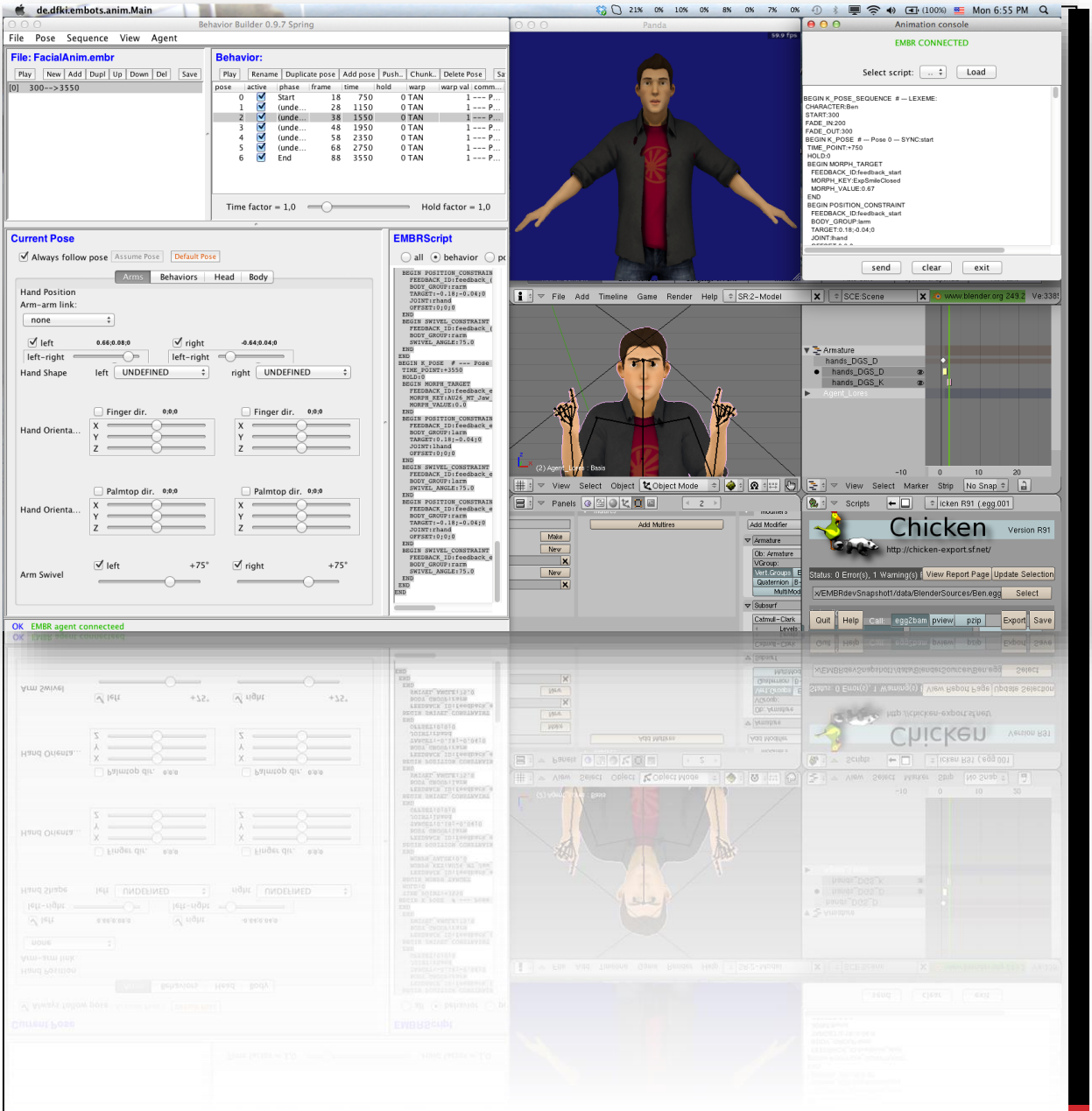


EMBR CURRENT STATE

LAST MODIFIED: JANUARY 2014.

This document reports all the current knowledge we have about EMBR tools such as Behavior Builder, Ben, EMBR Console, Blender, etc. Some of the issues addressed include installing, running, updating and combining the abovementioned tools and intention to work on them. We plan on updating this document based on input we have form Alexis, the EMBOTS Team and the modifications we do. So, make sure you are reading the latest version. ☺

EMBR CURRENT STATE

INDEX

FOLDERS ASSOCIATED:

bla

Bla

Bla

SYSTEM REQUIREMENTS

Java

Python

NeatBeans

PART I. CREATE AND RUN AN EMBR SCRIPT ON BEN

There are four available ways to play with EMBR and Ben (we call them combinations):

A. BehaviorBuilder GUI and Ben in a packaged version of EMBR

Running

- First, you can start the Java behaviorBuilder program by typing the "sh" command or double-clicking the "startEMBR" file.
- Then, you can double-click on the ben.osx_i386.p3d file to start the animation window.
- At this point you should change the "Agent" to "Ben" in behaviorBuilder using the Agent menu.
- Now, you can send commands to the character using the "Play" button.

Offering

The packaged version includes a Ben with lower textures (so that it occupies less memory and can run on MacBook Air) and restrictions on the supported morph targets and handshapes already implemented.

The interface of Behavior Builder adds some restrictions on the supported embrscripts (e.g. parallel POSE_SEQUENCES) but can be used for creating POSE_SEQUENCE(s) in an easy way to interact with Ben.

B. Console and Ben in a packaged version of EMBR

Running

- First, you can start the AnimConsole.jar by double clicking it.
- Then, you can double-click on the ben.osx_i386.p3d file to start the animation window.
- At this point you should see EMBR CONNECTED in top of the Console window.
- Now, you paste the script in the window; hit “Load” and then the "Send" button.

Offering

Same as “BehaviorBuilder GUI and Ben in a packaged version of EMBR” but there is no restriction on the supported handshapes and target morphs or on the parallel sequences. Of course it is not user friendly for creating a gesture pose by pose.

C. BehaviorBuilder GUI and Ben in EMBR’s development snapshot (better Ben)

Running

- First, you can start the Java behaviorBuilder program by typing the "sh" command or double-clicking the "startEMBR" file.
- Open a command line in the EMBRdevSnapshot1 folder. Provided that the panda3D SDK v.1.7 is installed on your machine (it might also require the installation of Cg Toolkit), EMBR should run by typing "python2.5 main.py". A window of Ben in Panda 3D is going to pop up. (Ignore the rest of errors in the console)
- *Note:* the computer must be connected to the Internet. Once installed panda3D tries to access Internet in the first run. Installing Panda3D rev1 (this message appears when running ben file .p3d for the first time in a computer where panda runtime engine is installed)
- At this point you should change the "Agent" to "Ben" in behaviorBuilder using the Agent menu.
- Now, you can send commands to the character using the "Play" button.

Offering

The development snapshot of EMBR includes now a refined version of Ben (with higher resolution textures and normal maps) but this cannot reach the high quality renderings achieved offline using a ray-tracing renderer capable of global illumination and subdivision surfaces[1]. The quality of the real time panda engine cannot reach the quality of off line global illumination. Now, on the other hand the development snapshot allows us to extract and modify the incorporated assets:

- modify the joint boundaries

- Include MPEG-4 FAPs morph targets (requires changes in the BehaviorBuilder GUI, Alexis can do that for us once we are done with the development of the new morph targets in Blender)

- Include more hand-shapes (to_be_verified: requires changes in the BehaviorBuilder GUI)

D. Console and Ben in EMBR's development snapshot (better Ben)

Running

- First, you can start the AnimConsole.jar by double clicking it.
- Open a command line in the EMBRdevSnapshot1 folder. Provided that the panda3D SDK v.1.7 is installed on your machine (it might also require the installation of Cg Toolkit), EMBR should run by typing "python2.5 main.py". A

[1] Cannot be used for animation. It is for still image only.

window of Ben in Panda 3D is going to pop up. (Ignore the rest of errors in the console)

- *Note:* the computer must be connected to the Internet. Once installed panda3D tries to access Internet in the first run. Installing Panda3D rev1 (this message appears when running ben file .p3d for the first time in a computer where panda runtime engine is installed)
- At this point you should see EMBR CONNECTED in top of the Console window.
- Now, you paste the script in the window; hit “Load” and then the "Send" button.

Offering

The development snapshot of EMBR includes now a refined version of Ben (with higher resolution textures and normal maps) but this cannot reach the high quality renderings achieved offline using a ray-tracing renderer capable of global illumination and subdivision surfaces. The quality of the real time panda engine cannot reach the quality of off line global illumination. Now, on the other hand the development snapshot allows us to extract and modify the incorporated assets:

- modify the joint boundaries,

- include MPEG-4 FAPs morph targets (no changes required for the console),

- include more hand-shapes (no changes required fro the console), and

- no restrictions in the EMBRscript and total handling of time.

PART II. CONVERT VCOM3D TO EMBR

This stage is almost at the final stage. Future work needed include:

- Redefine some of the coefficients of the linear conversion of the hand position based on the looser joint boundaries.
- Mapping of the hand-shapes based on the new hand-shapes also included.
- Test an extended set of examples.
- Finish the related documentation.

PART III. JOINT BOUNDARIES AND MESH IMPROVEMENT

A. Joint Boundaries

Modify the CharacterConfiguration.xml file accordingly. The modification made include:

B. Mesh Improvement

While displaying the better Ben there was some issues with the red blouse coming over the jacket. We kind of solve this by changing the mesh for the blouse. It is not perfect but kind of work. Maybe we can ask an artist to make the correct improvement and even more to get the jacket with shorter sleeves so maybe we could solve the Poke-Through problem.

PART IV. INCLUDE MORE HAND-SHAPES IN EMBR

- Add lines to ben.py for new handshapes.

- Add lines to CharacterConfiguration.xml file for new handshapes.

- Create (or duplicate) BVH files for new handshapes and add to the data folder. Use a free BVH editor to do this.

- Students can help with this over the summer.

Alexis notes:

As you will notice, two files need to be updated:

1: embr.py should be updated so that the pada .bam assets files are loaded with the charecter

(l. 54) replace:

```
self.ben.setActor( self.dataPath + 'models/ben/ben.bam',{},{})
```

with

```
self.ben.setActor( self.dataPath + 'models/ben/ben.bam',{"hands_ASL_6":self.dataPath + "models/ben/animations/Ben-ASL_6.bam"},{})
```

--> you actually register a dictionary entry into the first set of curly brackets: the motion described in the 'Ben-ASL_6.bam' file will be registered as a motion assets under the name 'hands_ASL_6'

2: in Ben.py, inform the embr runtime that a panda animation should be parsed, translated to the EMBR conventions (bvh like) and register under the keyword of your choice (e.g 'hands_ASL_6')

(l. 290) add:

```
self.addAnimation(self.name,'hands_ASL_6')
```

save the two files and the new keypose can be recalled in the EMBRScript language (also cf. eclosed picture):

```
BEGIN POSE_TARGET

  FEEDBACK_ID:feedback_start
```

BODY_GROUP:rhand

POSE_KEY:hands_ASL_6

For convenience, you might want to update the handshape static declaration in the JAVA sourcecode of the behavior builder.

PART V. INCLUDE MPEG-4 FAPS IN EMBR

Let's first see what is a proprietary MPEG-4 face model[2].

For a face model to be MPEG-4 compliant, it has to be able to execute all FAPs according to Sections 2.2.1 and 2.2.2. Therefore, the face model has to have at least as many vertices as there are FPs that can be animated. Thus, an MPEG-4-compliant face model may have as little as 50 vertices. Such a model would not generate a pleasing impression. We expect to require at least 500 vertices for pleasant and reasonable face models (Figure 2.3).

(We have to find out here how many vertices Ben's face has.)

A proprietary face model can be built in 4 steps:

1. We build the shape of the face model and define the location of the feature points on the face model according to Section 2.2.1 and Figure 2.2.
2. For each FAP, we define how the feature point has to move. For most feature points, MPEG-4 defines only the motion in one dimension. As an example, we consider FAP 54, which displaces the outer right lip corner horizontally. Human faces usually move the right corner of the lip backward as they move it to the right. *It is left to the face model designer to define a subjectively appealing face deformation for each FAP.*
3. After the motion of the feature points is defined for each FAP, we define how the motion of a feature point affects its neighboring vertices. This mapping of feature point motion onto vertex motion can be done using lookup tables such as FAT (section on definition of animation rules using FAT) [17], muscle-based deformation [8, 10, 11], distance transforms [18] or cloning from existing models [19, 20] (see Chapter 4).
4. For expressions, MPEG-4 provides only qualitative hints on how they should be designed (Table 2.4). Similarly, visemes are defined by giving sounds that correspond to the required lip shapes (Table 2.3). FAP 1 and 2 should be designed with care because they will generally be used for visually appealing animations.

A. MPEG-4 FAPs For EMBR

The MPEG-4 standard [1, 2] includes techniques designed for specifying and modeling the human face. A face is described with a total of 80 FDPs (Facial Definition Parameters). Each FDP is a point on the face (for example, the tip of the nose, the left corner of the left eye, the top of the right ear, etc.) and is defined with a three-dimensional vector giving the exact location in space. The FDPs refer to a neutral face, i.e., a face with no particular expression, eyes open, mouth closed and other constraints as defined in MPEG-4.

[2] [Pandzic02] MPEG-4 Facial Animation - The standard, implementations, applications, Igor S. Pandzic, R. Forchheimer, Editors, John Wiley & Sons Ltd.

Figure 2.1 shows the FDPs in MPEG-4 FBA [1, 2]. The solid dots (•) show FDPs that are directly affected by one or more FAPs. The open dots (o) show FDPs that are not affected by or only indirectly affected by FAPs. For example, feature point 8.1 is marked with a solid dot because it is directly affected by FAP # 51. Feature point 8.9 and feature point 8.10 are marked with open dots, because there are no FAPs that affect those points directly, but a good face model will move those points along with feature point 8.1.

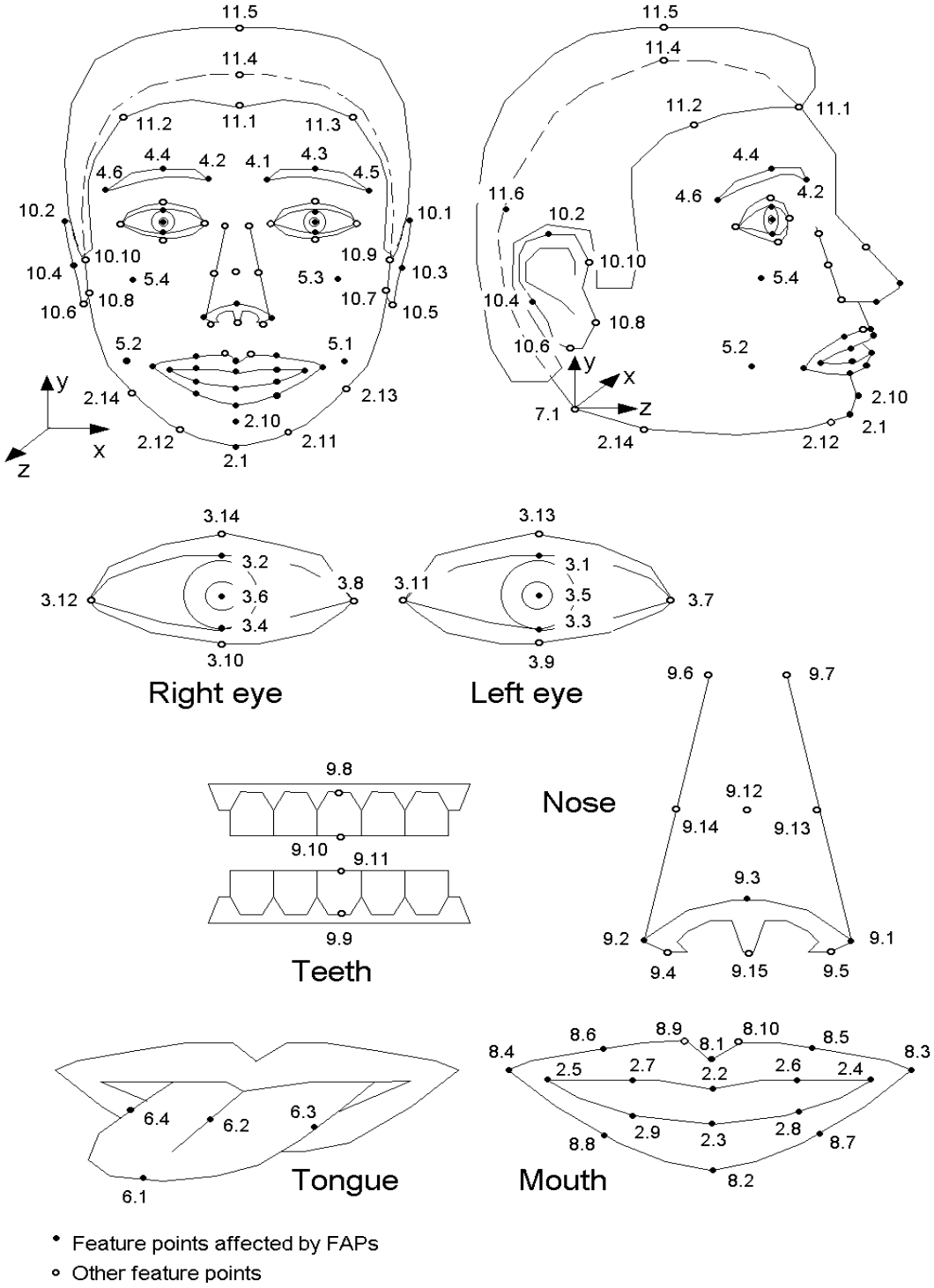
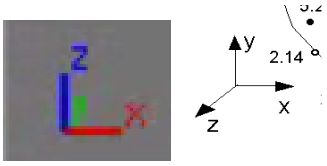


Figure 2.1 FDPs (Facial Definition Parameters) and FAPs (Facial Animation Parameters) in MPEG-4 [1]

Now, it is important to notice that Blender has a different orientation system as shown in

the following figure from the MPEG-4 standard that the tables in the next pages are considering. Though both orientation systems are right handed, we can just think of z instead of y and –y instead of z.



Mapping MPEG-4 FDPs

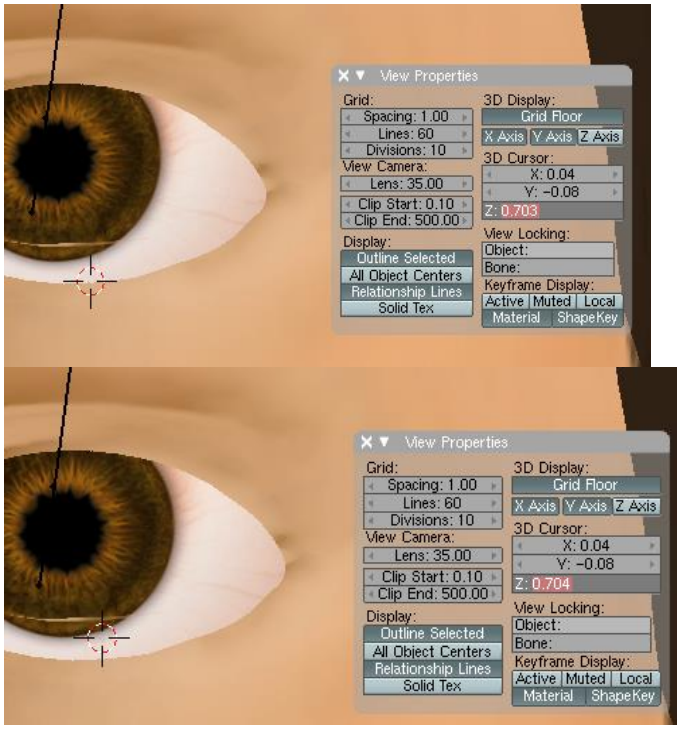
At the early stage of this work we would like to give priority to the following FAPS (highlighted with yellow color). But, since we are hiring an artist to a more detailed result it would be nice to include all the displacement FAPs, ignoring the high level and angle FAPs.

TABLE 1: FAP GROUPS

Group	Number of FAPs
1:visemes and expressions (not to be included)	2
2: jaw, chin, inner lowerlip, cornerlips, midlip	16
3: eyeballs, pupils (both of them not to be included), eyelids	12
4: eyebrow	8
5: cheeks	4
6: tongue	5
7: head rotation (not to be included)	3
8: outer lip positions	10
9: nose	4
10: ears	4

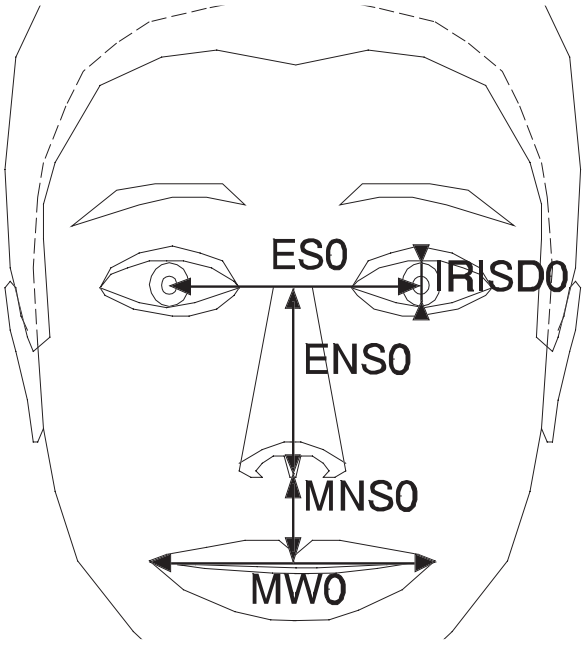
All the low level FAPs (angle and displacement FAPs) are expressed in terms of FAPU (Facial Animation Parameter Units). FAPUs correspond to fractions of distances between some key facial features. The fractions are represented with fixed precision. The use of FAPUs allows interpretation of FAPs on any face model in a consistent way, producing reasonable results in terms of expression and movement. To calculate the FAPU for our character Ben we will use the values in Blender defining the position of the FDPs we are interested in. To access these values in Blender choose **View>View Properties** in the 3D View header. The unit used by Blender to define these points doesn't really mater since we are using Blender again for creating the FAPs. But the scale of the unit is important. Working on the face, an area small in proportion to the Scene and the character, require more precision. The current smaller distance displayed

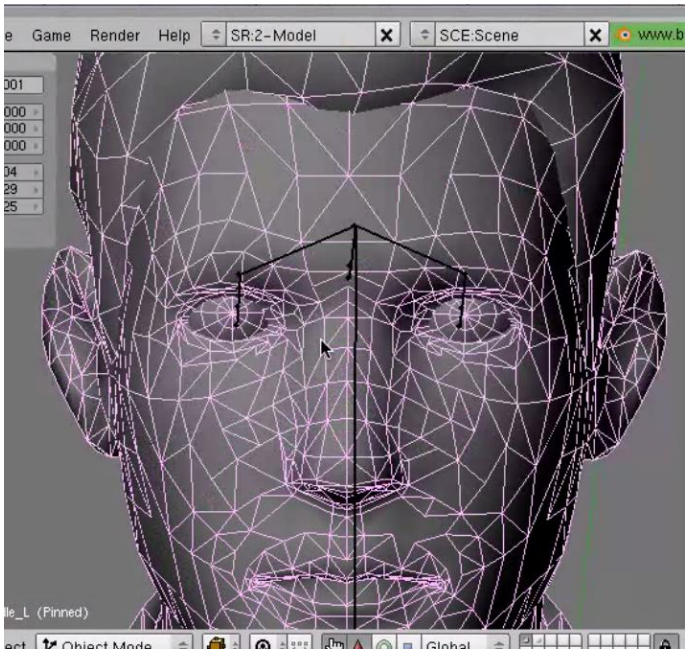
for the 3D cursor is 0.01 and that is almost equal to the IRISD (iris diameter), which is not enough precise. **You can get 0.001 precision if you click on the value.** In blender 4.49 a unit (Blender Unit – BU) can mean anything you want, 1cm, 1inch, 1mile, 0.5 lightyears, anything that is most appropriate. In blender 2.5 you can change units in the Scene settings from metric to imperial. *(I played modifying the value of grid spacing and divisions but there were no result in the precision of the 3D cursor. I don't know what else to do. Maybe the available precision is good enough as shown in the following picture 0.703-0.704).*



Description		FAPU Value
IRISD0 = 3.1.y - 3.3.y = 3.2.y - 3.4.y	Iris diameter (by definition it is equal to the distance between upper and lower eyelid) in a neutral face	IRISD = IRISD0 / 1024 = (0.716-0.703)/1024 BU = 0.013/1024 BU = 1.26953125e-05 BU
ES0 = 3.5.x - 3.6.x	Eye separation	ES = ES0 / 1024 = (0.035-(-0.035))/1024 BU = 0.07/1024 BU = 6.8359375e-05 BU
ENS0 = 3.5.y - 9.15.y	Eye - nose separation	ENS = ENS0 / 1024 = (0.711 – 0.665)/1024 BU
MNS0 = 9.15.y -	Mouth - nose separation	MNS = MNS0 / 1024

2.2.y		=(0.665-0.639)/1024
MW0 = 8.3.x - 8.4.x	Mouth width	MW = MW0 / 1024 = (0.026-(-0.026))/1024
AU	Angle Unit	10^{-5}rad





We are not implementing the full available list of FAPS. Here we describe the current support of MPEG-4 FAPS in EMBR. Information on the direction of the FAPs and the

FDPs there are directly applied to, are available in the Appendix III. To see how these FAPs affect other FDPs around them we will use the X-Face.

Visemes: high-level FAPS, implemented already in EMBR

Expressions: high-level FAPS, implemented already in EMBR

Angle FAPs: available through different controls in EMBR (mapping from EMBR to MPEG-4 FAPs is possible)

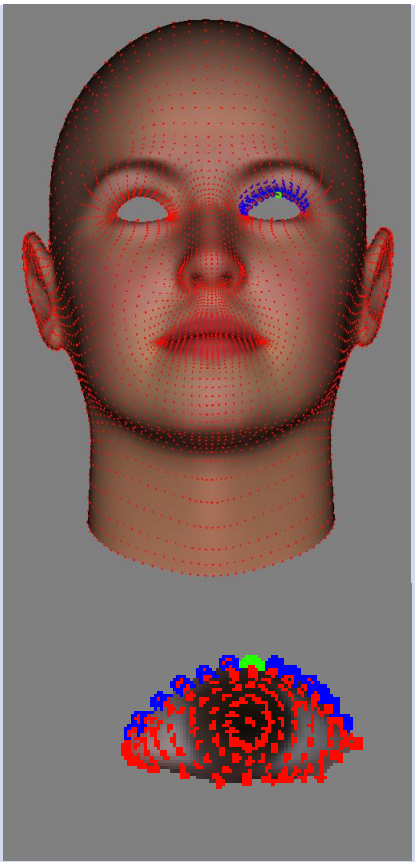
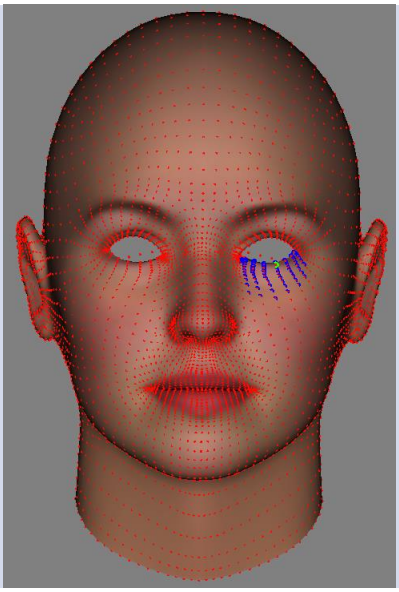
FAP Name	Mapping in EMBR
yaw_l_eyeball	
yaw_r_eyeball	
pitch_l_eyeball	
pitch_r_eyeball	
head_pitch	
head_yaw	
head_roll	

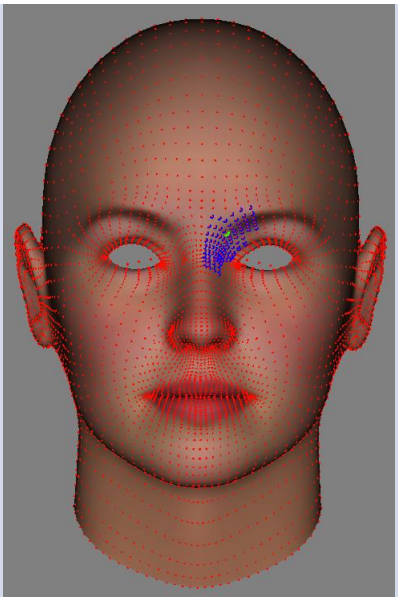
Displacement FAPs: our goal is to fully support this set. But in our development procedure we are going to give priorities and work based on them. To check with of the max is right I checked both the I-frame and P-frame min and max for some of the controls. It is quite obvious that the min max for us is defined by the values provided for **P-frame** (e.g. raise_l_i_eyebrow 0.016 vs. 0.04 (in which case it would reach the end of the face while going up or cover the eye while going down).

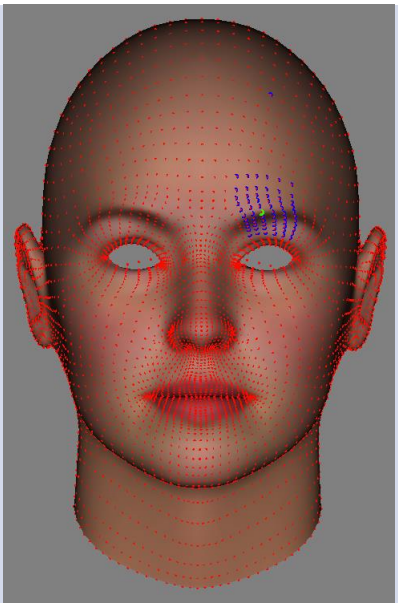
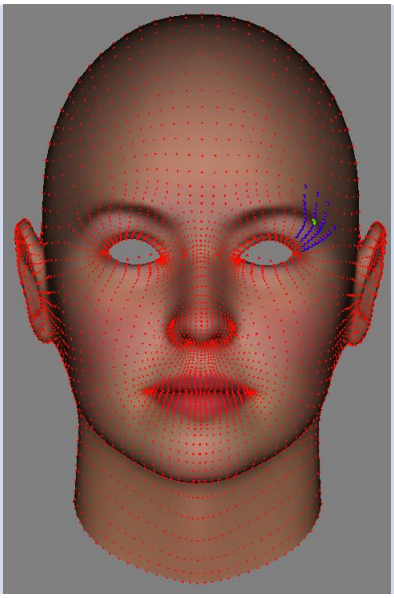
In the next table we provide the min and max displacement values for each of the FDPs, as well as the FAP point directly applied and the effect on the neighbors points of the face based on Alice. We assume symmetry and provide a sample of affects only for the left side. The provided images are cropped, which means that information such as influence function are still available if the image is outcropped.

TABLE 2: PRIORITY I FAPS

FAP Name	P-frame Min-Max (BU) FDPs applied	FDP Region Alice	Min/Max Result Alice

close_t_l_eyelid	+-600*IRISD0 +-0.0076 ---------- 3.1		IRISD = 16 +-1200*(16/1024) +-18.75
close_t_r_eyelid	+-600*IRISD0 +-0.0076 ---------- 3.2		
close_b_l_eyelid	+-240*IRISD0 +-0.003 ---------- 3.3		
close_b_r_eyelid	+-240*IRISD0		

	+-0.003 ---------- 3.4		
thrust_l_eyeball	+-180*ES0 +-0.0123 ----------	Not Available	This is not handled by EMBR eye gazing. His eyes cannot pop out.
thrust_r_eyeball	+-180*ES0 +-0.0123 ----------	Not Available	This is not handled by EMBR eye gazing. His eyes cannot pop out.
raise_l_i_eyebrow	+-360*ENS0 +-0.0161 ---------- 4.1		
raise_r_i_eyebrow	+-360*ENS0 +-0.0161 ---------- 4.2		

raise_l_m_eyebrow	+-360*ENS +-0.0161 ---------- 4.3	
raise_r_m_eyebrow	+-360*ENS +-0.0161 ---------- 4.4	
raise_l_o_eyebrow	+-360*ENS +-0.0161 ---------- 4.5	
raise_r_o_eyebrow	+-360*ENS +-0.0161 ---------- 4.6	

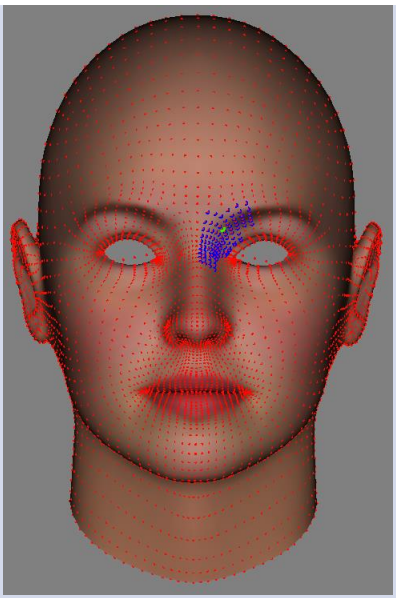
squeeze_l_eyebrow	**+-300*ES +-0.0205 ---------- 4.1**	
squeeze_r_eyebrow	**+-300*ES +-0.0205 ---------- 4.2**	
stretch_l_nose	**+-120*ENS +-0.0053 ---------- 9.1**	**Not defined for Alice**
stretch_r_nose	**+-120*ENS +-0.0053 ----------**	**Not defined for Alice**
raise_nose	**+-180*ENS 0.0080 ----------**	**Not defined for Alice**
bend_nose	**+-180*ENS 0.0080 ----------**	**Not defined for Alice**

TABLE 3: PRIORITY II FAPS

#	FAP name	FAP description	units	Uni-orBi dir	Pos motion	Grp	FDP subgrp num	Quant step size QP	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
1	viseme	Set of values determining the mixture of two visemes for this frame (e.g. pbm, fv, th)	na	na	na	1	na	1	viseme_blend: +63	viseme_blend: +-63
2	expression	A set of values determining the mixture of two facial expression	na	na	na	1	na	1	expression_intensity1, expression_intensity2: +63	expression_intensity1, expression_intensity2: +-63
3	open_jaw	Vertical jaw displacement (does not affect mouth opening)	MNS	U	down	2	1	4	+1080	+360
4	lower_t_midlip	Vertical top middle inner lip displacement	MNS	B	down	2	2	2	+-600	+-180
5	raise_b_midlip	Vertical bottom middle inner lip displacement	MNS	B	up	2	3	2	+-1860	+-600
6	stretch_l_cornerlip	Horizontal displacement of left inner lip corner	MW	B	left	2	4	2	+-600	+-180
7	stretch_r_cornerlip	Horizontal displacement of right inner lip corner	MW	B	right	2	5	2	+-600	+-180
8	lower_t_lip_lm	Vertical displacement of midpoint between left corner and middle of top inner lip	MNS	B	down	2	6	2	+-600	+-180

9	lower_t_lip_rm	Vertical displacement of midpoint between right corner and middle of top inner lip	MNS	B	down	2	7	2	+-600	+-180
10	raise_b_lip_lm	Vertical displacement of midpoint between left corner and middle of bottom inner lip	MNS	B	up	2	8	2	+-1860	+-600
11	raise_b_lip_rm	Vertical displacement of midpoint between right corner and middle of bottom inner lip	MNS	B	up	2	9	2	+-1860	+-600
12	raise_l_cornerlip	Vertical displacement of left inner lip corner	MNS	B	up	2	4	2	+-600	+-180
13	raise_r_cornerlip	Vertical displacement of right inner lip corner	MNS	B	up	2	5	2	+-600	+-180
14	thrust_jaw	Depth displacement of jaw	MNS	U	forward	2	1	1	+600	+180
15	shift_jaw	Side to side displacement of jaw	MW	B	right	2	1	1	+-1080	+-360
16	push_b_lip	Depth displacement of bottom middle lip	MNS	B	forward	2	3	1	+-1080	+-360
17	push_t_lip	Depth displacement of top middle lip	MNS	B	forward	2	2	1	+-1080	+-360
18	depress_chin	Upward and compressing movement of the chin (like in sadness)	MNS	B	up	2	10	1	+-420	+-180
19	close_t_l_eyelid	Vertical displacement of top left eyelid	IRISD	B	down	3	1	1	+-1080	+-600
20	close_t_r_eyelid	Vertical	IRIS	B	down	3	2	1	+-1080	+-600

FAP Name	Blender Min-Max	Notes
lower_t_midlip	+-180*MNS -- 2.2 (down)	
raise_b_midlip	+-600*MNS -- 2.3 (up)	
stretch_l_cornerlip	+-180*MW -- 2.4 (left)	
stretch_r_cornerlip	+-180*MW -- 2.5 (right)	
lower_t_lip_lm	+-180*MNS -- 2.6 (down)	
lower_t_lip_rm	+-180*MNS -- 2.7 (down)	
raise_b_lip_lm	+-600*MNS -- 2.8 (up)	
raise_b_lip_rm	+-600*MNS -- 2.9 (up)	
raise_l_cornerlip	+-180*MNS -- 2.4 (up)	
raise_r_cornerlip	+-180*MNS --	

	2.5 (up)
push_b_lip	+-360*MNS
	--
	2.3 (forward)
push_t_lip	+-360*MNS
	--
	2.2 (forward)
puff_l_cheek	
puff_r_cheek	
lift_l_cheek	
lift_r_cheek	
lower_t_midlip _o	+-180*MNS
raise_b_midlip_o	+-600*MNS
stretch_l_cornerlip _o	+-180*MW
stretch_r_cornerlip _o	+-180*MW
lower_t_lip_lm _o	+-180*MNS
lower_t_lip_rm _o	+-180*MNS
raise_b_lip_lm_o	+-600*MNS
raise_b_lip_rm_o	+-600*MNS
raise_l_cornerlip_o	+-180*MNS
raise_r_cornerlip _o	+-180*MNS

TABLE 4: PRIORITY III FAPS

FAP Name	Blender Min-Max	Notes
open_jaw		
thrust_jaw		
shift_jaw		
shift_tongue_tip		
raise_tongue_tip		
thrust_tongue_tip		
raise_tongue		
raise_l_ear		
raise_r_ear		

pull_l_ear	
pull_r_ear	

TABLE 5: NOT EXACLTY DISPLACEMENT FAPS

FAP Name	Blender Min-Max	Notes
depress_chin		
dilate_l_pupil		
dilate_r_pupil		
tongue_roll		

B. Use X-Face for MPEG-4 FAPs

In the above tables we provide photos on the FDP region that a FAP affects and how about should a min and a max position look. For the above we used XfaceEd[3][4] and their built-in face Alice. To be able to preview the min and max effects of a FAP we had to define those since the default min and max bounds for every FAP are not provided in the system.

We are given that the FAPU for this 3D model as the following

```
<!--FAPU (Facial Animation Parameter Units)-->
<fapu ENS0="51" ES0="69" IRISD0="16" MNS0="30" MW0="50"/>
```

[3] http://xface.fbk.eu/index.htm
[4] http://dl.acm.org/citation.cfm?id=1088500

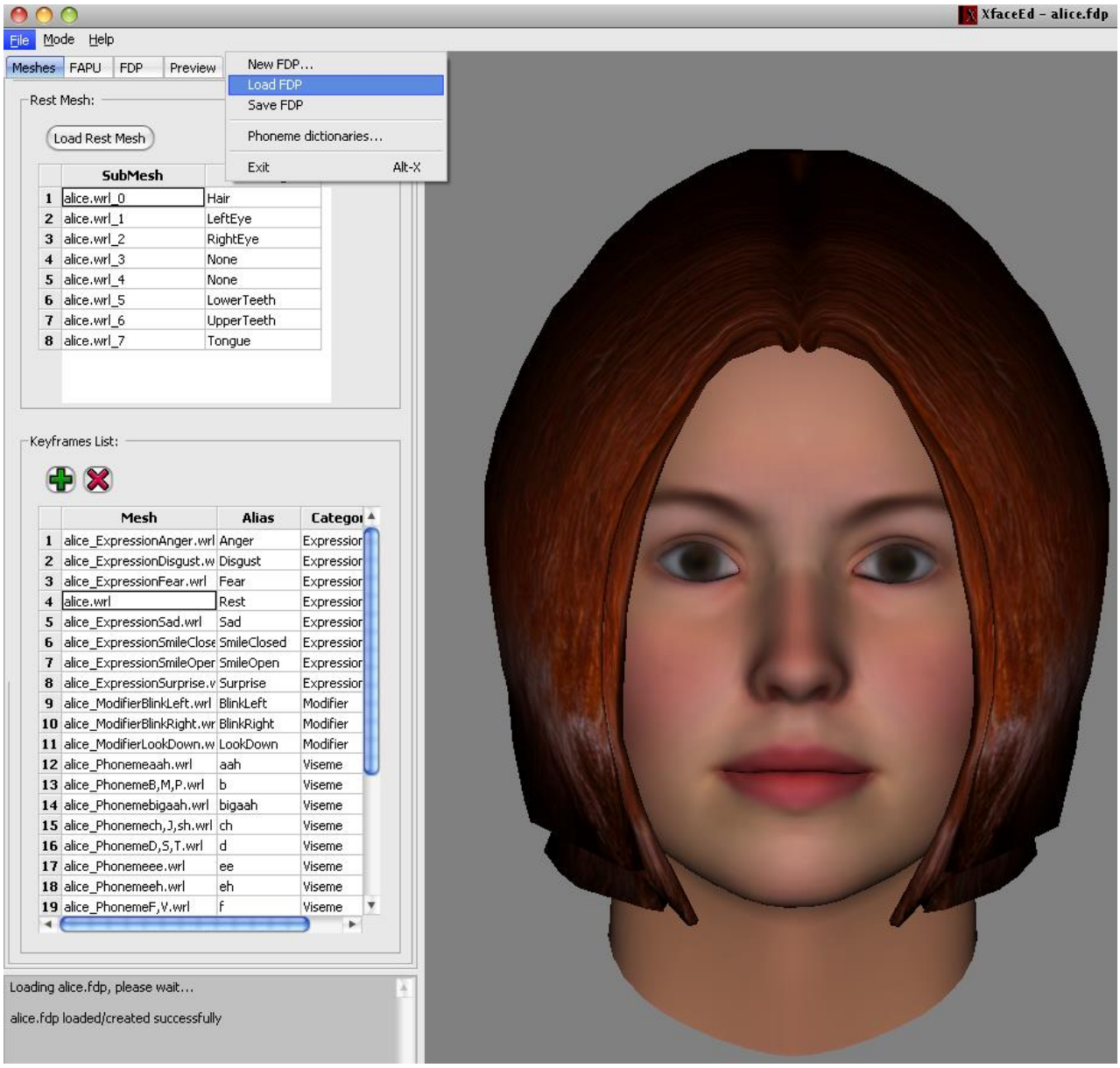


FIGURE V-1: XFACE EDITOR 1.0

Why do we need X-Face Ed?

Well, step 3 (in building a proprietary MPEG-4 face model) requires the following:

After the motion of the feature points is defined for each FAP, we define how the motion of a feature point affects its neighboring vertices. This mapping of feature point motion onto vertex motion can be done using lookup tables such as FAT (animation rules)[5],

[5]J. Ostermann, E. Haratsch, “An animation definition interface: rapid design of MPEG-4 compliant animated faces and bodies”, International Workshop on Synthetic – Natural Hybrid Coding and Three Dimensional Imaging, Rhodes, Greece, September, 1997, pp. 216–219.

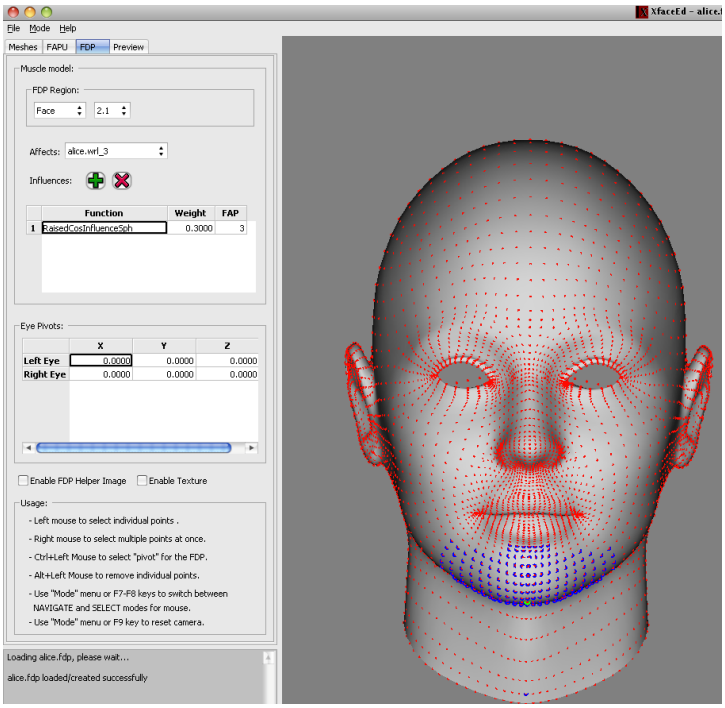
muscle-based deformation [8, 10, 11], distance transforms [18] or cloning from existing models [19, 20].

Since our FAPs are implemented by an animator (artist) FAT rules, muscle-based deformation, distance transforms and automated cloning from existing models, can't be directly applicable by a human. For that reason we try to approach the Expression Cloning methodology but this time a human does the cloning part; so maybe we should call it *Expressions Replicate*. An animator provided a sample facial expression in 3D (or better a FAP application region, min-max values, and their result in a sample model) is able to apply a similar affect in a new 3D model. The sample model FAPs are driven by one of the previously mentioned methodologies.

How do we use X-Face Ed?

1. File> Load FDP> alice.fdp

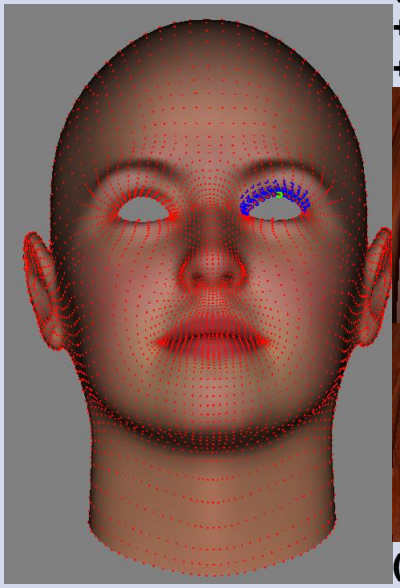
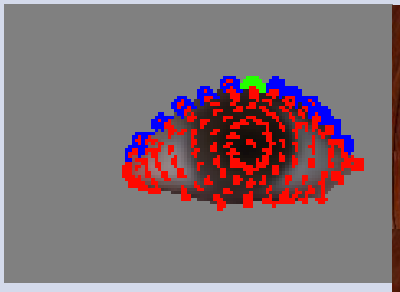
2. To see the area affected by a FAP you go to the *FDP Tab* and chose to see the part of the face including the FDP that the FAP is applied to. (There might be more than one part of the face e.g. 3.1 affect is included in both the face and left eye). The corresponding FDP is denoted with green and the affected area with blue.



3. To preview the min and max on a certain FAP go to the *Preview Tab* and define first FAP we want to see and then set the min and max values for those. I don't know why but for this model there are no predefined min and max on any of the FAPs. It requires us to calculate those. It could be possible given that we know the FAPU already. Using

the same methodology as for Ben we get something else. Still have to find the math behind it.

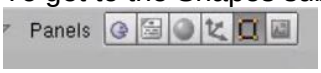
Example

FAP Name	Min-Max (BU) FDPs applied	FDP Region in Alice	Min/Max in Alice
close_t_l_eyelid	+-600*IRISD0 +-0.0076 --- 3.1	 	IRISD = 16 (P-frame) +-600*(16/1024) +-9.3  (I-frame) +-1200*(16/1024) +-19 

C. Create MPEG-4 FAPs In Blender

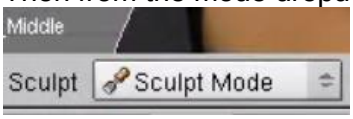
Add morphs to blender file with same name. You must use Blender version 2.49. Use chicken exporter to produce ben.egg and ben.bam. Some of the first steps required for creating new morph targets for the face are the following (screenshots are available at the Appendix):

5. Open Ben2.49.blend file
6. To **zoom in/out** use two fingers in touchpad (back and forth)
7. To **pan** (move) the character in the window press Shift+Option+LeftClick
8. To **orbit** (rotate) the character press Option and move with LeftClick.
9. Editing and Pose: to **select** a vertex or a bone do it with RightClick (double tap); to select more than one press Shift + RightClick.
10. Editing and Pose: to **move** a vertex or a bone you should click in one of the axes shown after the selection and then just move the mouse and LeftClick when you are at the wanted position. Clicking in one of the axes will lock any movement in the other two.
11. Pose: to **move only a part** of the skeleton without affecting the rest you should first disconnect it and then connect it again. How? Go to
12. To get to the Shapes sub-panel choose Editing (F9) from the panels.

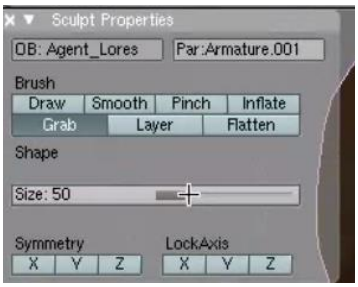


13. Then click on Shapes tab

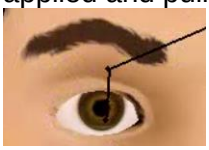
14. On the Shapes sub-panel click on Add Shape Key, name it (make sure that instead of dash you use underscore) and then make sure you pick the corresponding one from the VGroup. Now, we can try by choosing min -1 and Max 1 if we want to implement exactly as they are defined for some of the MPEG-4 FAPS. If I remember correctly there is no restriction for this in the EMBRscript but Alexis might need to do some modifications for the Behavior Builder GUI. I'll have to see the video to be sure about this. Otherwise we will have to create two shape keys (-1 to 0, and 0 to 1).
15. Then from the mode dropdown pick Sculpt Mode.



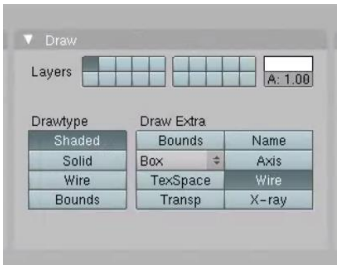
16. To display the Sculpt Properties press N key.



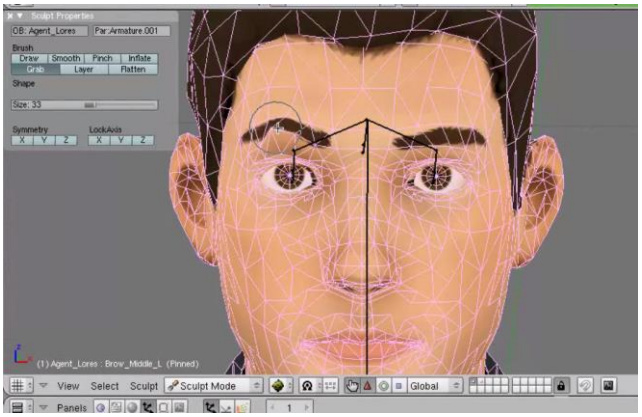
17. Make sure you pick Grab for the Brush and then define its size.
18. Apply the brush to the point where displacement of a facial component is required. (If you get the message *Error: "Cannot sculpt on unlocked shape key"* just click that little pin-button in the Shapes subpanel. 

. However, do not forget to lock it again before exporting.)
E.g. for the middle point of the eyebrow going all the way up the brush should be applied and pull to the direction needed.

It is important to note that sometimes to get a better-looking result a bigger size brush should be used. But given that the brush would affect all the mesh underneath and not just the face component we are interested in that would modify components of the face controlled by other FAPs also.
Next step: After defining the min and the max optical result check if there is a conflict in any of the cases. Then discuss with Matt.

19. Next step: define after a detailed study of the MPEG-4 FAPS how would the corresponding min and max look like according to static distances among some of the face components.
20. Now at this point or even earlier at step 10, you might need to have a better way to see the how much change was introduced. A likewise tool with the ruler in this case is the option of wires above the mesh. To achieve it pick Object from Panels 

and then at the Draw subpanel activate the boxes Shaded and Wire.


The result is going to look like the following:



21. In case you want to do editing in the Edit Mode by playing with the vertices then locking the displacement in one of the axis is useful. With right mouse click you can activate the vertices you are interested in. At that point you can see the axis system. With left mouse click you can pick one of the axes, by locking the displacement on it. More on axis locking: http://wiki.blender.org/index.php/Doc:2.4/Manual/3D_interaction/Transform_Control/Axis_Locking.

D. Export MPEG-4 FAPs From Blender

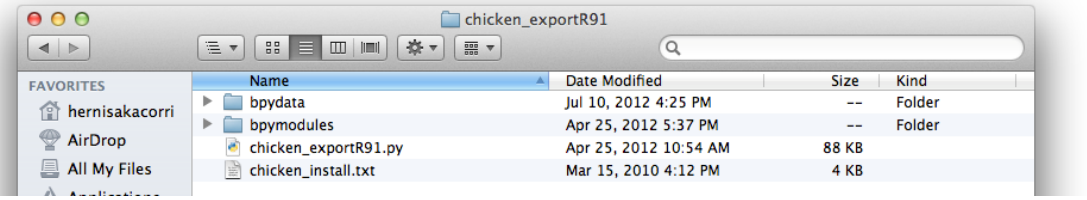
This section describe the procedure to export Ben with MPEG-4 Facial controls from Blender to Panda3D (ChickenExporter) on a MAC computer.

What to be already installed:

- Blender 2.49b
- Panda3D
- Python 2.6 or similar

What to download:

- chicken_exportR91
  It includes:



How to install it:

- blender.app is a folder in there hidden is the following path (use command line): blender.app/Contents/MacOS/.blender/scripts

- Copy the chicken _exportR91.py in here.
- Copy whatever was under bpydata in the chicken exporter in to the corresponding folder bpydata under blender.app/Contents/MacOS/.blender/scripts.
- Copy whatever was under bpymodules in the chicken exporter in to the corresponding folder bpymodules under blender.app/Contents/MacOS/.blender/scripts.
- Make sure the first time you run it that in the setting that the path /Developer/Tools/Panda3D is given. Under there you can see the egg2bam.

How to export:

- Make sure you saved the .blend file somewhere where there is no space in the name.
- It is really important that the folder's (EMBRSnapshot) tree structure and names on subfolders and files to remain the same.
- Set the "Export File: " path with the right path. E.g. if the EMBRSnapshot2 were in the desktop then the path would be /Users/hernisakacorri/Desktop/EMBRSnapshot2/data/BlenderSources/Ben.egg
  Note: It is really important that the egg is exported in the same folder were the .blend file is saved otherwise the textures won't work correctly.
- Make sure in the "Call: " line that the egg2bam and the egg-optchar are enabled.
- Set the following parameters to the "Call: "
  -keepall –inplace –pd "/Users/hernisakacorri/Desktop/EMBRSnapshot2/data/BlenderSources" –ps rel
  Note: the path given here is again under the assumption that the EMBRSnapshot2 were in the desktop.
- While exporting an error seems to occur (reported by the egg-optchar utility) as shown below, however everything seem to work just fine.
  8/7/12 1:10:47.610 AM [0x0-0xc30c3].org.blenderfoundation.blender: Assertion failed: !filename.empty() at line 956 of
  8/7/12 1:10:47.610 AM [0x0-0xc30c3].org.blenderfoundation.blender: panda/src/express/virtualFileSystem.cxx
  The same error seems to be at Alexis also. It does not affect the result.
- Finally after the Ben.egg and Ben.bam have been exported and saved, copy the Ben.bam file under the /Users/hernisakacorri/Desktop/EMBRSnapshot2/data/models/ben folder.

E. Include MPEG-4 FAPs In Behavior Builder

In NetBeans open the current project EMBOTS. To include a new face target you have to modify two files:

- EMBRFacialExpression.java
  e.g. include a new line in public enum EMBRFacialExpression {

such as raise_l_i_eyebrow("raise-l-i-eyebrow") (here we use underscore instead of dash because it wouldn't work with dash. But, everywhere else where it is treated as a string it should be written with -).

- EMBRMorphKey.java
  e.g. include a new line in public enum EMBRMorphKey{
  such as raise_l_i_eyebrow("raise-l-i-eyebrow"). In the GUI it would appear as raise_l_i_eyebrow.
- Build the project and copy the EMBOTS.jar at the Desktop where you have the running script.
  hernisakacorri>trunk>dist>EMBOTS.jar
- Note: The position of the control in these lists actually defines the position in the drop-down list in the GUI.
- TODO: Put the MPEG-4 controls first in the drop-down list.

F. Include MPEG-4 FAPs In Panda3D

Add lines to ben.py for new face morphs. (First, there is a list for the basic atomic deformations, then there is a list of the AUs that are composed from the atomic ones.)

- In this case you just have to modify the ben.py code in two points, as shown by Alexis. Further investigation on the values is required.
  e.g. add the following line in maxMorphTargets={
  'raise-l-i-eyebrow':self.agent.controlJoint(None, 'modelRoot', 'raise-l-i-eyebrow')
  add the following line in self.targets = {#
  'raise-l-i-eyebrow':{maxMorphTargets['raise-l-i-eyebrow']:1.0}
- Make sure the updated Ben.bam is included in the models/ben subfolder as mentioned above.
- Make sure you understand the correspondence of the 1.0 values (percentages?) and the relation to the control values in the GUI.

G. Aligning Blender and Panda3D Cameras

Currently EMBR comes with one ambient camera and one spotlight aiming at Ben's face. This does not allow for the wrinkles to show. To fix it we will use one ambient light and 3 spotlights arranged as shown in Fig.V-2.

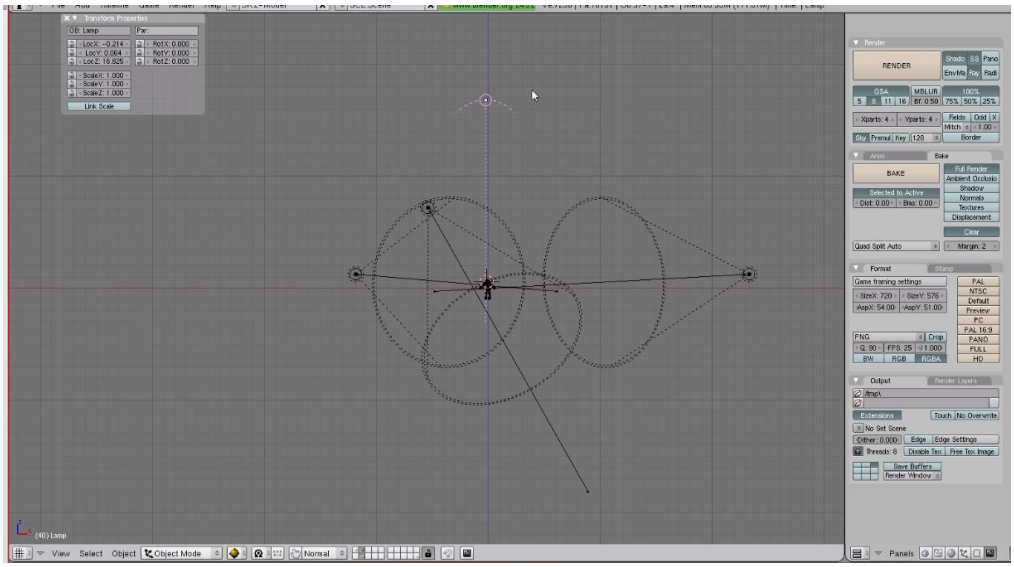


FIGURE V-2: BEN AND THE 4 LIGHTS SETTING IN BLENDER

We have the coordinates for the lights in Blender and now we want to figure out the code for Panda3D in embr.py. It seems like Blender and Panda3D have slightly different coordinate systems “It's the difference between y-up and z-up. You can multiply the model's transformation matrix with Mat4.yToZUpMat() or zToYUpMat() to convert back and forth.”[6].

See http://www.panda3d.org/manual/index.php/Lighting for a tutorial on Panda3D lighting and python.

[6] http://www.panda3d.org/forums/viewtopic.php?f=2&t=16125

PART VI. CREATE NEW HANDSHAPES IN BLENDER

A. Create new handshapes in Blender

If you don't remember how to do this make sure you have checked first the screenshots in the Appendix.

- Open the new Ben2.49.blend provided by Alexis make sure you are in the pose mode.
- In the NLA Editor make sure that the Armature is active.
- Add Blank Action Strip from the Strip menu in the NLA. (You can scale it to be the same size as the above by pressing the s key and then using the mouse and you can also moving so that it starts at the end of the previous one using the g key and then the mouse.)
- Go to Action Editor
- Name the new action (e.g. hands_ASL_6)
- Go back to the NLA Editor
- The green line defines which of the actions pose is in the screen. Put the green line above the action that you would like to be similar to. Copy the pose on the hands. You can either choose all the bones on both of the hands or use the box selection using the b key and then the mouse.
- To copy the pose, press the first out of the three images "Copies the current pose to the buffer". If you cannot see it then just enlarge the character's window.
- Then move the green line above the handshape (action) you are creating. Select the same bones and this time choose the second image that says "Pastes the pose from the buffer".
- Then press the i key and choose Rot.
- Now, since moving a bone is actually going to move the whole character it seems better to deactivate the IK.
- Pick the bone you want to move (right click or shift right click for more than one) and from the axes that appear choose the one you want your bone to rotate on.
- To understand better what you are doing make sure you have orbited the character such that the axis you want to rotate on is perpendicular to the screen (illustrated as a dot instead of an arrow). Left click on the dot and then use the mouse.
- Modify only the right hand and when you are happy with the result copy the whole hand and to paste it to the selected left hand bones use the third image named "Pastes the mirrored pose from the buffer".
- Don't forget to press the i key and then choose Rot again. Otherwise the pose won't be registered in the action.
- To export go in Object Mode and make sure the Agent_Lores is now the one active in the NLA Editor and that the green line is standing above your action.
- As before make sure you are using the right paths.
- Press "Add Animation"
- Put the number that the animation starts at the NLA Editor as the start and the end and a name for the pose (handshape).
- Make sure "Single File" is not activated.
- Make sure "Animation Only" is activated.

- Export (a file e.g. named Ben-hands_ASL_6.bam and .egg is saved)
- Copy the .bam under models/ben/animations
- Yes it seems creating a moving handshape is possible. In one action we make more than one pose using the green line, and giving a different end point while exporting. More need to be investigated in this case.

B. Include new handshapes in Panda3D

- Add lines to ben.py for new handshapes.

- Add lines to CharacterConfiguration.xml file for new handshapes.

- Create (or duplicate) BVH files for new handshapes and add to the data folder. Use a free BVH editor to do this.

Alexis notes:

As you will notice, two files need to be updated:

1: embr.py should be updated so that the pada .bam assets files are loaded with the charecter

(l. 54) replace:

```
self.ben.setActor( self.dataPath + 'models/ben/ben.bam',{},{})
```

with

```
self.ben.setActor( self.dataPath + 'models/ben/ben.bam',{"hands_ASL_6":self.dataPath + "models/ben/animations/Ben-ASL_6.bam"},{})
```

--> you actually register a dictionary entry into the first set of curly brackets: the motion described in the 'Ben-ASL_6.bam' file will be registered as a motion assets under the name 'hands_ASL_6'

2: in Ben.py, inform the embr runtime that a panda animation should be parsed, translated to the EMBR conventions (bvh like) and register under the keyword of your choice (e.g 'hands_ASL_6')

(l. 290) add:

```
self.addAnimation(self.name,'hands_ASL_6')
```

save the two files and the new keypose can be recalled in the EMBRScript language (also cf. enlosed picture):

BEGIN POSE_TARGET

FEEDBACK_ID:feedback_start

BODY_GROUP:rhand

POSE_KEY:hands_ASL_6

C. Include new handshapes in Behavior Builder

For convenience, you might want to update the handshape static declaration in the JAVA sourcecode of the behavior builder. In this case you should edit the file named EMBRHandshape.java by adding the handshapes in the enumerator EMBRHandshape where the first is the name of the handshape as it appears in the interface and the second the name of the handshape as it will appear in the generated EMBRscript, e.g.:

ASL_8("hands_ASL_8"),

The file to be changed in e.g. netbeans is named EMRHanshape.java and is located:

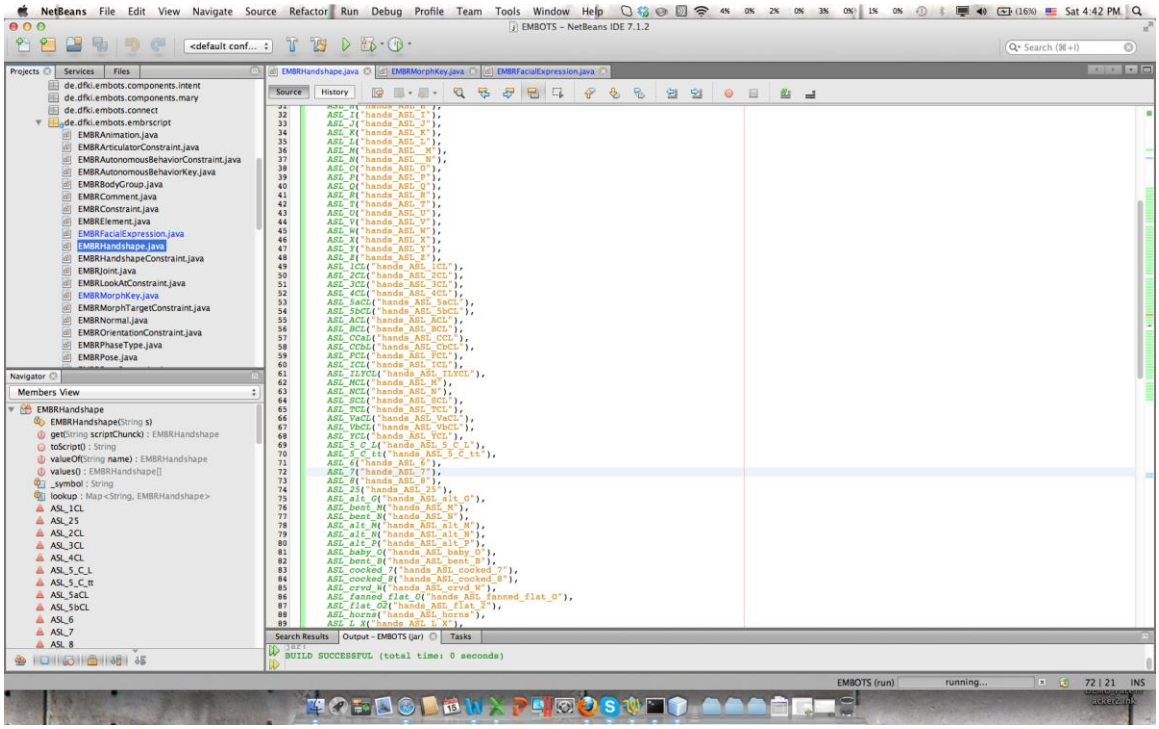


FIG. 1: THE EMRHANSHAPE.JAVA IN NETBEANS

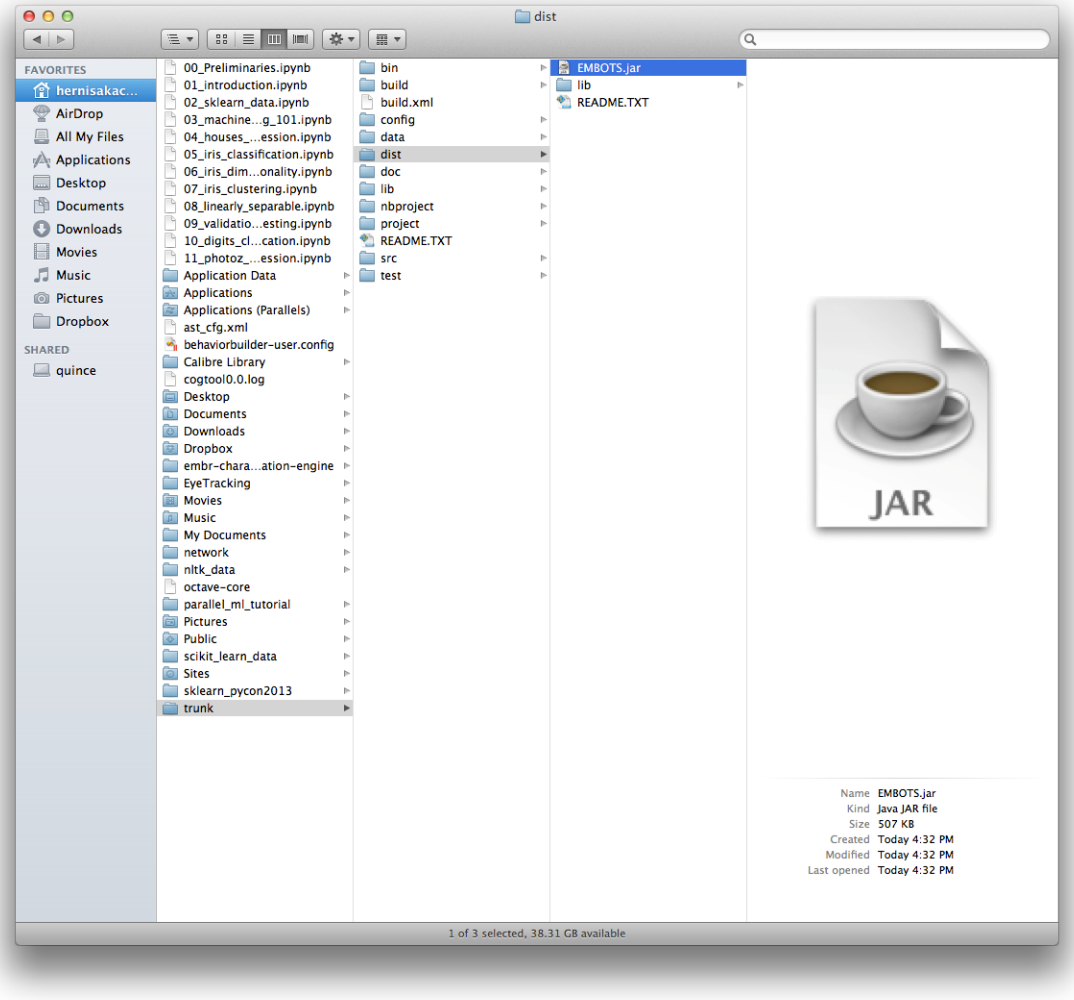


FIG. 2: AFTER COMPILING IN NETBEANS THE NEW JAR FILE IS LOCATED HERE.

D. What is done up to now?

We mapped the dictionary handshapes to the available EMBR handshapes and to the VCOM3D handshapes. The one missing were created by Andy and inserted following the dictionary naming convention. However this was not the case for the already existing ones? Should we do this? (Should I change them?) To preserve back compatibility I choose not to. This way we can still use the signs created the previous summer by the students. However, I could change the interface if that would simplify the process of creating new signs.

There was a conflict. hands_ASL_M and hands_ASL_N (.bvh) that are a copy of hands_DGS_M and hands_DGS_N actually are mapped to the hands_ASL_bent_M and hands_ASL_bent_N but for back compatibility reasons their name in the script is maintained. So the EMBRscript names for the new ASL handshapes are hands_ASL__M and hands_ASL__N.

Note: There is a restriction in the length of the handshape name inside the interface. E.g. ASL_x_over_thumb was too long. The consequence was that the right handshape dropdown menu will disappear. This can easily be fixed by changing the name e.g. to ASL_x_thumb.

TABLE 6: HANDSHAPE MAPPING DONE BY NISA AND WESLEY

Dictionary Handshape	EMBR Handshape	Similar but not identical embr hanshape	Similar embr handshape	Temporal Mapping for VCOM to EMBR converter	Dictionary Example	VCom3D	Similar VCom3D but not identical handshape
1	hands_index	hands_ASL_1CL			ALONE	**n-1+**	
3	hands_ASL_3CL				(3)ACCIDENT	**l+12v+**	
4	hands_ASL_4CL				(cocked-8)SHOOT-ARROW	**n-1234+**	
5	hands_open-spread	hands_DGS_SCH	hands_ASL_5aCL		ABANDON	**L+1234v+**	
6				**hands_DGS_W**	FEW		**o~c^~0**
7					FRESHMAN		
8					ns-nat-AUSTRALIA		
10	hands_ASL_ACL				LITTLE-BIT	**L+**	
25				**hands_ASL_ILYCL**	BALD		**L14v+**
5-C	hands_claw				OLD	**u^1234v~0**	
5-C-L				**hands_claw**	ABORTION_2		**u^1234v~0**
5-C-tt				**hands_claw**	CHANNEL		**u^1234v~0**
A	hands_DGS_A				A	**01+L~L~L~L~**	
B	hands_DGS_B				BACHELOR	**b^1234+**	
B-L	hands-open-straight				ABORTION	**01+H+H+H+H+**	
B-xd	hands_ASL_BCL				ENTER	**b^1234+**	
C	hands_DGS_C	hands_ASL_CCL			OLD+	**11^M~M~M~M~**	

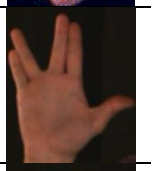
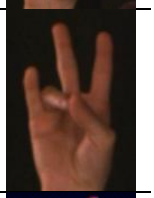
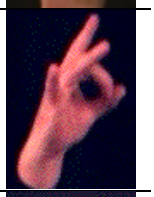
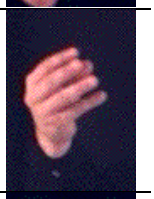
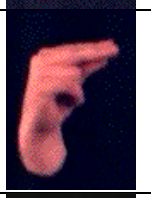
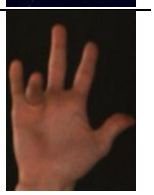
D	hands_DGS_D			SINGLE	**o^c678^**	
E	hands_DGS_E			ELEMENTARY	**b-c1234~**	
F/9	hands_DGS_F	hands_ASL_FCL		GUITAR	**o-c9v~0B**	
G/Q	hands_DGS_Q	hands_ASL_G		BACKGROUND	**o^1^**	
Horns			**hands_ASL_ILYCL**	ALCOHOL		**L14v+**
I	hands_DGS_I	hands_ASL_ICL		ART/DESIGN	**n-4+**	
I-L-Y	hands_ASL_ILYCL			AIRPLANE	**L14v+**	
L	hands_DGS_L			TWENTY	**L+1+**	
L-X			**hands_ASL_CbCL**	ARTICLE		**o~1~0**
M			hands_DGS_M	M		**b-123^**
N			hands_DGS_N	GOAT		**n-12^**
O	hands_DGS_O			ADVISE/INFLUENCE+	**o^c1^**	
O/2-Horns			**hands_ASL_ILYCL**	ns-nat-BOLIVIA		**L14v+**
Other				GIRL-SCOUT		
P/K	hands_DGS_K			PERMIT	**n-12K^**	
R	hands_DGS_R			REPORT	**n-12x+**	
R-L			**hands_DGS_R**	REVIEW		**n-12x+**
RIxd	hands_open-relaxed			THAN	**01+M~M~M~M~V01C12C23C34**	
S	hands_ASL_SCL			ABANDON	**n-1-**	
T	hands_ASL_T			TOILET	**u~uc1-**	
U-L			**hands_DGS_H**	BACON		**n-12+**
U/H	hands_DGS_H	hands_DGS_U		HEREDITY	**n-12+**	
V/2	hands_ASL_2CL	hands_ASL_VaCL		ANALYZE	**n-12v+**	

Vulcan					(Vulcan)FAX		
W	hands-DGS_W				WATER+OCEAN	**o~c^~0**	
X	hands_DGS_X				APPLE	**n-1~**	
X-over-thumb				**hands_ASL_T**	LITTLE-BIT		**u~uc1-**
Y	hands_DGS_J	hands_DGS_Y	hands_ASL_YCL		(Y)BATHING-SUIT	**L+4+**	
alt-G				**hands_ASL_G**	(G)GOVERNMENT		**11+L^L-L-L-**
alt-M				hands_DGS_M	MATH		**b-123^**
alt-N				hands_DGS_N	MIDNIGHT		**n-12^**
alt-P				**hands_DGS_K**	UP-TO-NOW		**n-12K^**
baby-O				**hands_ring**	TWENTY		**o~c1~0**
bent-1	hands_DGS_G				REGULAR	**n-1+**	
bent-B				**hands_ASL_4CL**	EXCEED		**n-1234+**
bent-B-L				**hands_ASL_4CL**	ABOVE		**n-1234+**
bent-B-xd				**hands_ASL_4CL**	BLANKET		**n-1234+**
bent-Horns				**hands_ASL_ILYCL**	KID		**L14v+**
bent-I-L-Y				**hands_ASL_ILYCL**	AIRPLANE		**L14v+**
bent-M	hands_DGS_M				MOSQUE	**b-123^**	
bent-N	hands_DGS_N				NEUTRAL	**n-12^**	
bent-U				**hands_DGS_N**	ns-nat-BEIJING		**n-12^**
bent-U-L				**hands_DGS_H**	BACON		**n-12+**
bent-V				**hands_DGS_N**	BOTH		**n-12^**
cocked-7					VOLLEYBALL		
cocked-8					(cocked-8)SHOOT-ARROW		
cocked-F				**hands_DGS_F**	BUTTON		**o-c9v~0B**
cocked-S				**hands_ASL_T**	THROW-AWAY		**u~uc1-**

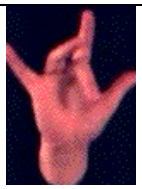
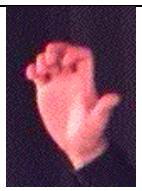
cocked-U		**hands_ASL_SCL**	(U)BOWTIE		**n-1-**
crvd-3		**hands_ASL_3CL**	BASKETBALL		**l+12v+**
crvd-5	hands_ASL_5bCL		ABORTION	**u^1234~0**	
crvd-B		**hands_DGS_C**	ENTER		**11^M~M~M~M~**
crvd-L		**hands_ASL_CbCL**	TWENTY		**o~1~0**
crvd-U		**hands_ASL_5bCL**	BACON_2		**u^1234~0**
crvd-V	hands_ASL_vbCL		ANALYZE	**12^H^H^L-L-**	
crvd-W			WEATHER		
crvd-flat-B		**hands_open-relaxed**	APPOINTMENT		**01+M~M~M~M~V01C12C23C34**
crvd-sprd-B	hands_open-relaxed		TOTAL	**01+M~M~M~M~V01C12C23C34**	
fanned-flat-O			EYELASHES		
flat-B		hands_ASL_BCL	BELOW		**b^1234+**
flat-G		**hands_ASL_G**	TWENTY		**11+L^L-L-L-**
flat-O	hands_purse		ACCEPT	**o^c12^**	
flat-O/2			DUCK		
full-M	hands_DGS_M		GUITAR	**b-123^**	
loose-E		**hands_DGS_E**	ELEMENTARY		**b-c1234~**
open-7			FRESHMAN		
open-8			BELONG		
open-F			CHAIN/OLYMPICS		
sml-C/3	hands_ASL_cbCL		PIN	**o~1~0**	
tight-C		**hands_purse**	AND		**o^c12^**

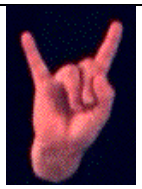
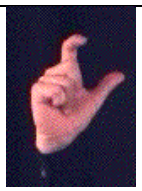
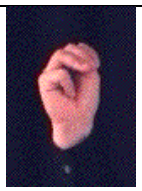
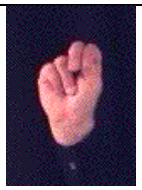
New Hanshapes created by Andy.

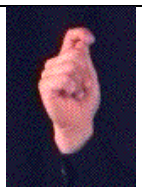
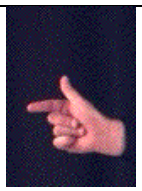
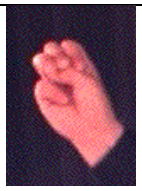
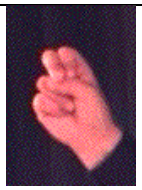
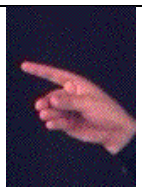
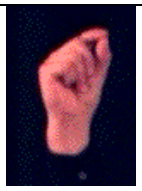
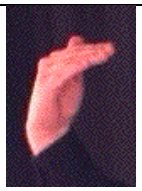
hands_ASL_[NAME].bam by substituting [NAME] with the name of the handshape.

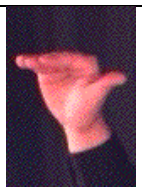
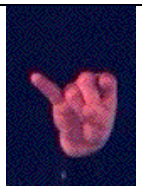
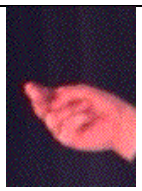
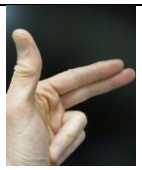
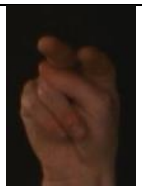
'7'	
'8'	
'Vulcan'	
'cocked-7'	
'cocked-8'	
'fanned-flat-O'	
'flat-O/2'	
'open-7'	

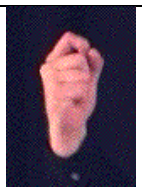
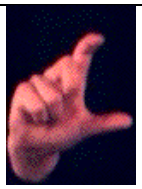
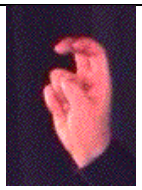
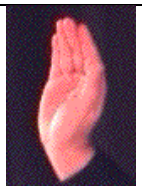
'open-8'	
'open-F'	
'crvd-W'	

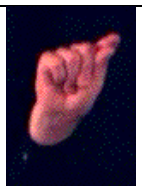
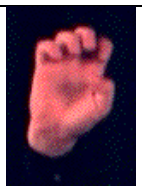
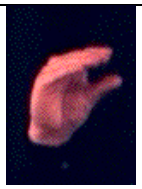
'6'	
'25'	
'5-C-L'	
'5-C-tt'	

'Horns'	
'L-X'	
'M'	
'N'	
'O/2-Horns'	
'R-L'	
'U-L'	

'X-over-thumb'	
'alt-G'	
'alt-M'	
'alt-N'	
'alt-P'	
'baby-O'	
'bent-B'	

'bent-B-L'	
'bent-B-xd'	
'bent-Horns'	
'bent-I-L-Y'	
'bent-U'	
'bent-U-L'	
'bent-V'	
'cocked-F'	

'cocked-S'	
'cocked-U'	
'crvd-3'	
'crvd-B'	
'crvd-L'	
'crvd-U'	
'crvd-flat-B'	
'flat-B'	

'flat-G'	
'loose-E'	
'tight-C'	

bla

PART VII. BUILD AN EMBR DICTIONARY

Refine the converted scripts

- Students will help with this by opening the converted gestures in combination C and try to improve them.
-

PART VIII. CREATE STIMULI IN EMBR

A. Create sentences

- Students will help with this by combining the already existed gestures in one or more sentences/ stories.

B. Create facial expression for each of the sentences

Given a sentence or a story try to define the corresponding facial expressions such that they are synchronized in time with the given sentence.

C. Create the final stimuli

Combine facial expression and sentences in one script and import the script in the Combination D. Use a screen capture tool, e.g. ishowu (Alexis uses this) for creating a video of the resulting animation.

How to define the size of Ben and initial pose?

Step 1: Change Panda3D Window size

Panda3D configuration file 'config.prc' is found under /Developer/Panda3D/etc. There you can modify rendering machine, window size or fullscreen, disable framrate meter etc. Adjust them accordingly to the size of the stimuli.

win-origin 50 50

win-size 800 600

show-frame-rate-meter #f (enable/ disable the framerate meter)

Step 2: Change Ben zooming, location, and pose within the Panda3D window.

PART IX. MORE TO BE INVESTIGATED IN EMBR

Moving handshapes

Exporting Ben’s head to .wrl in such a way that it can be opened by Xface.

PART X. VISAGE TRACKING

This section will include all the necessary information about tracking head pose and facial features in Visage SDK. Some of the information is copied from the Visage manuals.

A. Video requirements

Format. It is recommended that the video be DivX encoded AVI file. Use the best possible quality when encoding. You can use the XviD codec, downloadable from http://www.xvidmovies.com/codec/. A good free tool for capturing and editing videos is VirtualDub (http://www.virtualdub.org).

Resolution. The recommended resolutions are 640x480 or 800x600. Other similar or higher resolutions work well too. 320x240 generally works, though results may be inferior. Higher resolutions make tracking slower and improve quality to some extent. However, to realize the full potential of quality improvement with high resolutions it is necessary to tweak the parameters of the tracker, see the VisageTracker Configuration Manual.

Quality. Use reasonable off-the-shelf camera and lighting conditions.

Positioning. There is no strict requirement, but the face should be prominent in the image, taking perhaps one third of the image. It is not essential for the face to be in the center. See Figure 5 for some examples.

Neutral face in first frame. It is not a strict requirement, but it is best if the first frame of the video contains the face in a neutral position: facing forward, looking straight, eyes open, mouth closed, no smiling or other expressions. If not in the very first frame, neutral face should be available early in the video. The tracker will automatically search for the first frame with acceptable head pose.

Figure 5: Examples of neutral faces in the first video image

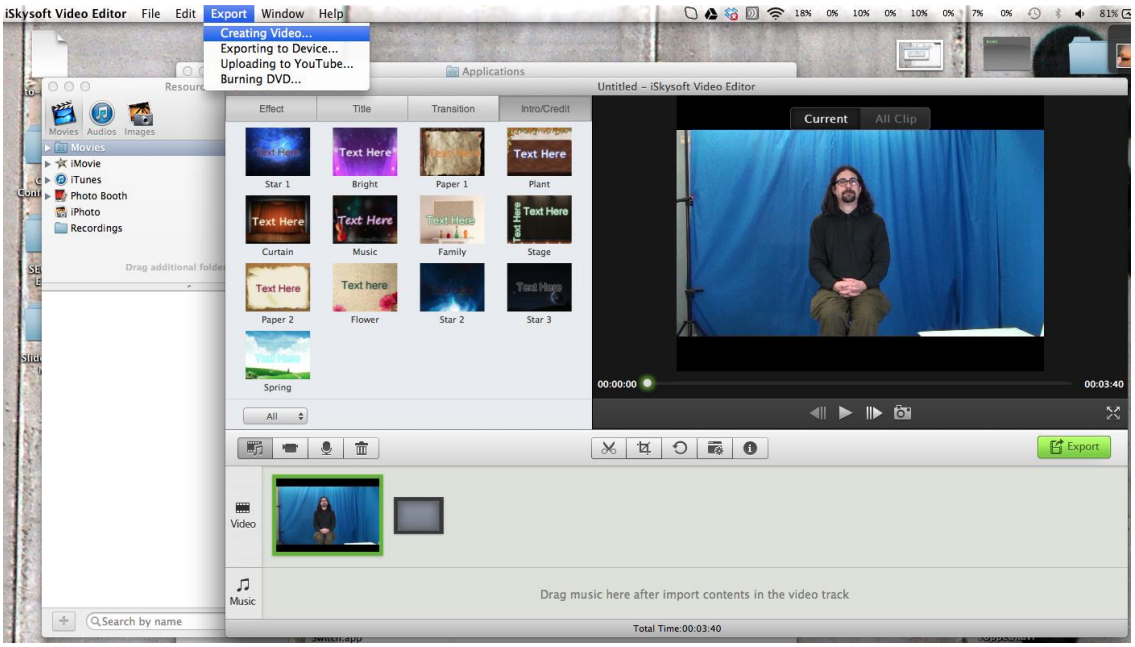
B. HOW TO EXPORT VIDEOS FOR VISAGE

STEP 1 (Best Recording)

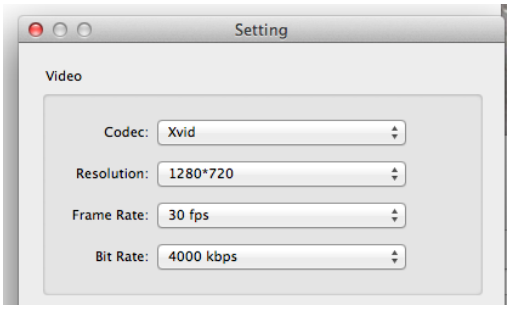
It is best if there is a second camera recording a close up of the face of the signer.

STEP 2: Convert MTS to AVI files

1. Open iSkySoft software
2. Select 4:3
3. Load the MTS file
4. Select 'Export' > 'Creating Video' from the menu



5. Choose format AVI and open 'Settings'

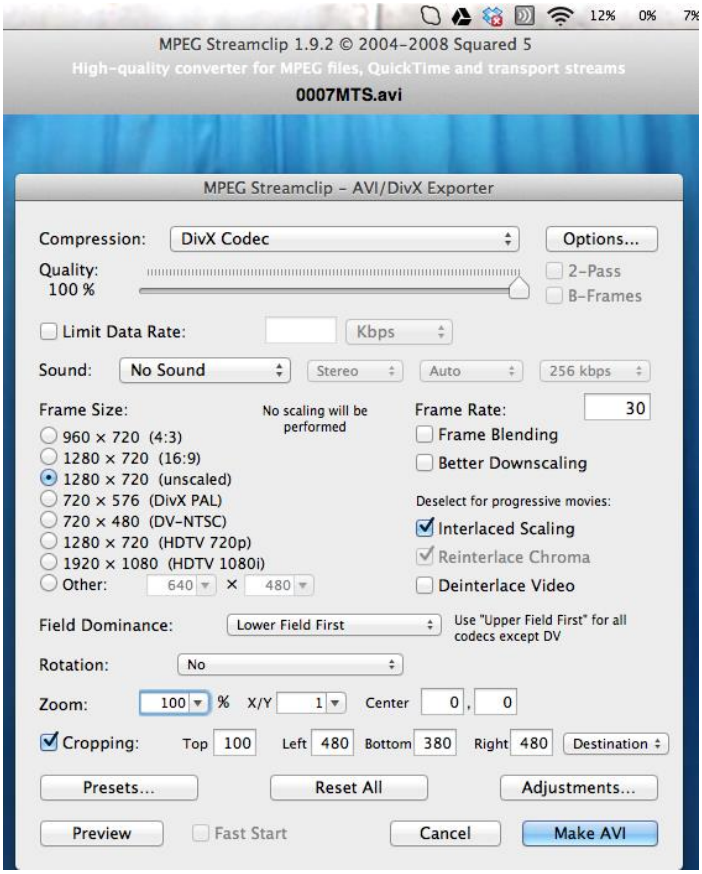


STEP 3: Prepare AVI for Visage

Focus on the face only (cover around 70% of the screen) and re-export to avi preferably with a high resolution (800x600 or 640x480)

1. Open MPEGStreamClip
2. Load video from step 2

3. Find the part of the video that is interesting and trim it with ‘i’ and ‘o’ from the keyboard.
4. Choose ‘Export to AVI’ and follow the settings (as shown below)



C. How to build a profile

Bla

D. How to obtain data from a video

Assume we already have the profile builded. If using the mac application then the only way to denote which profile files to load is by renaming the .su and .bmp file with the video name.

E.g. XXX.avi is in the same folder with XXX.su and XXX_texture.bmp

Bla

PART XI. BUILDING MACHINE-LEARNING MODELS FOR LIGUISTICALLY MEANINGFUL FACIAL EXPRESSIONS

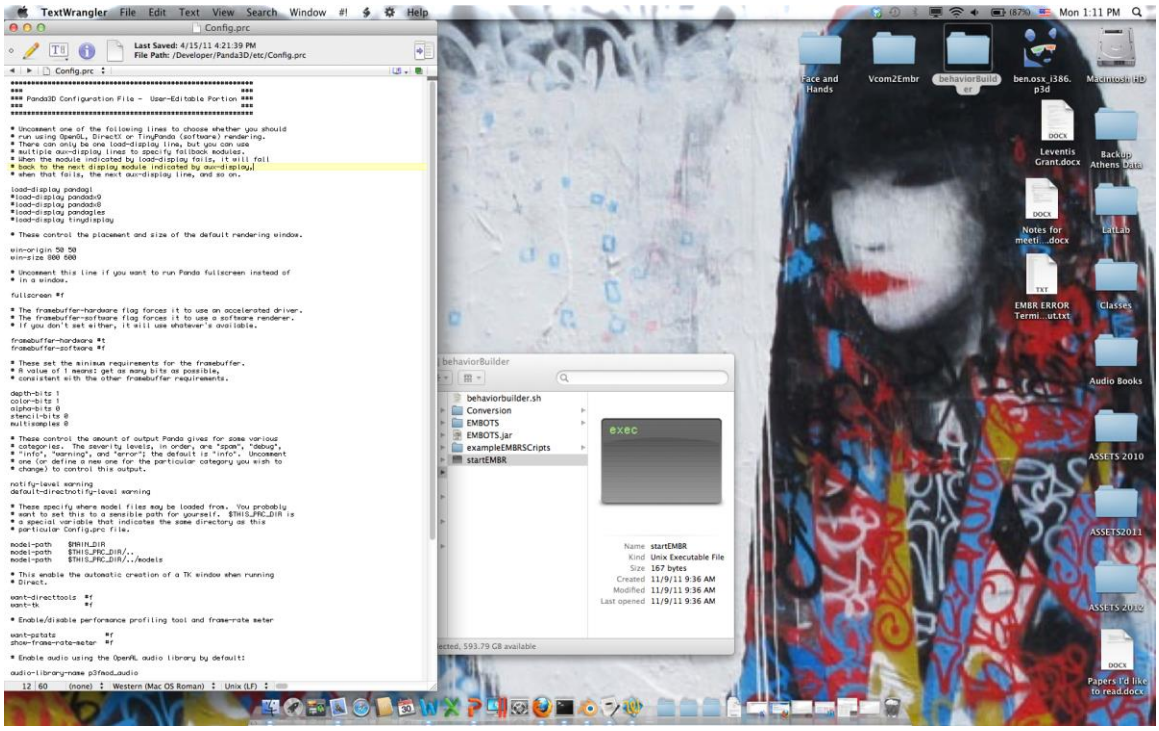
Note: models for describing the movement of face during e.g. y-n question. Beside these models based on machine learning techniques and the data extracted from the videos we could also develop (for comparison reason models based on the linguistic definition of each non-manual expression.)

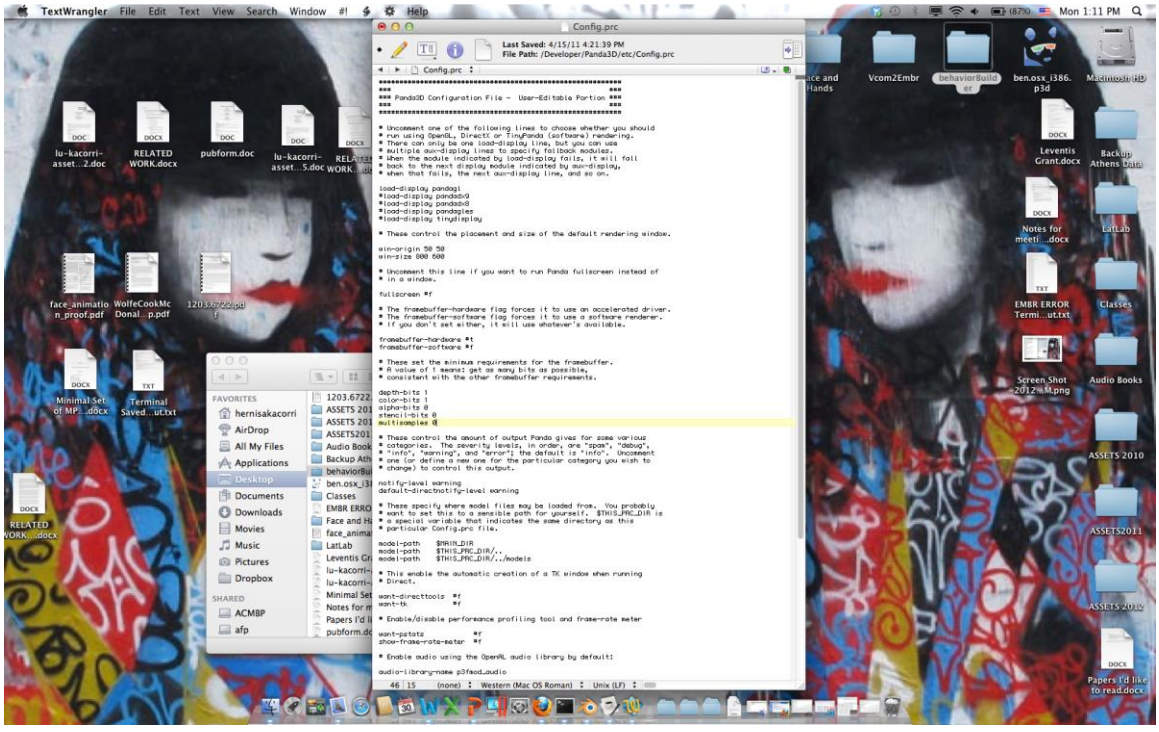
Models driven by data

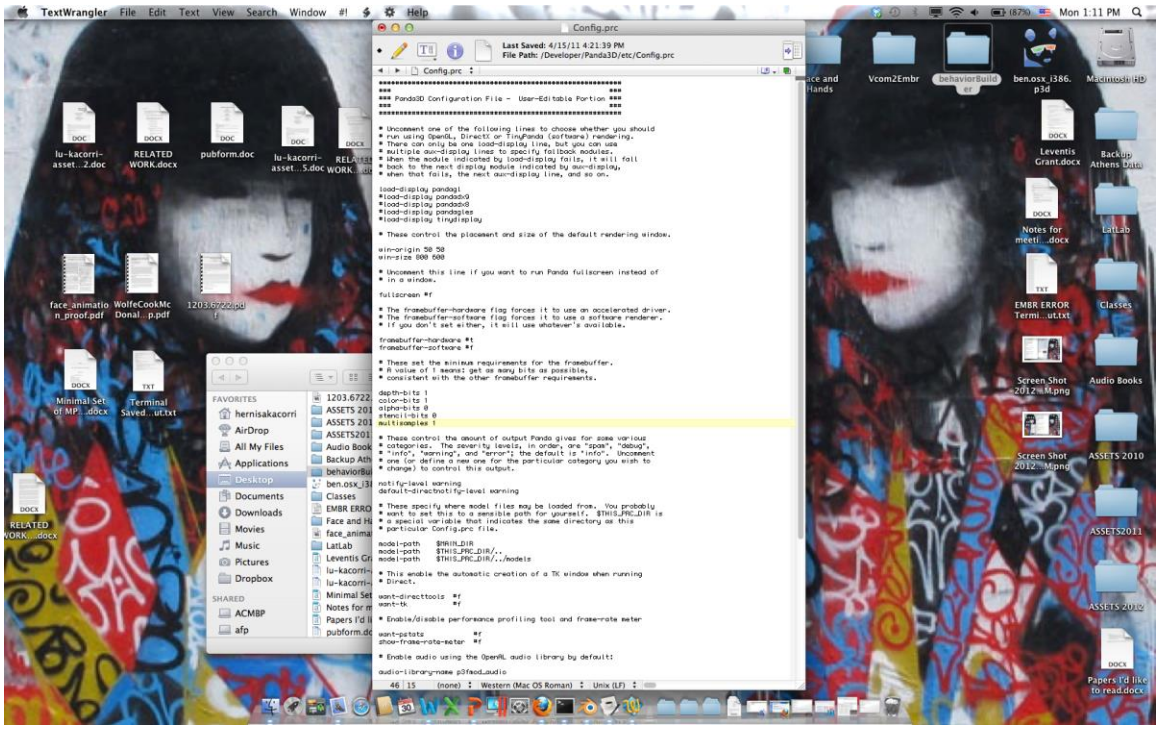
Models driven by linguists (assumption)

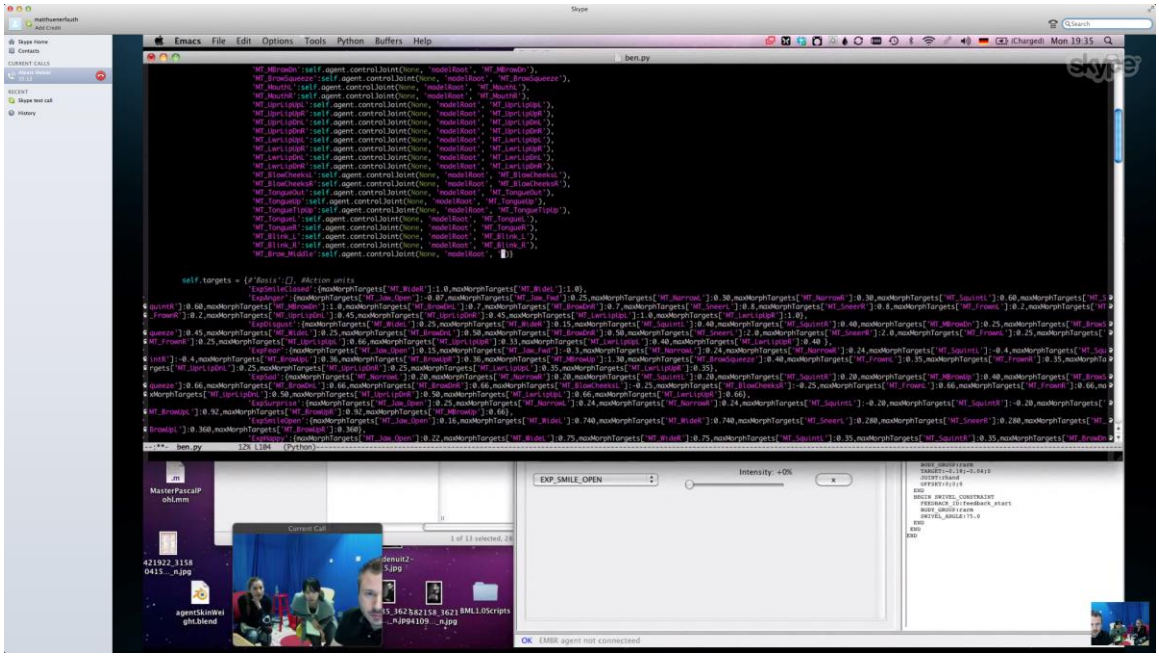
Combination

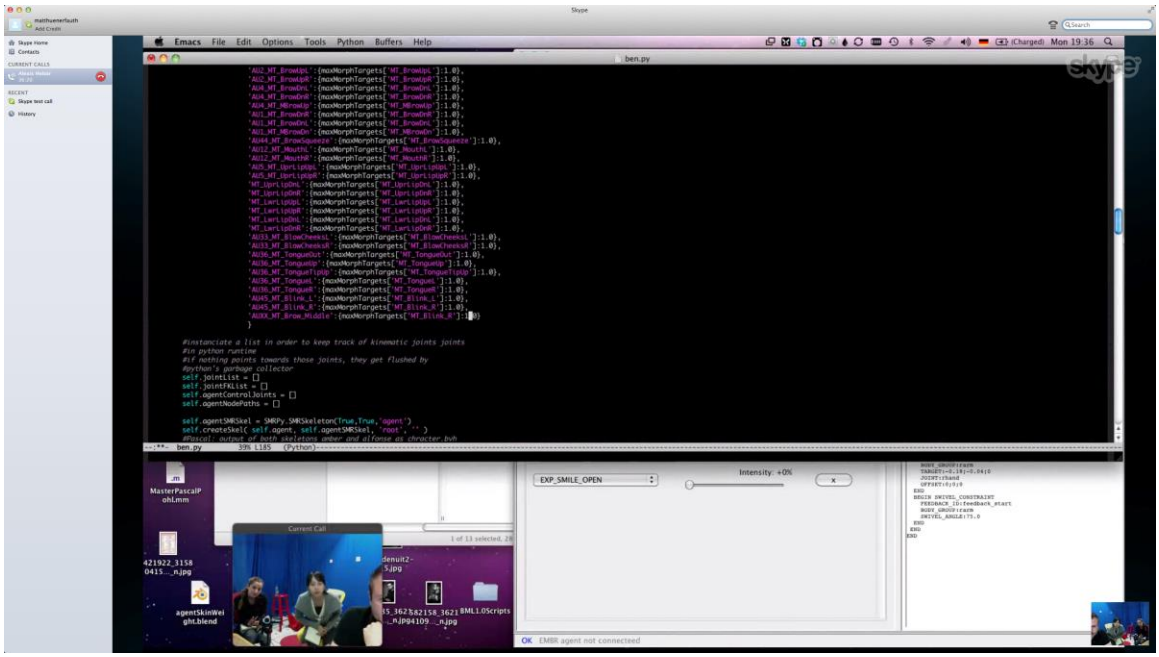
PART XII. APPENDIX I

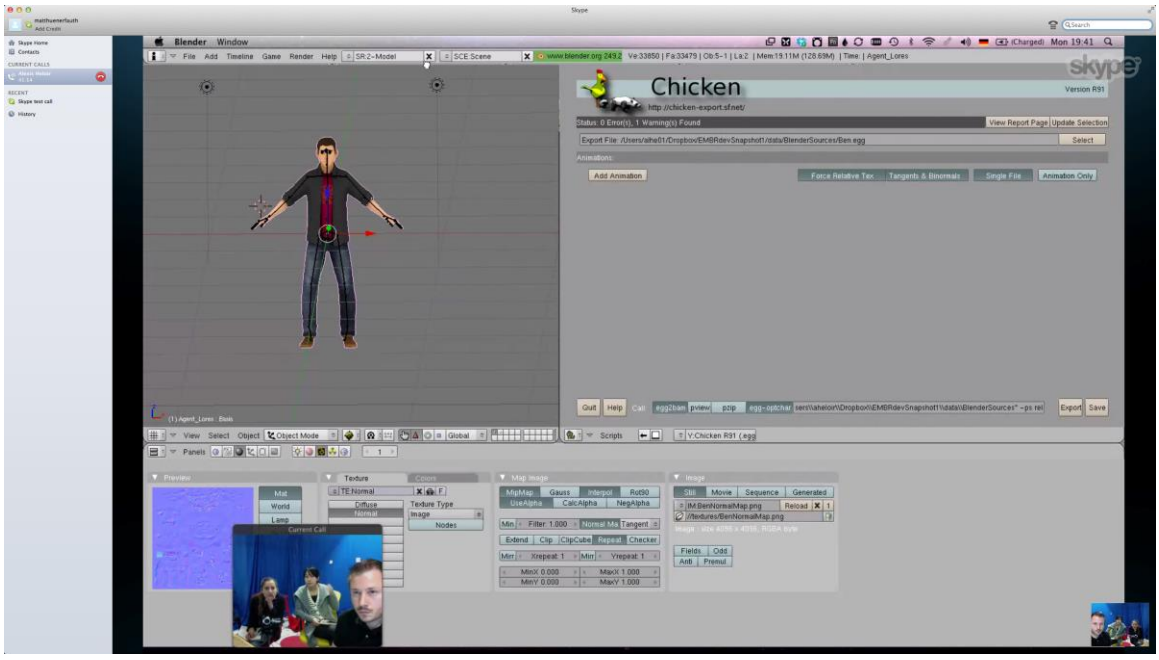


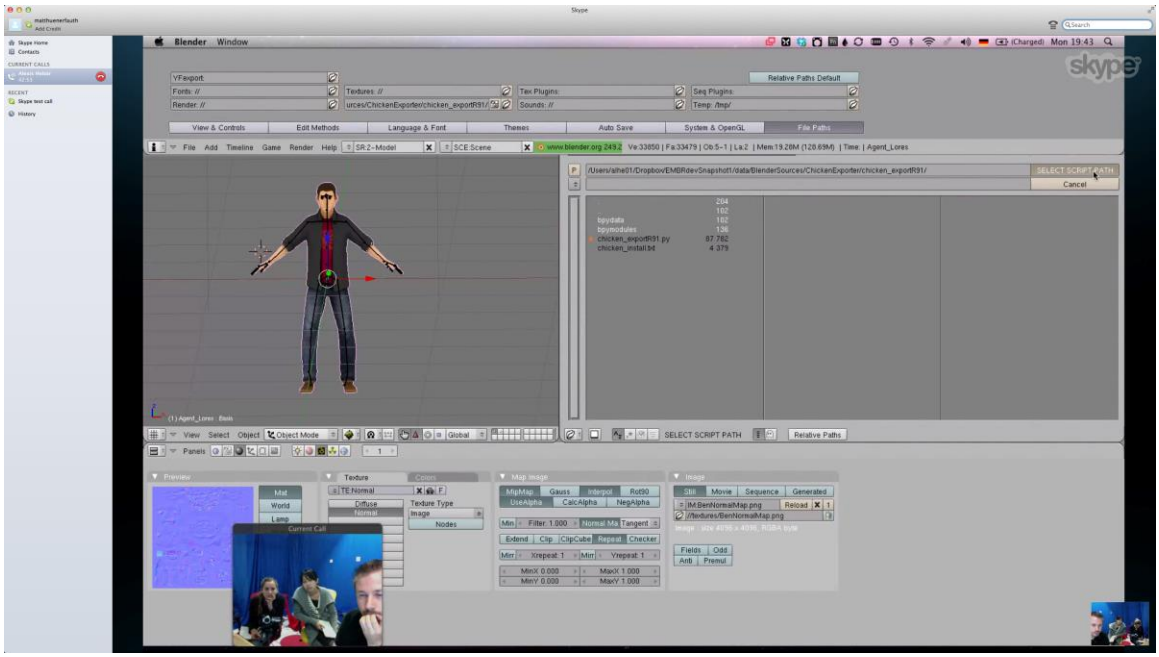


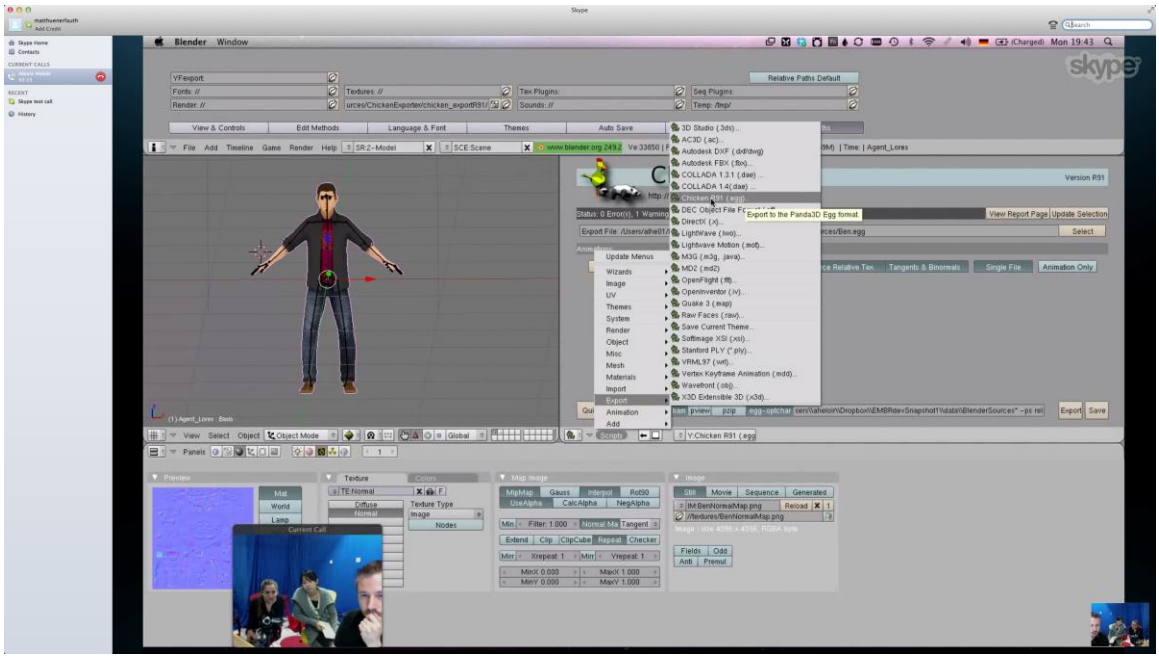

TextWrangler File Edit Text View Search Window #! Help
Mon 1:11 PM
Config.prc
Last Saved: 4/15/11 4:21:39 PM
File Path: /Developer/Panda3D/etc/Config.prc
Panda3D Configuration File - User-Editable Portion
load-display pandagl
load-display pandadx9
load-display pandadx8
load-display pandagles
load-display tinydisplay
win-origin 50 50
win-size 800 600
fullscreen #f
depth-bits 1
color-bits 1
alpha-bits 0
stencil-bits 0
multisamples 1
notify-level warning
default-directnotify-level warning
model-path $MAIN_DIR
model-path $THIS_PRC_DIR/..
model-path $THIS_PRC_DIR/../models
want-directtools #f
want-tk #f
want-pstats #f
show-frame-rate-meter #f
audio-library-name p3fmod_audio
Vcom2Embr
behaviorBuild
ben.osx_i386.p3d
Macintosh HD
Leventis Grant.docx
Backup Athens Data
Notes for meeti...docx
LatLab
EMBR ERROR Termi...ut.txt
Classes
Screen Shot 2012...4.png
Audio Books
Screen Shot 2012...4.png
ASSETS 2010
ASSETS2011
ASSETS 2012
Papers I'd like to read.docx
face_animation_proof.pdf
WolfeCookMcDonal...p.pdf
1203.6722.pdf
Minimal Set of MP...docx
Terminal Saved...ut.txt
RELATED WORK...docx
lu-kacorri-asset...2.doc
RELATED WORK.docx
pubform.doc
lu-kacorri-asset...5.doc
FAVORITES
hernisakacorri
AirDrop
All My Files
Applications
Desktop
Documents
Downloads
Movies
Music
Pictures
Dropbox
SHARED
ACMBP
afp

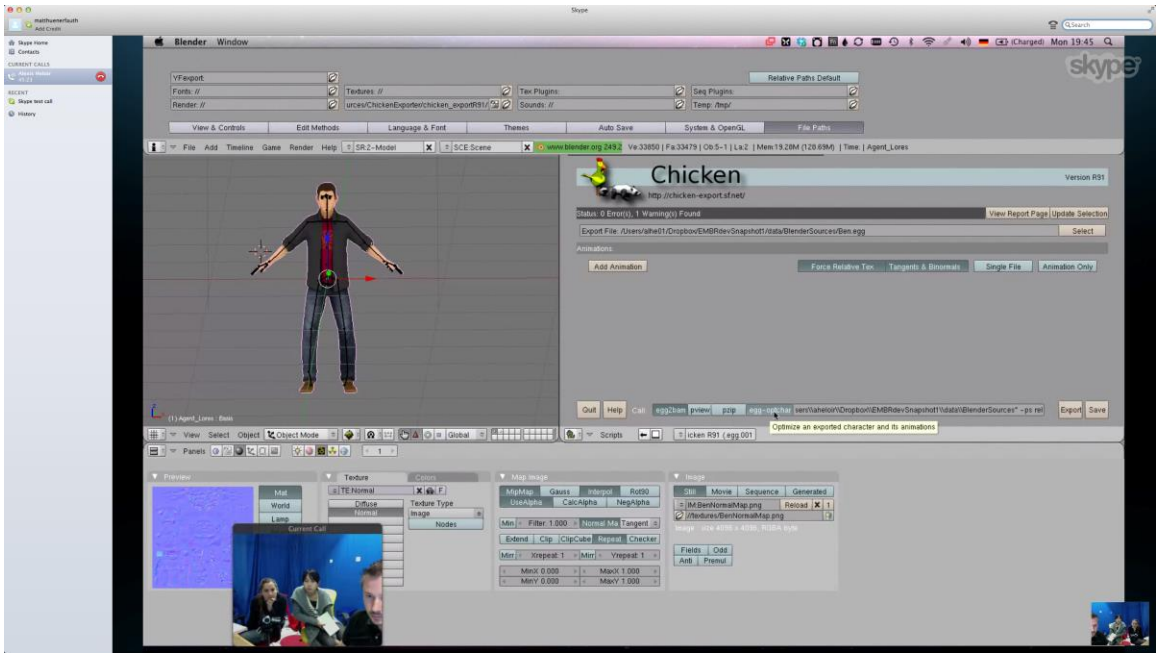

Skype
Emacs File Edit Options Tools Python Buffers Help
(Charged) Mon 19:35
ben.py
skype
MasterPascalPoh1.mm
agentSkinWeight.blend
EXP_SMILE_OPEN
Intensity: +0%

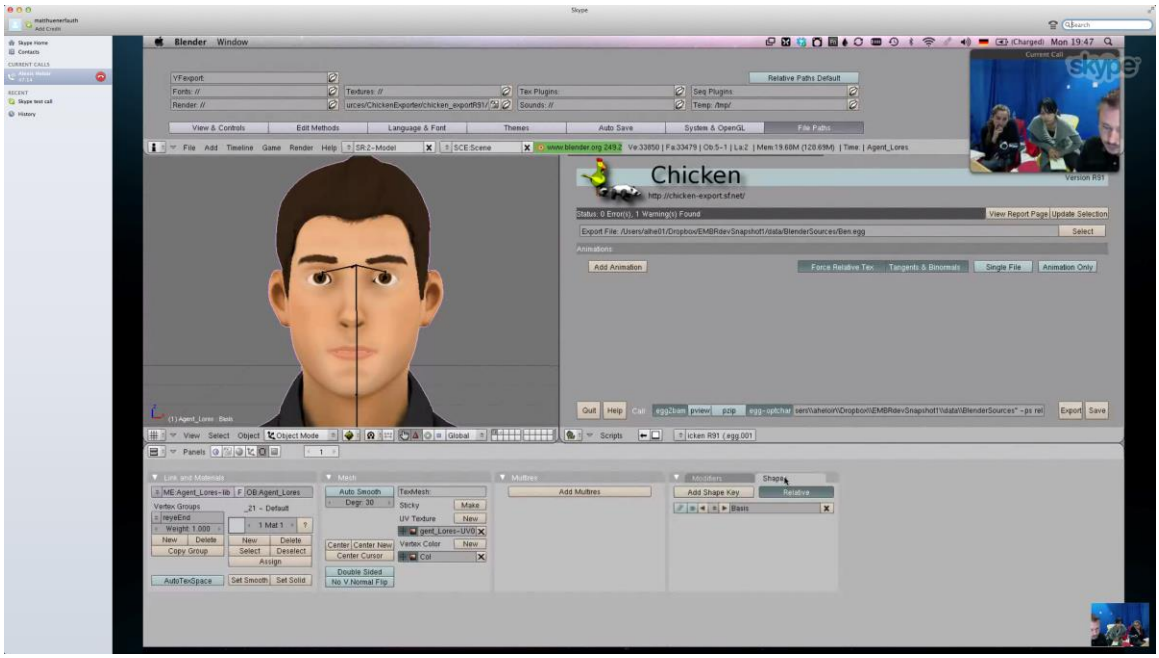

Emacs File Edit Options Tools Python Buffers Help
ben.py
EXP_SMILE_OPEN
Intensity: +0%
OK EMBR agent not connected

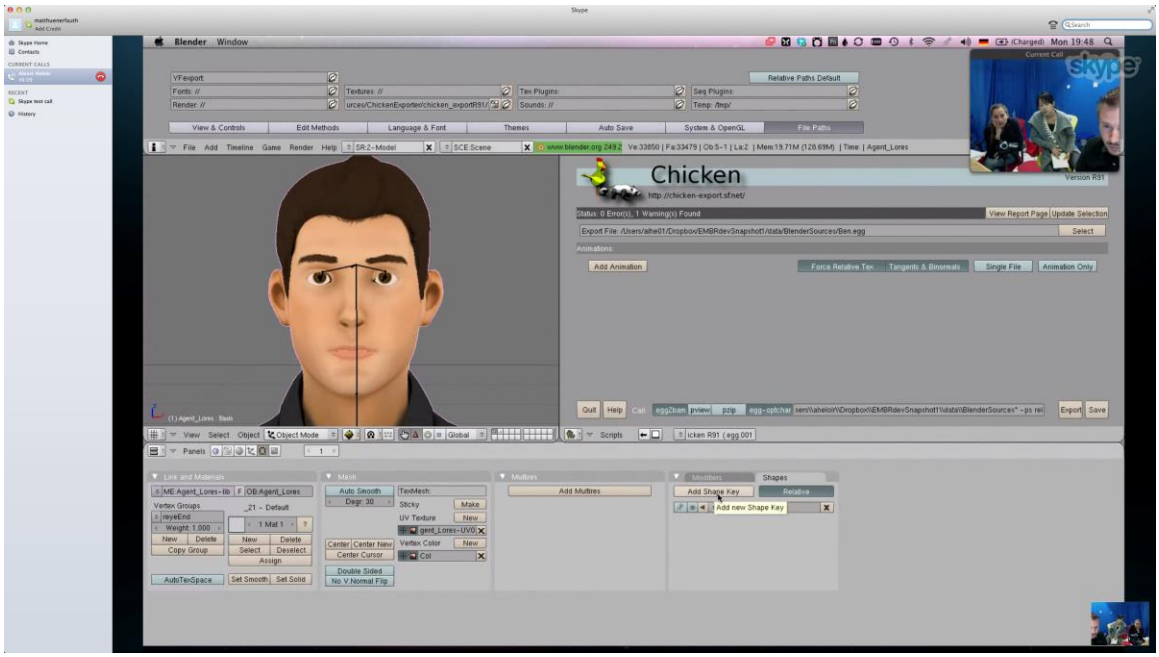

Blender Window
Chicken
Version R91
http://chicken-export.sf.net/
Add Animation
Export
Save

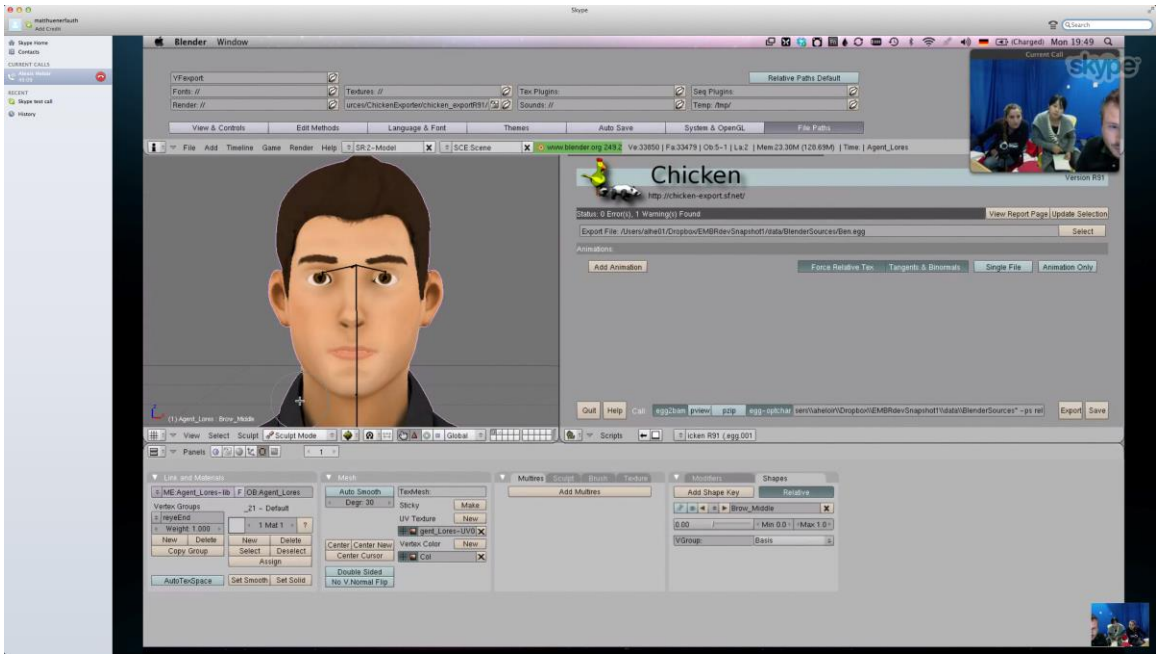

Blender Window
skype
Relative Paths Default
SELECT SCRIPT PATH
Cancel

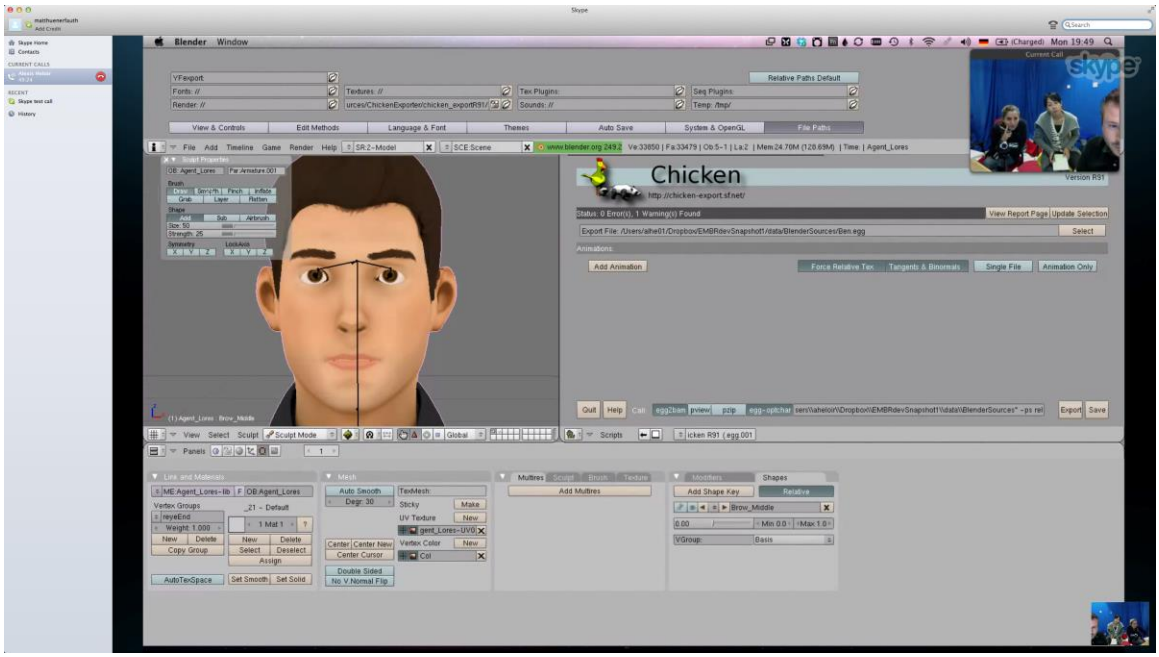

Blender Window
skype
Relative Paths Default
Export to the Panda3D Egg format.

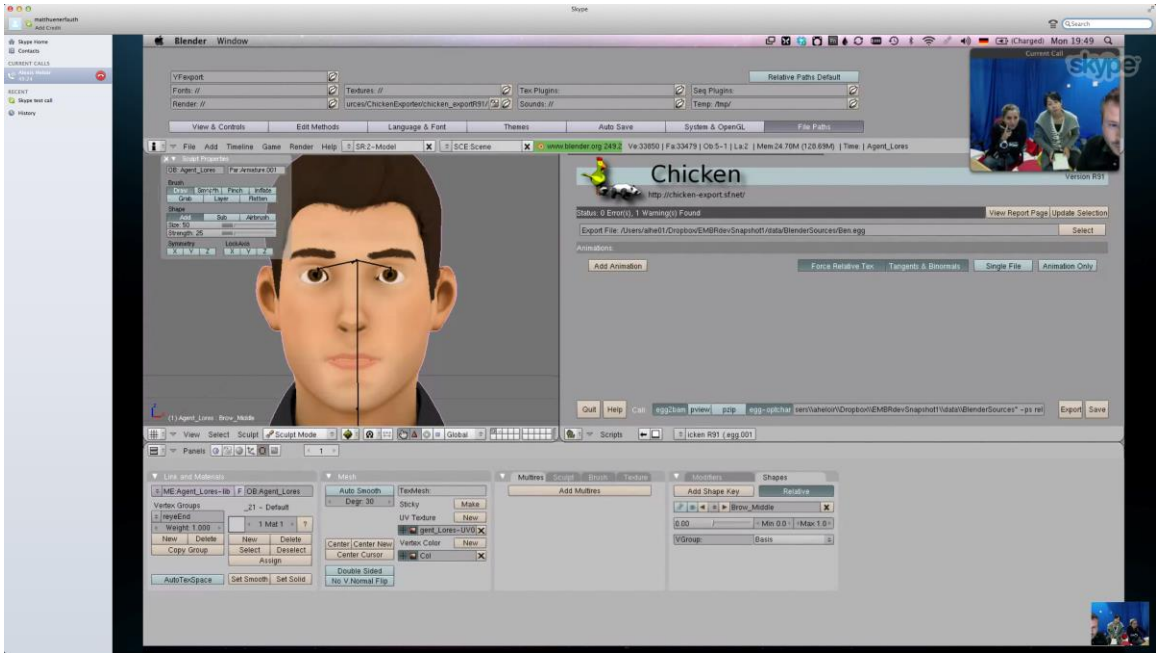

Blender Window
Chicken
Optimize an exported character and its animations

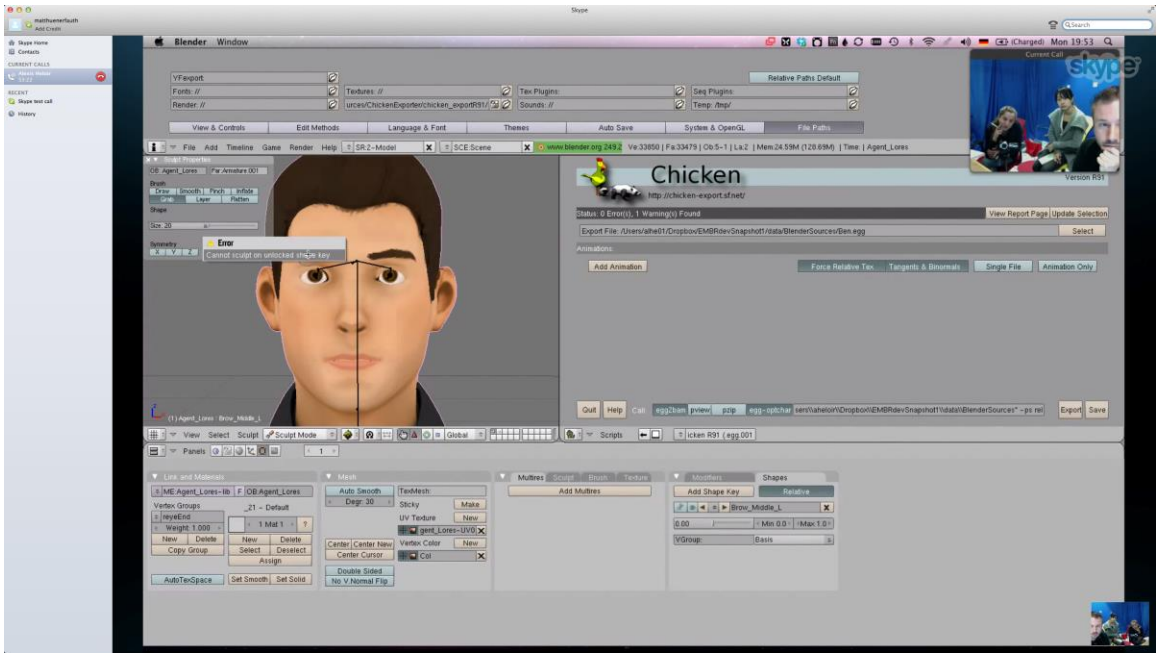

Blender Window
Chicken

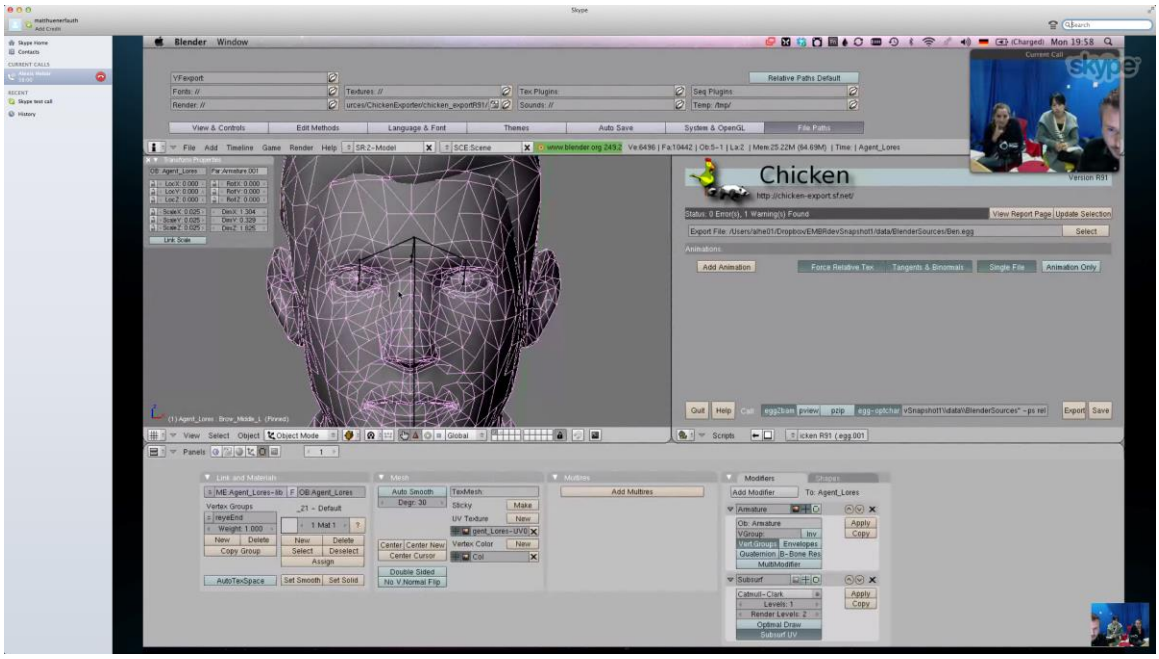

Blender Window
Chicken
http://chicken-export.sf.net/
Add Animation
Add Shape Key
Add new Shape Key

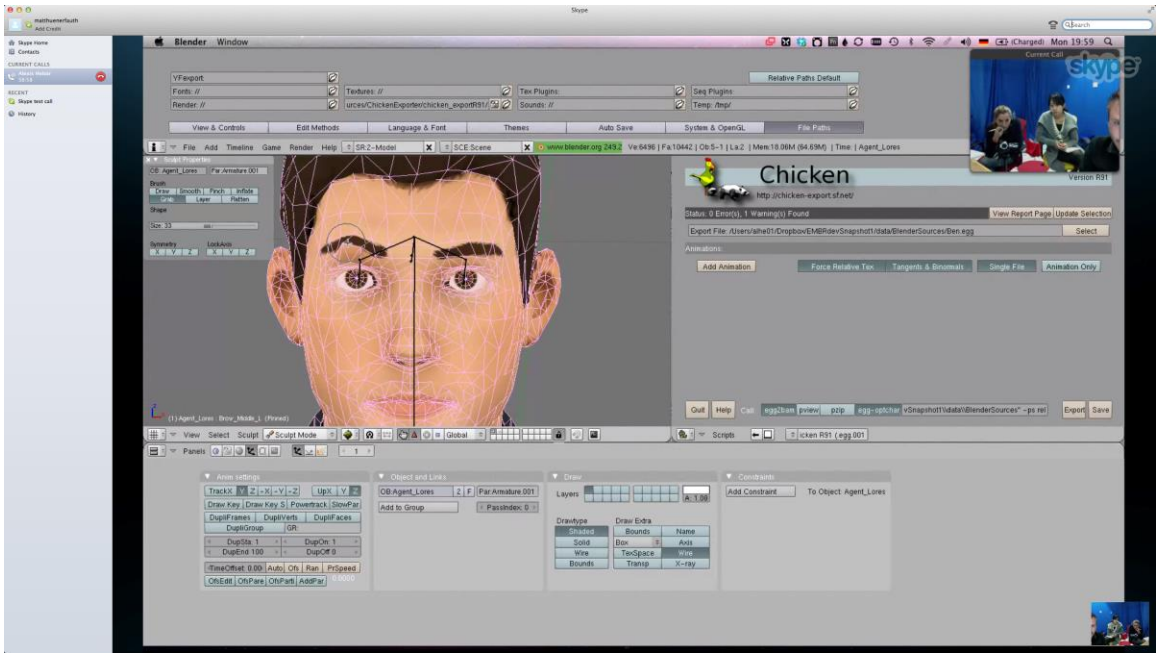

Blender Window
Chicken
http://chicken-export.sf.net/
Add Animation
Add Shape Key
Brow_Middle

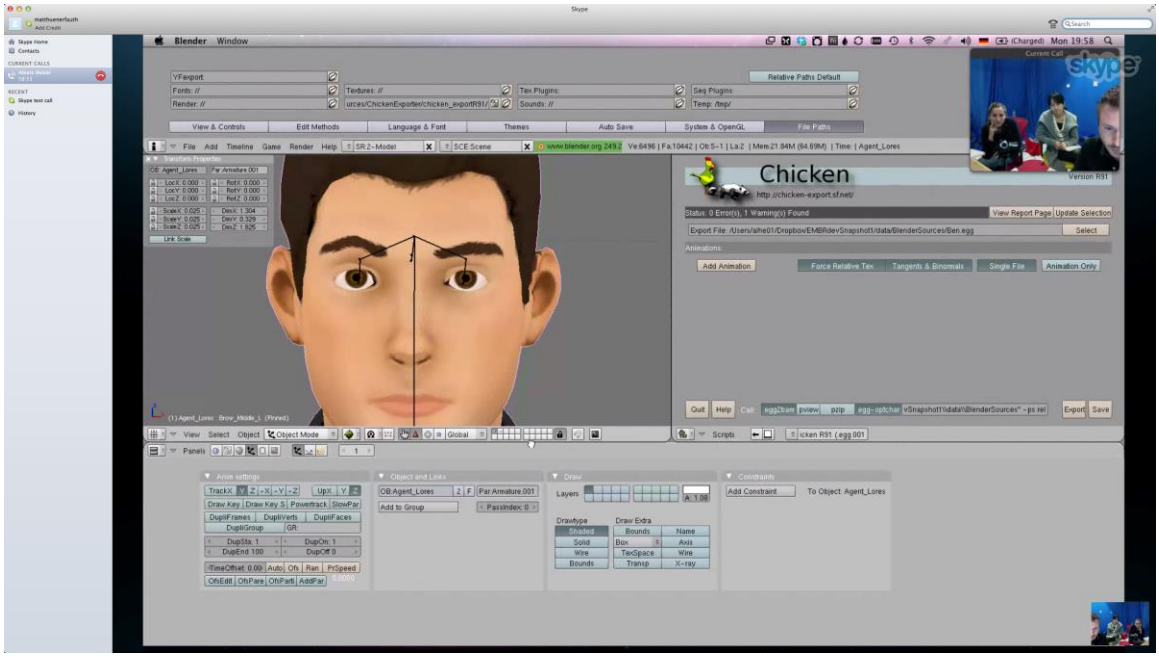

Blender Window
Chicken
http://chicken-export.sf.net/
Add Animation
Force Relative Tex
Tangents & Binormals
Single File
Animation Only
Quit
Help
Export
Save

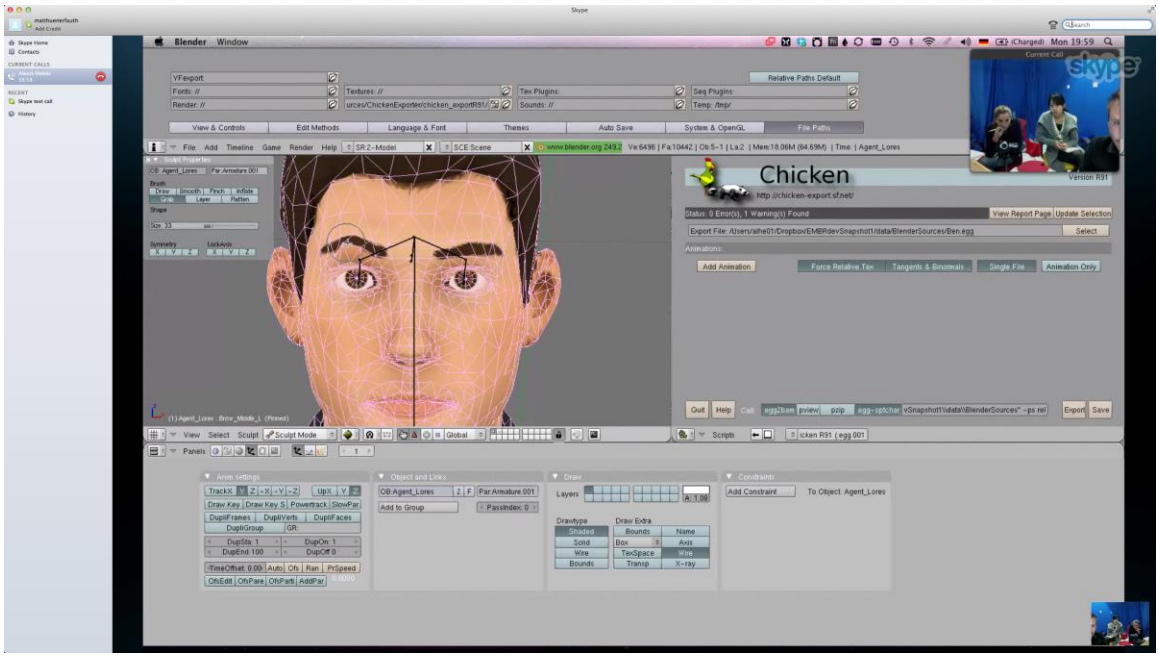

Blender Window
Chicken
http://chicken-export.sf.net/
Add Animation
Force Relative Tex
Tangents & Binormals
Single File
Animation Only
Quit
Help
Export
Save


Blender Window
Chicken
http://chicken-export.sf.net/
Error
Cannot sculpt on unlocked shape key

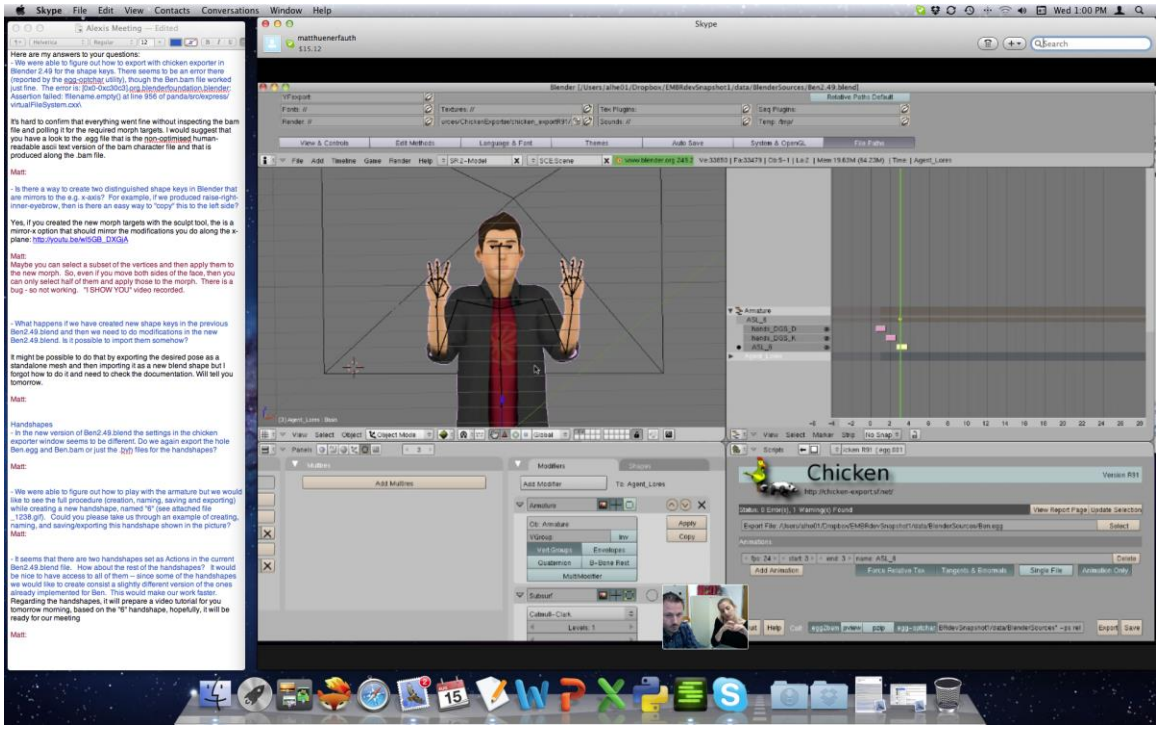

Blender Window
Chicken
http://chicken-export.sf.net/

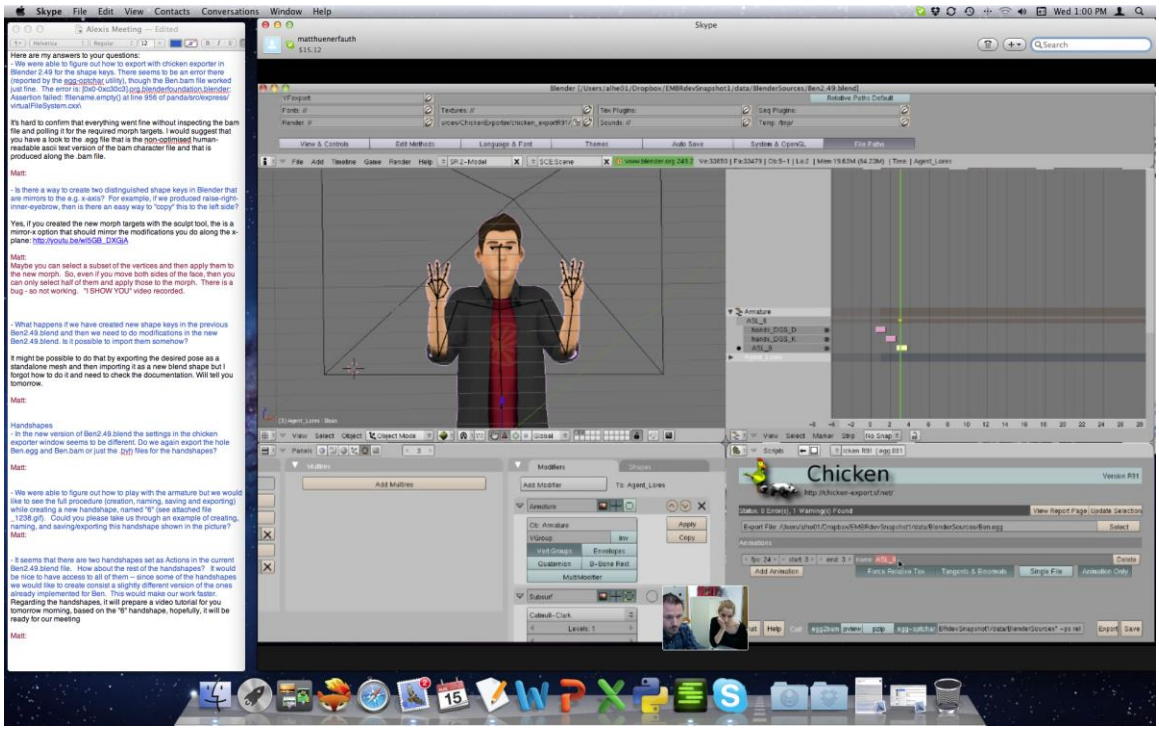

Blender Window
Chicken
http://chicken-export.sf.net/
Version R91

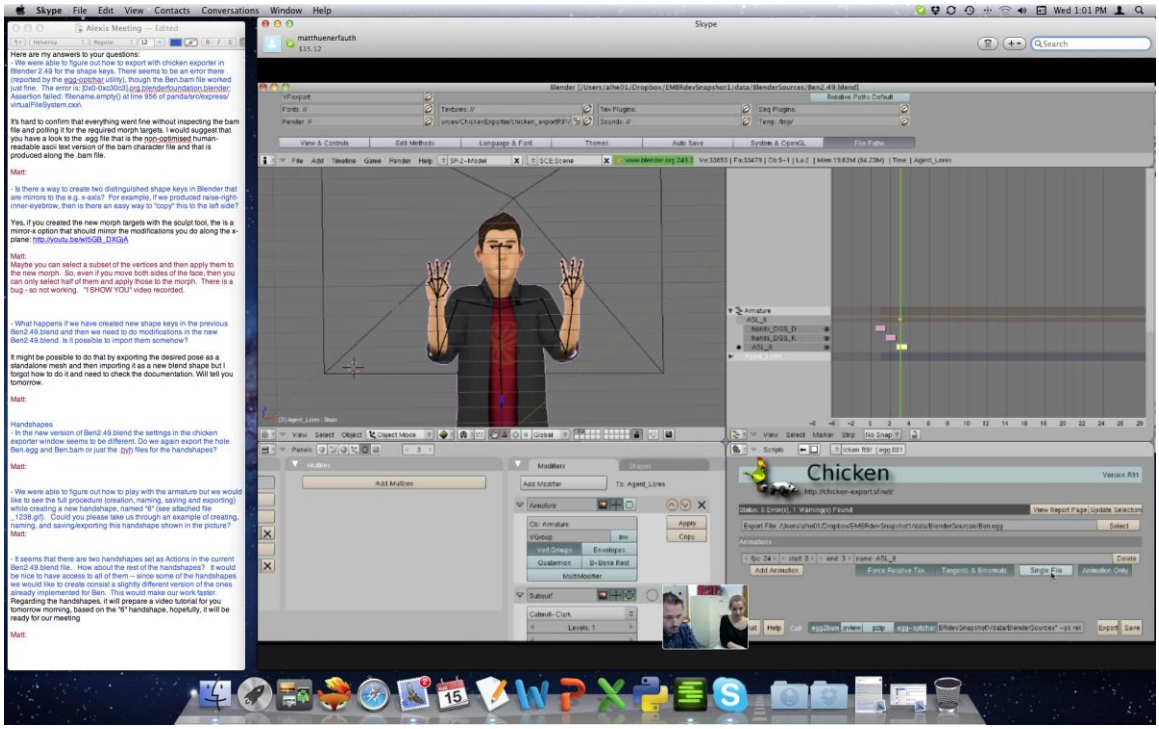

Blender Window
Chicken
http://chicken-export.sf.net/
Version R91


Blender Window
Chicken
http://chicken-export.sf.net/
Version R91
Add Animation
Force Relative Tex
Tangents & Binormals
Single File
Animation Only
Quit
Help
Export
Save
Add Constraint
Add to Group

PART XIII. APPENDIX II




Skype File Edit View Contacts Conversations Window Help
Wed 1:00 PM
Alexis Meeting — Edited
Here are my answers to your questions:
- We were able to figure out how to export with chicken exporter in Blender 2.49 for the shape keys. There seems to be an error there (reported by the egg-optchar utility), though the Ben.bam file worked just fine. The error is: [0x0-0xc30c3] org.blenderfoundation.blender: Assertion failed: filename.empty() at line 956 of panda/src/express/virtualFileSystem.cxx\
It's hard to confirm that everything went fine without inspecting the bam file and polling it for the required morph targets. I would suggest that you have a look to the .egg file that is the non-optimised human-readable ascii text version of the bam character file and that is produced along the .bam file.
Matt:
- Is there a way to create two distinguished shape keys in Blender that are mirrors to the e.g. x-axis? For example, if we produced raise-right-inner-eyebrow, then is there an easy way to "copy" this to the left side?
Yes, if you created the new morph targets with the sculpt tool, the is a mirror-x option that should mirror the modifications you do along the x-plane: http://youtu.be/wISGB_DXGjA
Matt:
Maybe you can select a subset of the vertices and then apply them to the new morph. So, even if you move both sides of the face, then you can only select half of them and apply those to the morph. There is a bug - so not working. "I SHOW YOU" video recorded.
- What happens if we have created new shape keys in the previous Ben2.49.blend and then we need to do modifications in the new Ben2.49.blend. Is it possible to import them somehow?
It might be possible to do that by exporting the desired pose as a standalone mesh and then importing it as a new blend shape but I forgot how to do it and need to check the documentation. Will tell you tomorrow.
Matt:
Handshapes
- In the new version of Ben2.49.blend the settings in the chicken exporter window seems to be different. Do we again export the hole Ben.egg and Ben.bam or just the .bvh) files for the handshapes?
Matt:
- We were able to figure out how to play with the armature but we would like to see the full procedure (creation, naming, saving and exporting) while creating a new handshape, named "6" (see attached file _1238.gif). Could you please take us through an example of creating, naming, and saving/exporting this handshape shown in the picture?
Matt:
- It seems that there are two handshapes set as Actions in the current Ben2.49.blend file. How about the rest of the handshapes? It would be nice to have access to all of them – since some of the handshapes we would like to create consist a slighlty different version of the ones already implemented for Ben. This would make our work faster.
Regarding the handshapes, it will prepare a video tutorial for you tomorrow morning, based on the "6" handshape, hopefully, it will be ready for our meeting
Matt:
Skype
matthuenerfauth
$15.12
Chicken
Version R91
http://chicken-export.sf.net/


Skype File Edit View Contacts Conversations Window Help
Wed 1:01 PM
Alexis Meeting — Edited
Here are my answers to your questions:
- We were able to figure out how to export with chicken exporter in Blender 2.49 for the shape keys. There seems to be an error there (reported by the egg-optchar utility), though the Ben.bam file worked just fine. The error is: [0x0-0xc30c3] org.blenderfoundation.blender: Assertion failed: filename.empty() at line 956 of panda/src/express/virtualFileSystem.cxx\
It's hard to confirm that everything went fine without inspecting the bam file and polling it for the required morph targets. I would suggest that you have a look to the .egg file that is the non-optimised human-readable ascii text version of the bam character file and that is produced along the .bam file.
Matt:
- Is there a way to create two distinguished shape keys in Blender that are mirrors to the e.g. x-axis? For example, if we produced raise-right-inner-eyebrow, then is there an easy way to "copy" this to the left side?
Yes, if you created the new morph targets with the sculpt tool, the is a mirror-x option that should mirror the modifications you do along the x-plane: http://youtu.be/wISGB_DXGjA
Matt:
Maybe you can select a subset of the vertices and then apply them to the new morph. So, even if you move both sides of the face, then you can only select half of them and apply those to the morph. There is a bug - so not working. "I SHOW YOU" video recorded.
- What happens if we have created new shape keys in the previous Ben2.49.blend and then we need to do modifications in the new Ben2.49.blend. Is it possible to import them somehow?
It might be possible to do that by exporting the desired pose as a standalone mesh and then importing it as a new blend shape but I forgot how to do it and need to check the documentation. Will tell you tomorrow.
Matt:
Handshapes
- In the new version of Ben2.49.blend the settings in the chicken exporter window seems to be different. Do we again export the hole Ben.egg and Ben.bam or just the .bvh) files for the handshapes?
Matt:
- We were able to figure out how to play with the armature but we would like to see the full procedure (creation, naming, saving and exporting) while creating a new handshape, named "6" (see attached file _1238.gif). Could you please take us through an example of creating, naming, and saving/exporting this handshape shown in the picture?
Matt:
- It seems that there are two handshapes set as Actions in the current Ben2.49.blend file. How about the rest of the handshapes? It would be nice to have access to all of them – since some of the handshapes we would like to create consist a slighlty different version of the ones already implemented for Ben. This would make our work faster.
Regarding the handshapes, it will prepare a video tutorial for you tomorrow morning, based on the "6" handshape, hopefully, it will be ready for our meeting
Matt:
Skype
matthuenerfauth
$15.12
Chicken
Version R91
http://chicken-export.sf.net/

PART XIV. APPENDIX III – MPEG-4 FAPS

MPEG-4 FAP definitions, group assignments, and step sizes. FAP names may contain letters with the following meaning: l = left, r = right, t = top, b = bottom, i = inner, o = outer, m = middle. The quantizer step-size is a scaling factor for coding.

#	FAP name	FAP description	units	Uni-orBidir	Pos motion	Grp	FDP subgrp num	Quant step size QP	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
1	viseme	Set of values determining the mixture of two visemes for this frame (e.g. pbm, fv, th)	na	na	na	1	na	1	viseme_blend: +63	viseme_blend: +-63
2	expression	A set of values determining the mixture of two facial expression	na	na	na	1	na	1	expression_intensity1, expression_intensity2: +63	expression_intensity1, expression_intensity2: +-63
3	open_jaw	Vertical jaw displacement (does not affect mouth opening)	MNS	U	down	2	1	4	+1080	+360
4	lower_t_midlip	Vertical top middle inner lip displacement	MNS	B	down	2	2	2	+-600	+-180
5	raise_b_midlip	Vertical bottom middle inner lip displacement	MNS	B	up	2	3	2	+-1860	+-600
6	stretch_l_cornerlip	Horizontal displacement of left inner lip corner	MW	B	left	2	4	2	+-600	+-180
7	stretch_r_cornerlip	Horizontal displacement of right inner lip corner	MW	B	right	2	5	2	+-600	+-180
8	lower_t_lip_lm	Vertical displacement of midpoint between left corner and middle of top inner lip	MNS	B	down	2	6	2	+-600	+-180

9	lower_t_lip_rm	Vertical displacement of midpoint between right corner and middle of top inner lip	MNS	B	down	2	7	2	+-600	+-180
10	raise_b_lip_lm	Vertical displacement of midpoint between left corner and middle of bottom inner lip	MNS	B	up	2	8	2	+-1860	+-600
11	raise_b_lip_rm	Vertical displacement of midpoint between right corner and middle of bottom inner lip	MNS	B	up	2	9	2	+-1860	+-600
12	raise_l_cornerlip	Vertical displacement of left inner lip corner	MNS	B	up	2	4	2	+-600	+-180
13	raise_r_cornerlip	Vertical displacement of right inner lip corner	MNS	B	up	2	5	2	+-600	+-180
14	thrust_jaw	Depth displacement of jaw	MNS	U	forward	2	1	1	+600	+180
15	shift_jaw	Side to side displacement of jaw	MW	B	right	2	1	1	+-1080	+-360
16	push_b_lip	Depth displacement of bottom middle lip	MNS	B	forward	2	3	1	+-1080	+-360
17	push_t_lip	Depth displacement of top middle lip	MNS	B	forward	2	2	1	+-1080	+-360
18	depress_chin	Upward and compressing movement of the chin (like in sadness)	MNS	B	up	2	10	1	+-420	+-180
19	close_t_l_eyelid	Vertical displacement of top left eyelid	IRISD	B	down	3	1	1	+-1080	+-600
20	close_t_r_eyelid	Vertical	IRIS	B	down	3	2	1	+-1080	+-600

		displacement of top right eyelid	D							
21	close_b_l_eyelid	Vertical displacement of bottom left eyelid	IRIS D	B	up	3	3	1	+-600	+-240
22	close_b_r_eyelid	Vertical displacement of bottom right eyelid	IRIS D	B	up	3	4	1	+-600	+-240
23	yaw_l_eyeball	Horizontal orientation of left eyeball	AU	B	left	3	na	128	+-1200	+-420
24	yaw_r_eyeball	Horizontal orientation of right eyeball	AU	B	left	3	na	128	+-1200	+-420
25	pitch_l_eyeball	Vertical orientation of left eyeball	AU	B	down	3	na	128	+-900	+-300
26	pitch_r_eyeball	Vertical orientation of right eyeball	AU	B	down	3	na	128	+-900	+-300
27	thrust_l_eyeball	Depth displacement of left eyeball	ES	B	forward	3	na	1	+-600	+-180
28	thrust_r_eyeball	Depth displacement of right eyeball	ES	B	forward	3	na	1	+-600	+-180
29	dilate_l_pupil	Dilation of left pupil	IRIS D	B	growing	3	5	1	+-420	+-120
30	dilate_r_pupil	Dilation of right pupil	IRIS D	B	growing	3	6	1	+-420	+-120
31	raise_l_i_eyebrow	Vertical displacement of left inner eyebrow	ENS	B	up	4	1	2	+-900	+-360
32	raise_r_i_eyebrow	Vertical displacement of right inner eyebrow	ENS	B	up	4	2	2	+-900	+-360
33	raise_l_m_eyebrow	Vertical displacement of left middle eyebrow	ENS	B	up	4	3	2	+-900	+-360
34	raise_r_m_eyebrow	Vertical displacement of right middle eyebrow	ENS	B	up	4	4	2	+-900	+-360

35	raise_l_o_eyebrow	Vertical displacement of left outer eyebrow	ENS	B	up	4	5	2	+-900	+-360
36	raise_r_o_eyebrow	Vertical displacement of right outer eyebrow	ENS	B	up	4	6	2	+-900	+-360
37	squeeze_l_eyebrow	Horizontal displacement of left eyebrow	ES	B	right	4	1	1	+-900	+-300
38	squeeze_r_eyebrow	Horizontal displacement of right eyebrow	ES	B	left	4	2	1	+-900	+-300
39	puff_l_cheek	Horizontal displacement of left cheeck	ES	B	left	5	1	2	+-900	+-300
40	puff_r_cheek	Horizontal displacement of right cheeck	ES	B	right	5	2	2	+-900	+-300
41	lift_l_cheek	Vertical displacement of left cheek	ENS	U	up	5	3	2	+-600	+-180
42	lift_r_cheek	Vertical displacement of right cheek	ENS	U	up	5	4	2	+-600	+-180
43	shift_tongue_tip	Horizontal displacement of tongue tip	MW	B	right	6	1	1	+-1080	+-420
44	raise_tongue_tip	Vertical displacement of tongue tip	MNS	B	up	6	1	1	+-1080	+-420
45	thrust_tongue_tip	Depth displacement of tongue tip	MW	B	forward	6	1	1	+-1080	+-420
46	raise_tongue	Vertical displacement of tongue	MNS	B	up	6	2	1	+-1080	+-420
47	tongue_roll	Rolling of the tongue into U shape	AU	U	concave upward	6	3, 4	512	+300	+60
48	head_pitch	Head pitch angle from top of spine	AU	B	down	7	na	170	+-1860	+-600
49	head_yaw	Head yaw angle from top of spine	AU	B	left	7	na	170	+-1860	+-600
50	head_roll	Head roll angle from top of spine	AU	B	right	7	na	170	+-1860	+-600

51	lower_t_midlip _o	Vertical top middle outer lip displacement	MNS	B	down	8	1	2	+-600	+-180
52	raise_b_midlip_o	Vertical bottom middle outer lip displacement	MNS	B	up	8	2	2	+-1860	+-600
53	stretch_l_cornerlip _o	Horizontal displacement of left outer lip corner	MW	B	left	8	3	2	+-600	+-180
54	stretch_r_cornerlip _o	Horizontal displacement of right outer lip corner	MW	B	right	8	4	2	+-600	+-180
55	lower_t_lip_lm _o	Vertical displacement of midpoint between left corner and middle of top outer lip	MNS	B	down	8	5	2	+-600	+-180
56	lower_t_lip_rm _o	Vertical displacement of midpoint between right corner and middle of top outer lip	MNS	B	down	8	6	2	+-600	+-180
57	raise_b_lip_lm_o	Vertical displacement of midpoint between left corner and middle of bottom outer lip	MNS	B	up	8	7	2	+-1860	+-600
58	raise_b_lip_rm_o	Vertical displacement of midpoint between right corner and middle of bottom outer lip	MNS	B	up	8	8	2	+-1860	+-600
59	raise_l_cornerlip_o	Vertical displacement of left outer lip corner	MNS	B	up	8	3	2	+-600	+-180
60	raise_r_cornerlip _o	Vertical displacement of right outer lip corner	MNS	B	up	8	4	2	+-600	+-180
61	stretch_l_nose	Horizontal	ENS	B	left	9	1	1	+-540	+-120

		displacement of left side of nose								
62	stretch_r_nose	Horizontal displacement of right side of nose	ENS	B	right	9	2	1	+-540	+-120
63	raise_nose	Vertical displacement of nose tip	ENS	B	up	9	3	1	+-680	+-180
64	bend_nose	Horizontal displacement of nose tip	ENS	B	right	9	3	1	+-900	+-180
65	raise_l_ear	Vertical displacement of left ear	ENS	B	up	10	1	1	+-900	+-240
66	raise_r_ear	Vertical displacement of right ear	ENS	B	up	10	2	1	+-900	+-240
67	pull_l_ear	Horizontal displacement of left ear	ENS	B	left	10	3	1	+-900	+-300
68	pull_r_ear	Horizontal displacement of right ear	ENS	B	right	10	4	1	+-900	+-300