it comes to the animation of the characters. Yeah, TGF2/MMF2 have lots of animated characters for you to use, but maybe you don't find any that fit your game, or you simply don't want to use something that you haven't created yourself. TGF2/MMF2 include a better tool for character animation than their predecessors, but still it requires you to have way above average drawing skills to make a decent animation with the animation editor.

This tutorial is for all of you that don't have these particular skills. I don't have them, and I am not going to learn to draw animations for my games when I can use the eventeditor to make the animations for me. Yes, you read right. TGF2/MMF2 has a very powerful eventeditor, and that's the tool we are going to use to make our animations! Of course, we have to use the animation editor as well, but not to make animations, just to draw one-frame pictures like head, body, feet and hands, etc.

This tutorial is going to teach you how to make a fully animated character out of body parts you make in the animation editor.

You will among other things learn how to create items and weapons that your character can pick up and use and how to create cloth and armor that your character can pick up and wear.

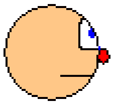
It is recommended that you allready know the basics of The Games Factory 2 / Multimedia Fusion 2 before doing this tutorial.

PART I - CREATING THE PLAYER CHARACTER

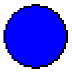
1. Making the body parts

Let's start with drawing some simple bodyparts for our character. What we need is a body, a head, a foot and a hand. Later you might want to add more, like weapons, helmets, armor, etc, but for now let's stick with these four. You totally decide for yourself how the bodyparts will look like. As stated earlier, this tutorial doesn't tech you how to draw. For the head for instance, draw a filled circle, place an eye a mouth and a nose. Yes, one eye only, as the head is looked at from the side! These are the bodyparts I'll use for this tutorial. You can use them if you don't want to draw your own.

Head



Body



Foot

Hand

The head's hotspot should be placed at the bottom center, or at the root of the neck if the neck is not at the bottom center of the image.
For the other bodyparts it's ok to place the hotspot in the center. Later you would also want to place the actionpoints, but you don't need to care about that now. In the frame editor, duplicate the foot and hand so that there are two feet and two hands. Do NOT clone them, only duplicate. We will seperate them in the eventeditor later.

2. Animating the body parts

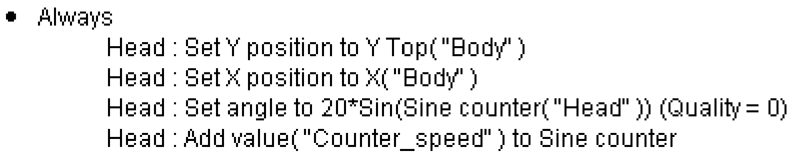
Before we open the event editor, we want to change the names of the alterable values we are going to use. In the frame editor, select all the bodyparts. In the properties window, click the A-Z icon, labeled Values. Click on New. Double click on Alterable Value A and change it's name to Sine counter. Now select only the foot and hand and change the name of Alterable Value B to id. Since there are two feet and two hands, we can use this id value to separate them, instead of having to create two different foot and hand objects.

The last preparation we need is to make a counter. Name this counter Counter_speed and set it's initial value to 3.

Go to the event editor. We will organize all our events in eventgroups. Not because we need it, but because it will help us read and understand the events better. Make an event group called Body. In here we will add all the events that control where and how the bodyparts will act.

The head

Will we start with the head. Look at the event below and try to create it yourself in the event editor.



```
• Always
        Head : Set Y position to Y Top( "Body" )
        Head : Set X position to X( "Body" )
        Head : Set angle to 20*Sin(Sine counter( "Head" )) (Quality = 0)
        Head : Add value( "Counter_speed" ) to Sine counter
```

Don't worry, I'll explain the actions for you.
The first two actions should be quite self-explanatory. Since the head's hotspot is located at the root of the neck, we can just tell it to set it's Y position to the TOP Y of the body. Then we just set the head's X position to the Body's X position. The third and fourth actions are what animates the head. The head is animated by that it's angle is constantly changing.
Here is what you set the angle to:

20*Sin(Sine counter("Head"))

20 is how many degrees you want the head to bend. Play around with this value. 360 will get the head to do a full circle. The rest of the action calculates the SINE value of the Head's Sine counter.

The last action adds the current value of the speed counter to the Head's Sine counter value. The head will rotate because this value is constanly increasing. Delete this action and nothing will happen. The Counter_speed's value obviously control how fast the head will rotate back and forth.

The feet

Now that you know how to rotate an object using sine, it shall be easy to follow the rest of the tutorial. The feet and hands will not rotate, but their positions will be set using sine, in the same way we set the angle for the head. Since we have two feet but only one object called foot, we need to do some extra code to make them work as two different feet. We allready gave it an id value slot. Not let's give the two feet an id and then test if the id is 1.

```
• Start of Frame
        Foot : Spread value 0 in id

• Start of Frame
+ Id of Foot = 1
        Foot : Set Sine counter to 180
```

Spreading 0 into an value will set the first object's value to 0, the second object's value to 1 etc.. That means that one of the feet has id=0, and the other one has id=1.
By adding 180 to one of the feet's Sine counter, we make this foot start in the other direction than the other foot.

The next event very much looks like the one we did for the head, only this time we don't worry about the angle, but does the calucation on the x position instead.

```
• Always
        Foot : Set Y position to Y Bottom( "Body" )
        Foot : Set X position to X( "Body" )+10*Sin(Sine counter( "Foot" ))
        Foot : Add value( "Counter_speed" ) to Sine counter
```

The hands

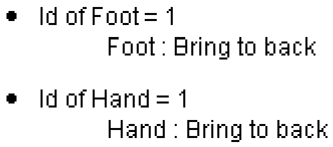
For the hands do exactly the same as you did with the feet. The only difference is that you set it's Y position to the center of the body instead of at the bottom.

```
• Start of Frame
        Hand : Spread value 0 in id

• Start of Frame
+ Id of Hand = 1
        Hand : Set Sine counter to 180

• Always
        Hand : Add value( "Counter_speed" ) to Sine counter
        Hand : Set Y position to Y( "Body" )
        Hand : Set X position to X( "Body" )+10*Sin(Sine counter( "Hand" ))
```

Now you are almost all set. Give Body the movement you want, for instance the platform movement and play around with it! Remember to put all events regarding movement in an even group named Movement. You will have to add direction Left for the Head, as it's now will look at the right all the time. So copy the Head image and paste it in direction Left and flip it. In the event editor simply have an action to set it's direction to left when the player press the Left arrow key, and another event for right.

Final touches

You probably want the animation speed to match the movement speed of your character. So make an event to always set Counter_speed to speed of Body. You also want control the view order of the objects. For instance the back hand and foot should be placed behind the body. Create an event group named View Order and do all the order actions here. This group should be placed at the bottom of the event editor, after all other events are executed.



```
•  Id of Foot = 1
        Foot : Bring to back

•  Id of Hand = 1
        Hand : Bring to back
```

3. Carry a gun

Now you are free to add more body parts or items that your character will wear. One cool thing about this way of designing your character is that it is very easy to let your character carry different types of weapons and armor. The only thing you need is to have one picture of each weapon the character shall carry.

Next we will make a gun and make the character be shooting with it.

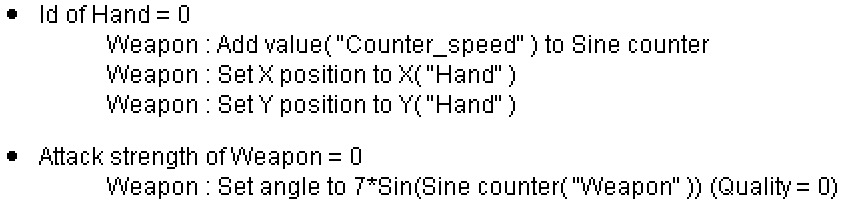
Create a new Active object and name it Weapon. Make it look like a gun.
The one I'm using looks like this:

This is for direction Right. Paste the image into direction Left and flip it so you have an image for both directions. Also make sure to set the direction of the weapon when the player press left and right. Do that in the same event you made earlier.
Give the weapon these alterable values: Sine counter and Attack strength.
And finally give it an alterable string named Type and set the value to gun.

The reason we give it a Type attribute is because we are later going to deal with several kinds of weapons, but still want to work with the same Active object.

Walking stance

In the Body event group, add these events:



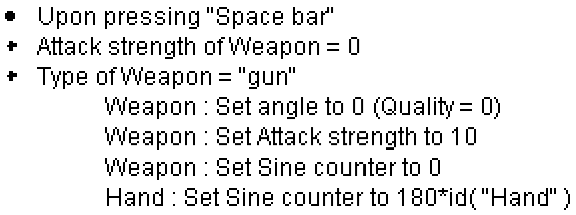
```
•  Id of Hand = 0
        Weapon : Add value( "Counter_speed" ) to Sine counter
        Weapon : Set X position to X( "Hand" )
        Weapon : Set Y position to Y( "Hand" )

•  Attack strength of Weapon = 0
        Weapon : Set angle to 7*Sin(Sine counter( "Weapon" )) (Quality = 0)
```

Notice that we have placed the gun the hand with id=0, which is the hand that is visible all the time. The other hand is behind the character.

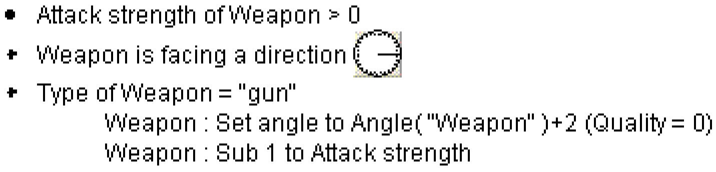
Fighting stance

When the player presses Spacebar, the character will do an action with the weapon, whatever weapon it carries. We use the Type attribute to check that the weaspon is a gun. When the player shoot, we want the hand and the gun to rotate upwards a little.

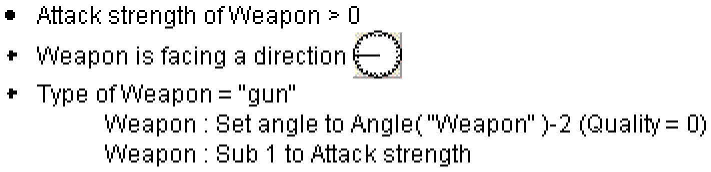
The events should be easy to understand:



```
• Upon pressing "Space bar"
• Attack strength of Weapon = 0
• Type of Weapon = "gun"
        Weapon : Set angle to 0 (Quality = 0)
        Weapon : Set Attack strength to 10
        Weapon : Set Sine counter to 0
        Hand : Set Sine counter to 180*id( "Hand" )
```



```
• Attack strength of Weapon > 0
• Weapon is facing a direction →
• Type of Weapon = "gun"
        Weapon : Set angle to Angle( "Weapon" )+2 (Quality = 0)
        Weapon : Sub 1 to Attack strength
```

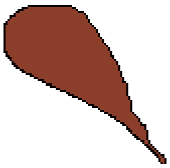


```
• Attack strength of Weapon > 0
• Weapon is facing a direction ←
• Type of Weapon = "gun"
        Weapon : Set angle to Angle( "Weapon" )-2 (Quality = 0)
        Weapon : Sub 1 to Attack strength
```

Now your character can walk around with a gun and shoot. Of course, the actual shooting you will have to code yourself :)

4. Weapon swing

Next we'll add another weapon. This time your character will carry a club and swing it on attack. Add a new animation in the ojbect called Weapon. The animation of course is still just one frame and can look like this:

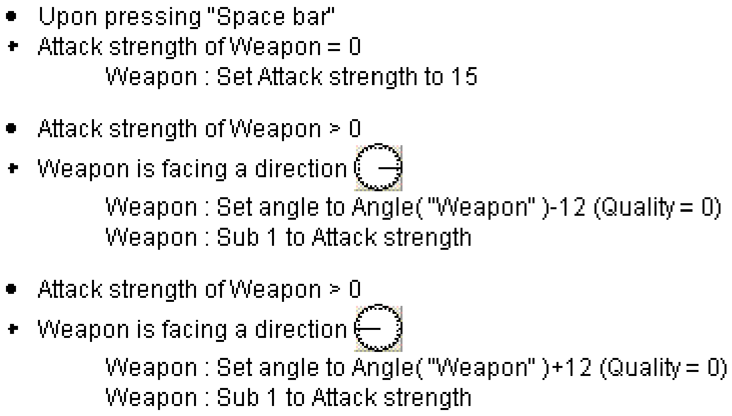


Set it's hotspot at where you want the character to hold it.
This is for direction Right. As you did with the gun. paste the image into direction Left and flip it so you have an image for both directions.

We allready have made the events for carrying the weapon, all we need to do to show the club is to set the Type attribute to club and make an event in the event editor to set the animation to Club if the Type of Weapon is club.

Fighting stance

Let's make the club swingable!
The events are pretty self-explanatory so I'll let them speak for themselves:



```
• Upon pressing "Space bar"
+ Attack strength of Weapon = 0
        Weapon : Set Attack strength to 15

• Attack strength of Weapon > 0
+ Weapon is facing a direction
        Weapon : Set angle to Angle( "Weapon" )-12 (Quality = 0)
        Weapon : Sub 1 to Attack strength

• Attack strength of Weapon > 0
+ Weapon is facing a direction
        Weapon : Set angle to Angle( "Weapon" )+12 (Quality = 0)
        Weapon : Sub 1 to Attack strength
```

5. Changing outfit and items

It is extremely easy to have your character wearing different sets of armor or clothes or carry different types of weapons, boots, helmets etc. All you need to do is to go to the animation editor and add an animation and draw the armor or weapon. Most the code is allready done, you only need to tell the objects which animation to display.

PART II - MAKING MULTIPLE CPU CHARACTERS

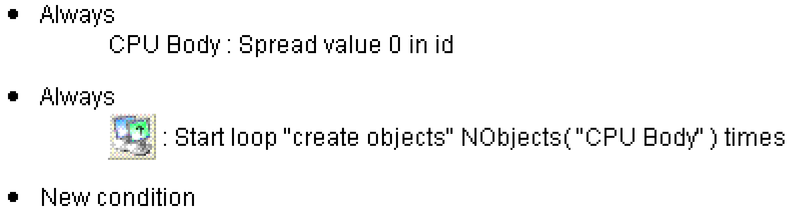
The challenge most people have by using characters made up by bodyparts, is to attach each part to the correct body if there are multiple instances of the character. If you have cpu characters in your game, you most probably want to have more than one on screen at the same time. To solve this we will use fastloops and alterable values to keep track of each bodypart.
It is recommended that you are at least a little experienced with fastloops before going further on with this tutorial.

1. Preparing the bodyparts

First, make new body parts for the cpu character. This time we make two hands and two feet just to make it a little simpler. Name the bodyparts CPU Body, CPU Head, CPU Foot, CPU Foot 2, CPU Hand and CPU Hand 2. Give all the bodyparts an id and a Sine counter attribute. The id of the cpu bodyparts will have a different role than the ones for the Player character. The id will tell each bodypart which body they belong to. Now give the Body a movement, for instance the Bouncing Ball movement with a speed of 10. Finally make some duplicates of the body. You don't need to make duplicates of the other bodyparts, as we will make them in the event editor.

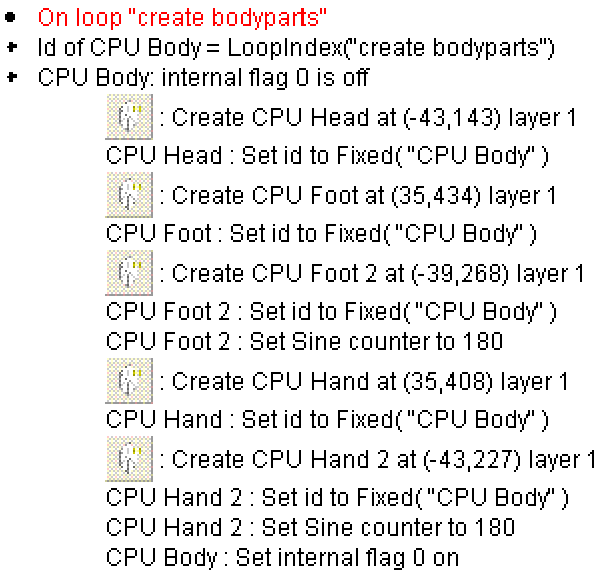
2. Creating the bodyparts and assigning id's

Make a new event group named CPU. Put all events regarding the computer controlled objects in this group.
Let's start with spreading values and making the loop:



```
• Always
        CPU Body : Spread value 0 in id

• Always
        : Start loop "create objects" NObjects( "CPU Body" ) times

• New condition
```

The idea now is to create new bodyparts as long as there are more bodies than one kind of bodyparts. If there are 10 bodies and 0 heads, we make more heads.
The following event has many actions, but it shouldn't be hard to understand them. In short, first we create an object, then we attach it to the body by setting the bodypart's id to the Body's fixed id. A fixed id is just a unique id. No duplicates have the same fixed id, and therefore we can use this id to attach the bodyparts. You'll see that the order of the actions are important.



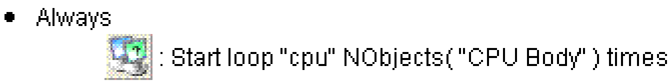
```
• On loop "create bodyparts"
+ Id of CPU Body = LoopIndex("create bodyparts")
+ CPU Body: internal flag 0 is off
          : Create CPU Head at (-43,143) layer 1
        CPU Head : Set id to Fixed( "CPU Body" )
          : Create CPU Foot at (35,434) layer 1
        CPU Foot : Set id to Fixed( "CPU Body" )
          : Create CPU Foot 2 at (-39,268) layer 1
        CPU Foot 2 : Set id to Fixed( "CPU Body" )
        CPU Foot 2 : Set Sine counter to 180
          : Create CPU Hand at (35,408) layer 1
        CPU Hand : Set id to Fixed( "CPU Body" )
          : Create CPU Hand 2 at (-43,227) layer 1
        CPU Hand 2 : Set id to Fixed( "CPU Body" )
        CPU Hand 2 : Set Sine counter to 180
        CPU Body : Set internal flag 0 on
```

Note that flag 0 is used to test if a body allready has bodyparts attached to it. This may be useful later in your game if enemies can be destroyed and new ones can be created in-game.
Also keep in mind that we make two hands and two feet for each Body. To figure out if a foot is a character's first or second foot, you can make an event to test if id of foot mod 2 equals 1. If it does it's the second foot and you can order it to the back.

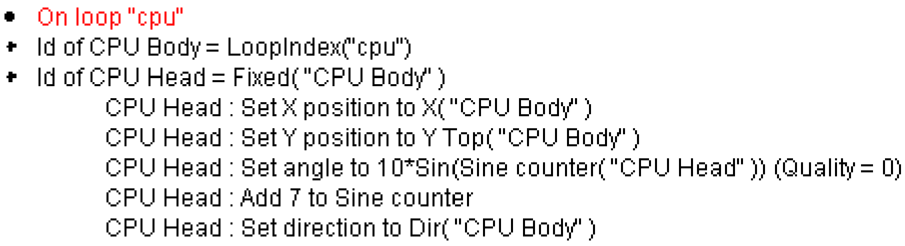
3. Attaching the bodyparts

Now that we have created bodyparts for each Body object, all that is left is to attach the parts to the bodies. We do that by always starting a loop that tests if the current Body object's fixed id equals a bodypart's id. If it does, it attaches the bodypart to the body that we test for.
Next follows a bunch of code, but all the events are quite similar and easy to understand, so I didn't feel the need to comment every event. Take your time to understand what each condtion and action does.

The CPU Head

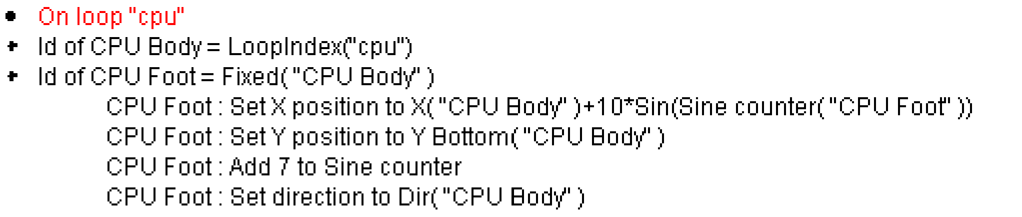


```
• Always
          : Start loop "cpu" NObjects( "CPU Body" ) times
```

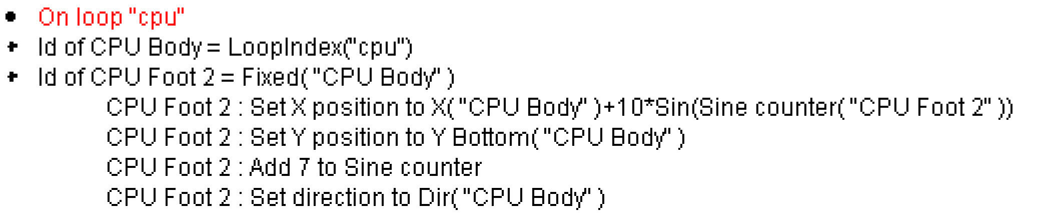


```
• On loop "cpu"
+ Id of CPU Body = LoopIndex("cpu")
+ Id of CPU Head = Fixed( "CPU Body" )
        CPU Head : Set X position to X( "CPU Body" )
        CPU Head : Set Y position to Y Top( "CPU Body" )
        CPU Head : Set angle to 10*Sin(Sine counter( "CPU Head" )) (Quality = 0)
        CPU Head : Add 7 to Sine counter
        CPU Head : Set direction to Dir( "CPU Body" )
```

The CPU Feet

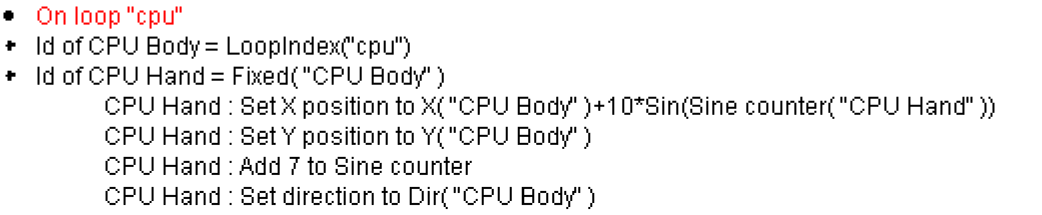


```
●  On loop "cpu"
✦  Id of CPU Body = LoopIndex("cpu")
✦  Id of CPU Foot = Fixed( "CPU Body" )
          CPU Foot : Set X position to X( "CPU Body" )+10*Sin(Sine counter( "CPU Foot" ))
          CPU Foot : Set Y position to Y Bottom( "CPU Body" )
          CPU Foot : Add 7 to Sine counter
          CPU Foot : Set direction to Dir( "CPU Body" )
```

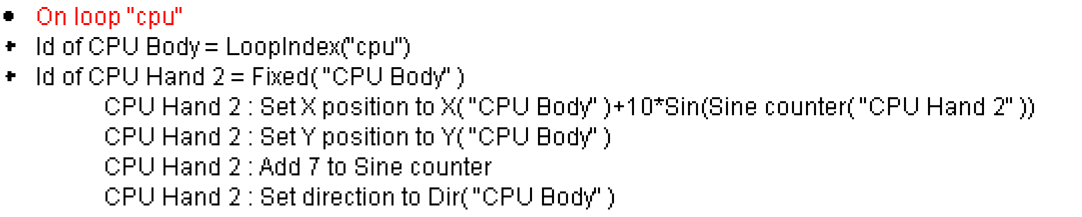


```
●  On loop "cpu"
✦  Id of CPU Body = LoopIndex("cpu")
✦  Id of CPU Foot 2 = Fixed( "CPU Body" )
          CPU Foot 2 : Set X position to X( "CPU Body" )+10*Sin(Sine counter( "CPU Foot 2" ))
          CPU Foot 2 : Set Y position to Y Bottom( "CPU Body" )
          CPU Foot 2 : Add 7 to Sine counter
          CPU Foot 2 : Set direction to Dir( "CPU Body" )
```

The CPU Hands



```
●  On loop "cpu"
✦  Id of CPU Body = LoopIndex("cpu")
✦  Id of CPU Hand = Fixed( "CPU Body" )
          CPU Hand : Set X position to X( "CPU Body" )+10*Sin(Sine counter( "CPU Hand" ))
          CPU Hand : Set Y position to Y( "CPU Body" )
          CPU Hand : Add 7 to Sine counter
          CPU Hand : Set direction to Dir( "CPU Body" )
```



```
●  On loop "cpu"
✦  Id of CPU Body = LoopIndex("cpu")
✦  Id of CPU Hand 2 = Fixed( "CPU Body" )
          CPU Hand 2 : Set X position to X( "CPU Body" )+10*Sin(Sine counter( "CPU Hand 2" ))
          CPU Hand 2 : Set Y position to Y( "CPU Body" )
          CPU Hand 2 : Add 7 to Sine counter
          CPU Hand 2 : Set direction to Dir( "CPU Body" )
```

4. Final notes

Now you should have learned the basics of attaching objects to each other using fastloops and alterable values. Do experiment upon what you have learned in this tutorial to go even further than this tutorial. I hope you have enjoyed reading this tutorial as much as I did enjpy making it.
If you are stuck, have questions about the tutorial or have any comments at all, feel free to e-mail me at ilikepopcorn@hotmail.com!