Workshop on Research Methods in Linguistics: Scripting with PennController for IbexFarm

Spanish and Portuguese Department, Univerity of California Los Angeles

Gemma Repiso-Puigdelliura

What is PCIbex?

- **IbexFarm:** Ibex Farm is an online platform that allows researchers to run online experiments and store results at no cost.

- **PennController**: *PennController for IBEX* is a free open-sourceis a solution to script experiments with multimedia features in a user-friendly manner.

- Disclaimer: This tutorial is a shortened version of the official tutorial of PCIbex (https://www.pcibex.net/wiki/00-overview/).

*My contribution is a demonstration of how to collect voice recordings.

First Steps

- Create an account on PCIbex: https://www.pcibex.net/
- Request an account at HumSpace: https://humspace.ucla.edu/
  - If you do not have a UCLA account, you may use a server ($).
  - Note: Box may work as a server for pictures but not to store recording.
    - I have tried free server options with no success. Using your PC as a server does not work either, because you have to purchase a hosting space.

Experiment

- A welcome screen with
  (i) some introductory text,
  (ii) a text input box to record a participant ID, and
  (iii) a button to start the experiment
- A series of trials all following the same pattern:
  - Two pictures appearing side by side
  - A sentence and audio
  - Wait until one of the two pictures is selected (by either click or button press) and the audio is done playing
  - End of trial
- A final screen with a confirmation link

- Extra: Collecting audio!

Setting Up

Log in: https://expt.pcibex.net/login
Set up:
- Create a new experiment
- In the input box (*repo url*) enter
https://github.com/PennController/TimedPictureSelection

Reminders:

The script ends with a semicolon ;
And we need semicolons after every trial (but PCIbex tells us when semicolons are missing)
Remember to close every parentheses

Inserting elements: text and image

newText("The fish swim in a tank which is perfectly round")

1. .print()
2. ,
3. newImage("2fishRoundTank.png")
4. .print()

Adding a trial

1. newTrial(
2. newText("The fish swim in a tank which is perfectly round")
3. .print()
4. ,
5. newImage("2fishRoundTank.png")
6. .print()
7.)

Adding another image and a key response

```
1. newTrial(
2. newText("The fish swim in a tank which is perfectly round")
3. .print()
4. ,
5. newImage("2fishRoundTank.png")
6. .print()
7. ,
8. newImage("1fishSquareTank.png")
9. .print()
10. ,
11. newKey("FJ")
12. .wait()
13. )
```

Adjusting images

```
1. newImage("2fishRoundTank.png")
2. .size(200,200)
3. .print()
4. ,
5. newImage("1fishSquareTank.png")
6. .size(200,200)
7. .print()
8. ,
```

Images side by side

```
1.  newTrial( “experiment”,
2.  newText("The fish swim in a tank which is perfectly round")
3.  .print()
4.  ,
5.  newImage("two", "2fishRoundTank.png")
6.  .size(200,200)
7.  // .print()
8.  ,
9.  newImage("one", "1fishSquareTank.png")
10. .size(200,200)
11. // .print()
12. ,
13. newCanvas(450,200)
14. .add( 0 , 0 , getImage("two") )
15. .add( 250 , 0 , getImage("one") )
16. .print()
17. ,
18. newKey("FJ")
19. .wait()
)
```

Storing information

```
1. newKey("FJ")
2. .log()
3. .wait()
```

Initial Instructions

```
1. newTrial( “welcome”,
2. defaultText
3. .print()
4. ,
5. newText("<p>Welcome!</p>")
6. ,
7. newText("<p>In this experiment, you will have to report
   which of two pictures matches a description.</p>")
8. ,
9. newText("<p>Press the <strong>F</strong> key for the
   picture on the left, or the <strong>J</strong> key for the
   picture on the right.</p>")
10.,
11.newText("<p>Click the button below to start the
   experiment.</p>")
12.,
13.newButton("Start")
14..print()
15..wait()
16.)
```

Collecting participants ID

Add after instructions

```
1. newTextInput("inputID")
2. .print()
3. ,
4. newButton("Start")
5. .print()
6. .wait()
7. ,
8. newVar("ID")
9. .global()
10..set( getTextInput("inputID") )
11.)
12..log( "ID" , getVar("ID") )
```

Adding completion screen

```
1. SendResults(“send”)
2.
3.  ,
4. newTrial( “final”,
5. newText("<p>Thank you for your participation!</p>")
6. .print()
7. ,
8. newText("<p><a href='https://www.pcibex.net/'>Click here
   to validate your participation.</a></p>")
9. .print()
10.,
11.newButton("void")
12..wait()
13.)
```

More: Adding audio

```
1.newAudio("2fishRoundTank.mp3")
2..play()
3.,
4.newText("The fish swim in a tank which is
perfectly round")
5..print()
6.,
7.newImage("two", "2fishRoundTank.png")
8..size(200,200)
9.,
10.newImage("one", "1fishSquareTank.png")
11..size(200,200)
12.,
13.newCanvas(450,200)
14..add( 0 , 0 , getImage("two") )
15..add( 250 , 0 , getImage("one") )
16..print()
17.,
18.newKey("FJ")
19..log()
20..wait()
```

Stopping audio (if option has been selected)

```
1.newAudio("description",
"2fishRoundTank.mp3")
2..play()
3.,
4.newText("The fish swim in a tank which is
perfectly round")
5..print()
6.,
7.newImage("two", "2fishRoundTank.png")
8..size(200,200)
9.,
10.newImage("one", "1fishSquareTank.png")
11..size(200,200)
12.,
13.newCanvas(450,200)
14..add( 0 , 0 , getImage("two") )
15..add( 250 , 0 , getImage("one") )
16..print()
17.,
18.newKey("FJ")
19..log()
20..wait()
21.,
22.getAudio("description")
```

Selecting images

```
1.// newKey("FJ")
2.newSelector()
3..add( getImage("two") , getImage("one") )
4..keys( "F" , "J" )
5..log()
6..wait()
```

Adding trials

Now, we only have on trial. But our experiment consists of 100 trials. Do we have to copy every sentence? No! We just create a template.

```
Template( variable =>
  newTrial(
    newAudio("description", variable.AudioFile)
        .play()
    ,
    newText(variable.Description)
        .unfold(2600)
    ,
    newImage("two", variable.PluralImageFile)
        .size(200,200)
    ,
    newImage("one", variable.SingularImageFile)
        .size(200,200)
    ,
    newCanvas(450,200)
        .add(   0 , 0 , getImage("two") )
        .add( 250 , 0 , getImage("one") )
        .print()
    ,
    newSelector()
        .add( getImage("two") , getImage("one") )
        .keys(          "F"    ,          "J"   )
        .log()
        .wait()
    ,
    getAudio("description")
       .wait("first")
  )
  .log( "ID" , getVar("ID") )
)
```

Spreadsheet

- Go to Resources in the main project page
- comma-separated-value (CSV) format
- Group creates automatically a latin square design

	Treatment Order			
	1	2	3	4
Participant 1	A	B	D	C
Participant 2	B	C	A	D
Participant 3	C	D	B	A
Participant 4	D	A	C	B

AudioFile	Description	PluralImageFile	SingularImageFile	Item	Group	Ending	Duration
1fishSquareTank.mp3	The fish swims in a tank which is perfectly square	2fishRoundTank.png	1fishSquareTank.png	fish	A	-s	2600

Logging results

```
1.Template( variable =>
2.newTrial(
3.newAudio("description", variable.AudioFile)
4..play()
5.,
6.newText(variable.Description)
7..unfold(2600)
8.,
9.newImage("two", variable.PluralImageFile)
10..size(200,200)
11.,
12.newImage("one", variable.SingularImageFile)
13..size(200,200)
14.,
15.newCanvas(450,200)
16..add( 0 , 0 , getImage("two") )
17..add( 250 , 0 , getImage("one") )
18..print()
19.,
20.newSelector()
21..add( getImage("two") , getImage("one") )
22..keys( "F" , "J" )
23..log()
24..wait()
25.,
26.getAudio("description")
27..wait("first")
28.)
29.log( "ID" , getVar("ID") )
30.log( "Item" , variable.Item )
31.log( "Ending" , variable.Ending )
32.log( "Group" , variable.Group )
33.)
```

Adding a timer

- Adding a pause between trials

```
1.Template( variable =>
2.newTrial(
3.newTimer(500)
4..start()
5..wait()
6.,
7.newAudio("description", variable.AudioFile)
8..play()
9.,
10.newText(variable.Description)
11..unfold(2600)
12.,
13.newImage("two", variable.PluralImageFile)
14..size(200,200)
15.,
16.newImage("one", variable.SingularImageFile)
17..size(200,200)
18.,
19.newCanvas(450,200)
20..add( 0 , 0 , getImage("two") )
21..add( 250 , 0 , getImage("one") )
22..print()
23.,
```

```
1.newSelector()
2..add( getImage("two") , getImage("one") )
3..keys( "F" , "J" )
4..log()
5..wait()
6.,
7.getAudio("description")
8..wait("first")
9.,
10.newTimer(500)
11..start()
12..wait()
13.)
14..log( "ID" , getVar("ID") )
15..log( "Item" , variable.Item )
16..log( "Ending" , variable.Ending )
17..log( "Group" , variable.Group )
18.)
```

Initial command

The initial command establishes the sequence in which the elements will appear and the randomization. Note that the initial sequence requires that the trials are named:

newTrial("welcome" ,
defaultText
.print()

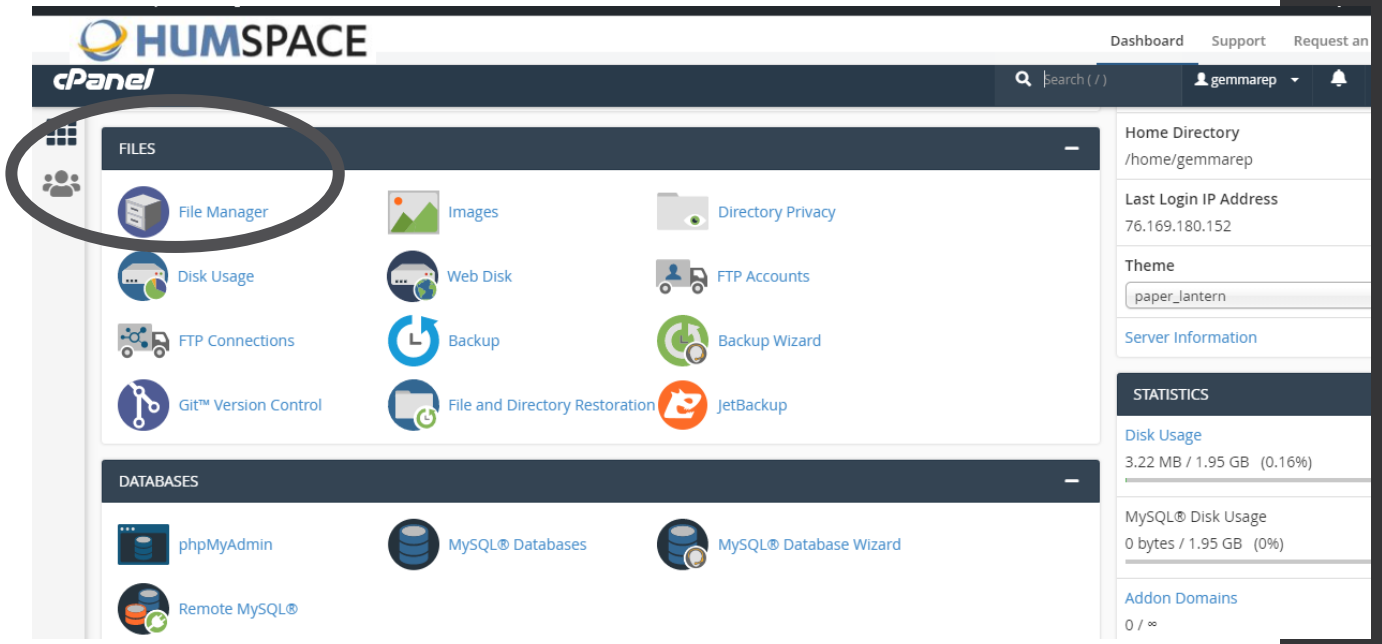
```
PennController.ResetPrefix(null);
Sequence( "welcome" , randomize("experiment") , "send" , "final" )
```

Server Set up

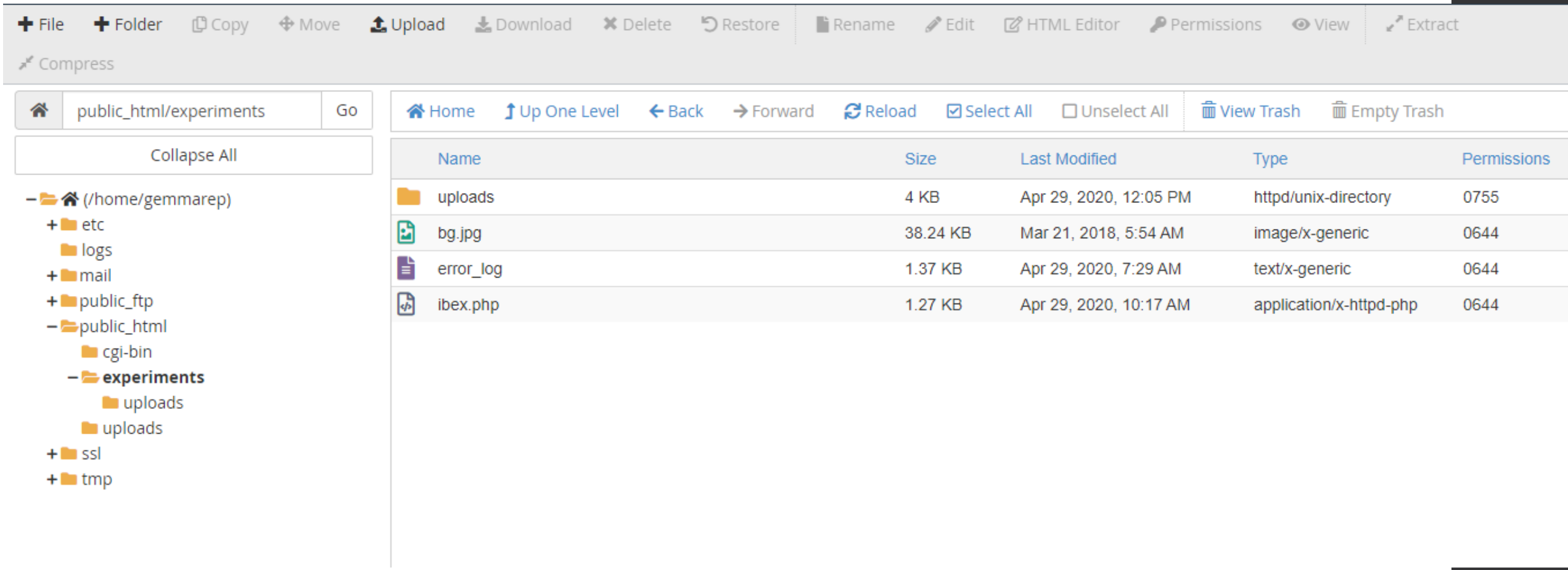
- Create an account at HumSpace
- Go to file manager
- Create a folder: experiments
- Upload the audio files
- Copy path
- My path is:

https://gemmarepisopu.humspace.ucla.edu/experiments/

Yours will be similar!



- Go to public_html
- Create a folder experiments. Inside experiment -> uploads
- Copy ibex.php inside experiments (NOT INSIDE UPLOADS!)



Collecting audio recordings

```
PennController.ResetPrefix(null); // Initiates PennController

PennController.InitiateRecorder("https://gemmarepisopu.humspace.ucla.edu/experiments/ibex.php")

.label("init")


  Template( variable =>
  newTrial(

  newTimer(500)
        .start()
        .wait()
  ,
  newVoiceRecorder("recorder")
        .print()
  ,
```

¡Y esto es todo!

https://www.pcibex.net/

Support: https://www.pcibex.net/forums/
(you will get an answer the same day!)

Don't forget to cite them:

Zehr, J., & Schwarz, F. (2018). PennController for Internet Based Experiments (IBEX). https://doi.org/10.17605/OSF.IO/MD832