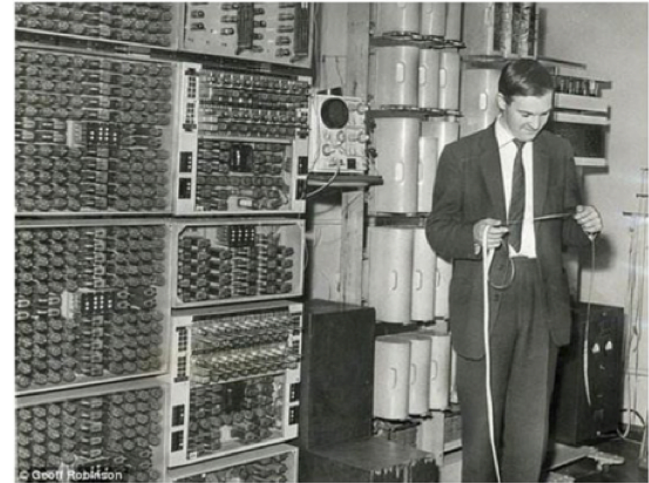
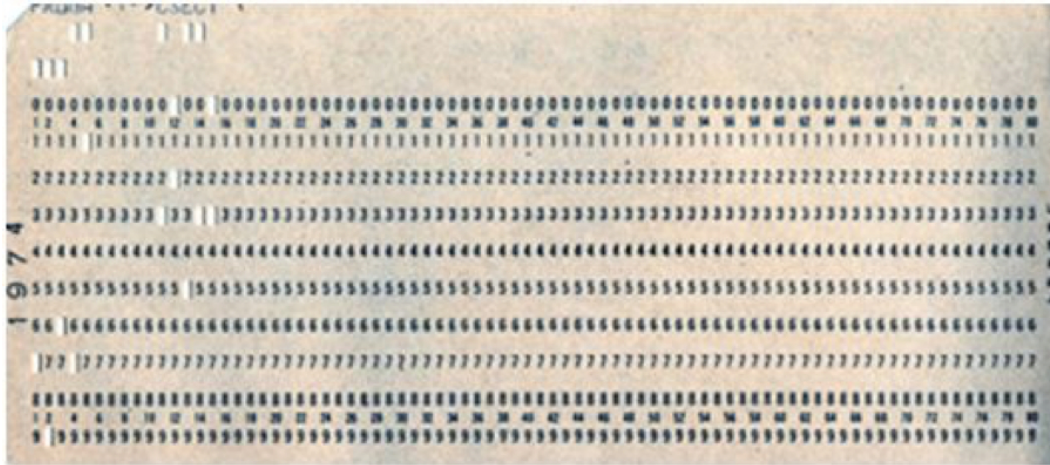


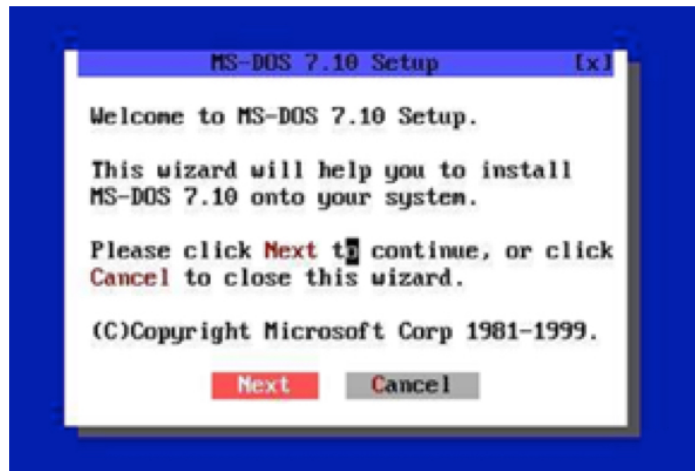
Animal interaction

Xinyue Yu

Interaction history



The earliest human-computer interaction is the cassette type.
Eniac (1943)



In the mid-1960s, the command line interface appeared, and the way of human-computer interaction became a question-and-answer style. The programmer inputs instructions through the command line interface. After the computer receives the instructions, it sends feedback to the programmer through a single character.

For example: Drawing Machine in 1960s

Ivan Sutherland's SketchPad-1963 PhD

Sophisticated drawing package

introduced many ideas/concepts now found in today's interfaces

- **hierarchical structures** defined pictures and sub-pictures
- **object-oriented programming**: master picture with instances
- **constraints**: specify details which the system maintains through changes
- **icons**: small pictures that represented more complex items
- **copying**: both pictures and constraints
- **input techniques**: efficient use of light pen
- **world coordinates**: separation of screen from drawing coordinates
- **recursive operations**: applied to children of hierarchical objects



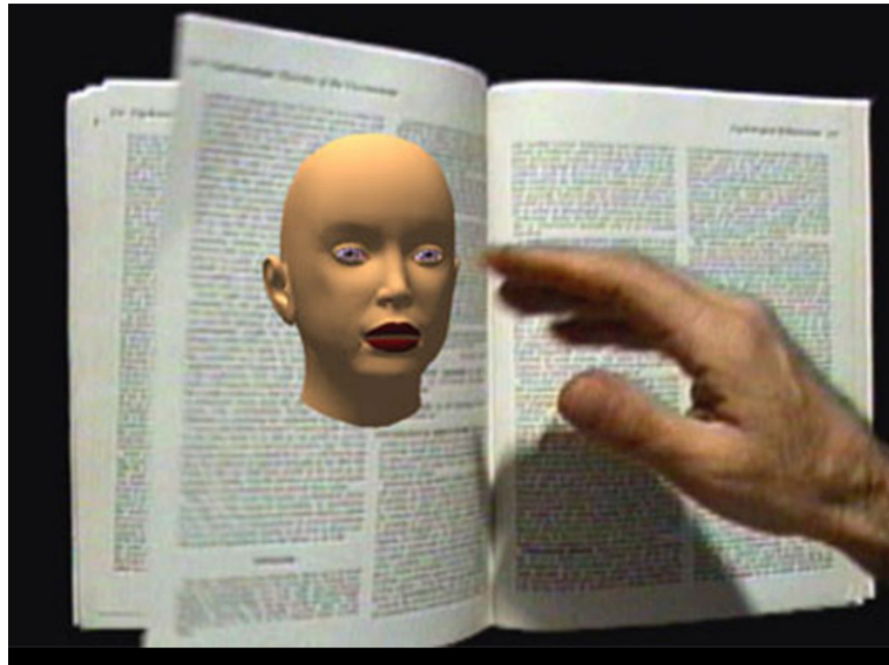
Now:

graphical user interface interaction

In the stage, the computer is still input and output through the hands and eyes, but the computer receives the input of the user in many forms like touching the screen; and the forms of output to the user are also more and more, like chart/image feedback.

Still developing:

1. Gesture tracking
2. Eye tracking
3. Voice control.
4. Interactive touch
5. Natural interaction, brain-computer interface
6. Body interaction



"Pressure to Speak (House of Cards)" 2002

My project

The theme of my project is to make a **visual garden that interacts with cats**, which I call the Cat Garden. A beautiful garden existing in a picture frame is projected onto the drab interior wall . And when the **cat walks** into the painting and **touches** the garden image, through the **real-time cat recognition and location tracking technology**, the surveillance camera will calculate the coordinates of the wall location that the cat touches and transmit the **data** to the interactive animation software.

Through the animation software, a virtual flower will be projected into the garden, **sprouting and growing at the location touched by the cat.** The whole growth process of the flower will be represented by stop-motion animation, and finally the flower will become part of the cat garden. In addition to flowers, many things, such as human eyes, butterflies, etc will also appear in this garden, appearing and changing with the cat's touch. All interaction between the cat and the garden will be recorded as the final presentation of the project. I used this whole installation to **build a cat garden in cat's mind.**

My inspiration comes from the situation of cats in different countries. Cats raised in China cannot go outside freely like German cats. They stay in the room for their whole life. When I went to study in Germany, my neighbor asked me why I didn't put the cats in the garden downstairs. I was very surprised because I had never think about such thing in China and I was afraid that the cat would run away. But my cats seem to really like the small garden downstairs. I try to put them in the cat bag and put them in the garden. They make satisfying sounds in the cat bag, so I want to move the garden downstairs to my appartment. Then cats can have a small garden that they can create by themselves in the room.

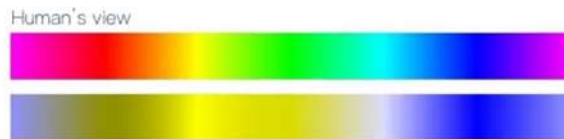


Some details about my project



Human View
(No UV Sensitivity)

Dog View
(Some UV Sensitivity)



Cat's view



Human's view 120°

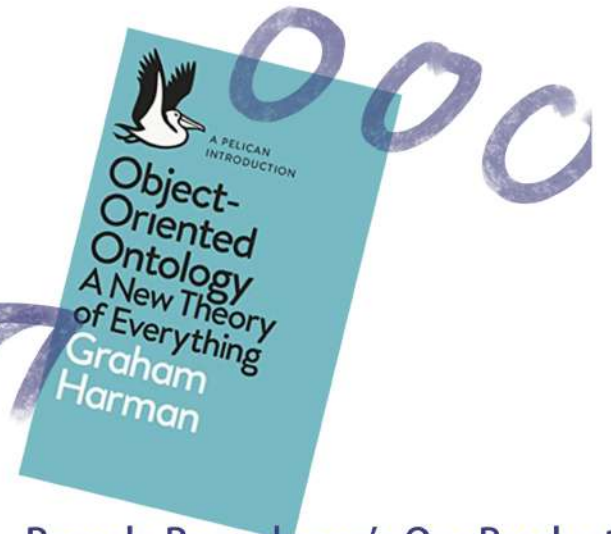


Cat's view 200°



Color

My idea



Pamela Rosenkranz's Our Product



In order to make this installation, first of all I had to understand the vision of cats. I read a lot of research from livescience about cats. In a 2014 article by studying the eyes of a large number of dead animals, I found that cats and dogs, as well as many other mammals, may be able to see ultraviolet light. This means that when our cat is staring at a certain place in the house, there may not be any ghosts there, but because there is a strange light frequency world that you can't see. During the day, cats' eyesight is not so good, but at night it is different. The retina of cat's eyes is not larger than that of humans, but compared with human eyes, they can collect and perceive abnormally low amounts of light.

Cats have a very superficial understanding of color, but they are definitely not color blind. They can understand blue, green, purple, and yellow. And their dynamic vision and night vision capabilities are very strong

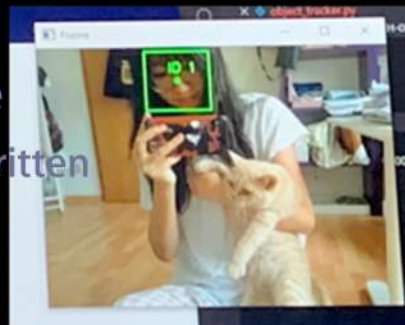
Background with the color of cat's view



I drew the background of the garden from the perspective of cats, and I used the colors which the cats can see.

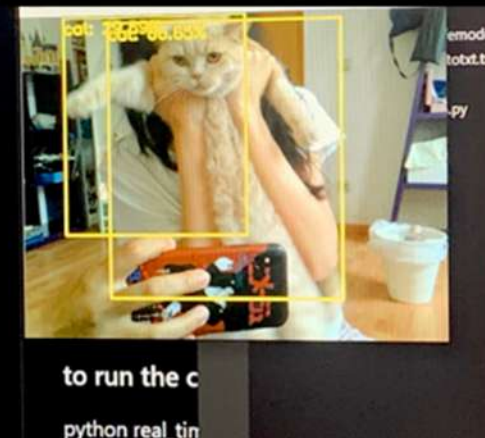


Then I had to work on cat recognition and real-time tracking technology. I used a camera to detect the target wall area in real time, and if the cat was very close to or touching the wall, the cat's movement was detected and information about the position of its head or paw was fed into the computer and used to generate an interactive animation. I used a deep neural network to identify the cat in the live video and track it in real time using object-tracking technology to output the location data. The whole software was written in python.



The tracking program using the neural network for detecting human faces

Human



The tracking program using the neural network for detecting cats

Cat

EXPLORER

- garden.png
- game.py
- real_time_object_detection.py
- data.json

game.py

```
1 import pygame, sys, json, time, os, re
2 from json.decoder import JSONDecodeError
3
4 def zoom_func(image, num):
5     zoom = num
6     wnd_w, wnd_h = image.get_size()
7     return (round(wnd_w/zoom), round(wnd_h/zoom))
8
9 def sorted_alphanumeric(data):
10     convert = lambda text: int(text) if text.isdigit() else text.lower()
11     alphanum_key = lambda key: [convert(c) for c in re.split('([0-9]+)', key)]
12     return sorted(data, key=alphanum_key)
13
14 def append_sprite(filepath):
15     dirs = sorted_alphanumeric(os.listdir(filepath))
16     try:
17         dirs.remove('.DS_Store')
18     except ValueError:
19         pass
20     dirs.reverse()
21     dirs = dirs + [dirs[-1]]*100
22     sprites = []
23     for i in dirs:
24         image = pygame.image.load(filepath + str(i))
25         sprites.append(pygame.transform.rotozoom(image, 0, 0.14))
26     return sprites
27
28
29 class Player(pygame.sprite.Sprite):
30     def __init__(self, pos_x, pos_y, image_folder, xia = True):
31         super().__init__()
32         self.appear_animation = False
33         self.pause = False
```

TERMINAL

```
found it
found it
found it
found it
found it
found it
File "real_time_object_detection.py", line 62, in <module>
  detections = net.forward()
KeyboardInterrupt
roz:Real-Time-Object-Detection-master roz
```

object tracking

game.py

```
66 self.image = pygame.transform.rotozoom(self.image, 0, 0.14)
67 self.rect = self.image.get_rect()
68 self.rect.bottomLeft = (pos_x, pos_y)
69 self.i = 0
70 self.image.set_alpha(self.i)
71 self.j = 0
72
73 def appear(self):
74     self.appear_animation = True
75
76 def update(self, speed):
77     if self.appear_animation == True:
78         self.i += speed*2
79         if self.i >= 255:
80             self.i = 255
81             self.appear_animation = False
82             self.pause = True
83             self.t0 = time.time()
84         else:
85             self.i = 255
86             self.image.set_alpha(self.i)
87
88 # General setup
89 pygame.init()
90 clock = pygame.time.Clock()
91
92 # Game Screen
93 screen = pygame.display.set_mode((int(1281*0.14), int(472*0.14)), 0, 0, pygame.FULLSCREEN)
94 background_image = pygame.image.load("garden.png").convert()
95 w1, h1 = background_image.get_size()
96 background_image = pygame.transform.scale(background_image, (int(w1*0.14), int(h1*0.14)))
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

TERMINAL

```
found it
found it
found it
found it
found it
found it
File "real_time_object_detection.py", line 62, in <module>
  detections = net.forward()
KeyboardInterrupt
roz:Real-Time-Object-Detection-master roz
```

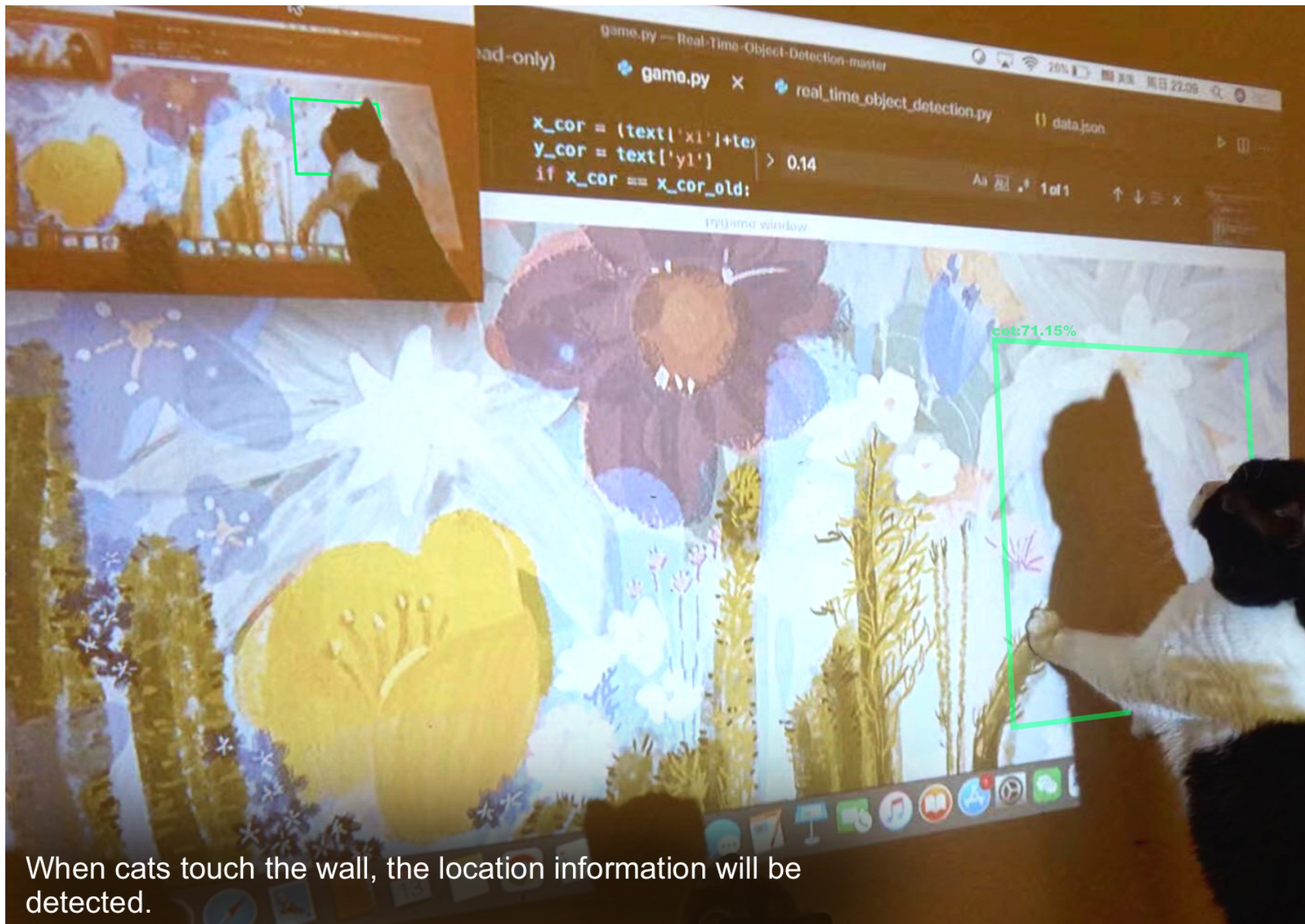
interactive animation

game.py

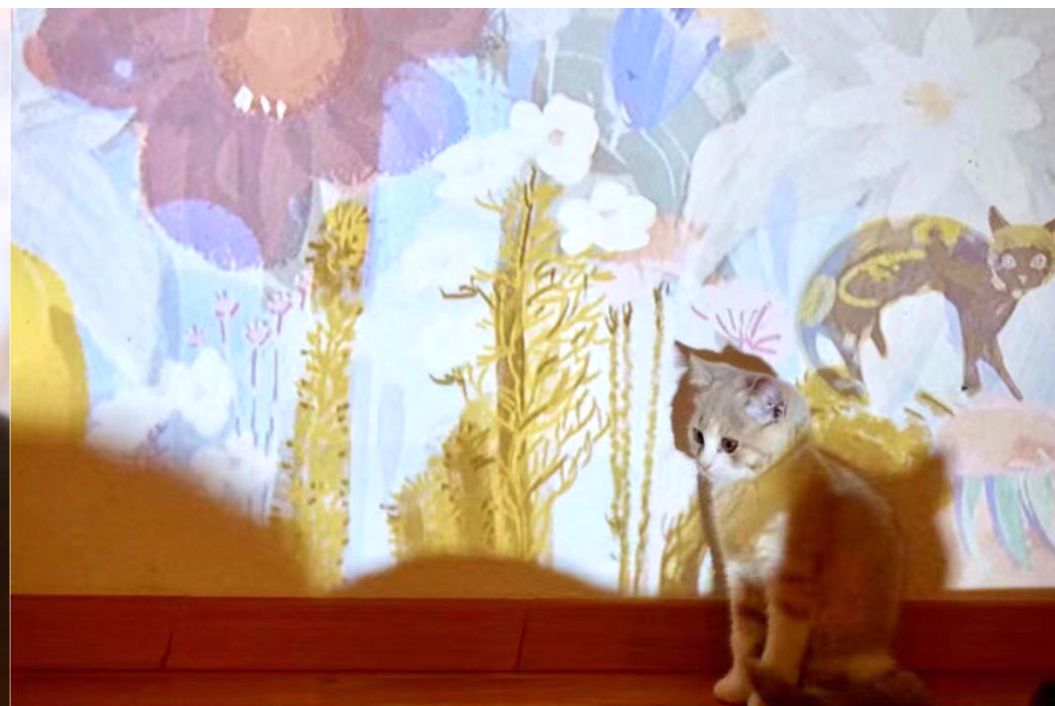
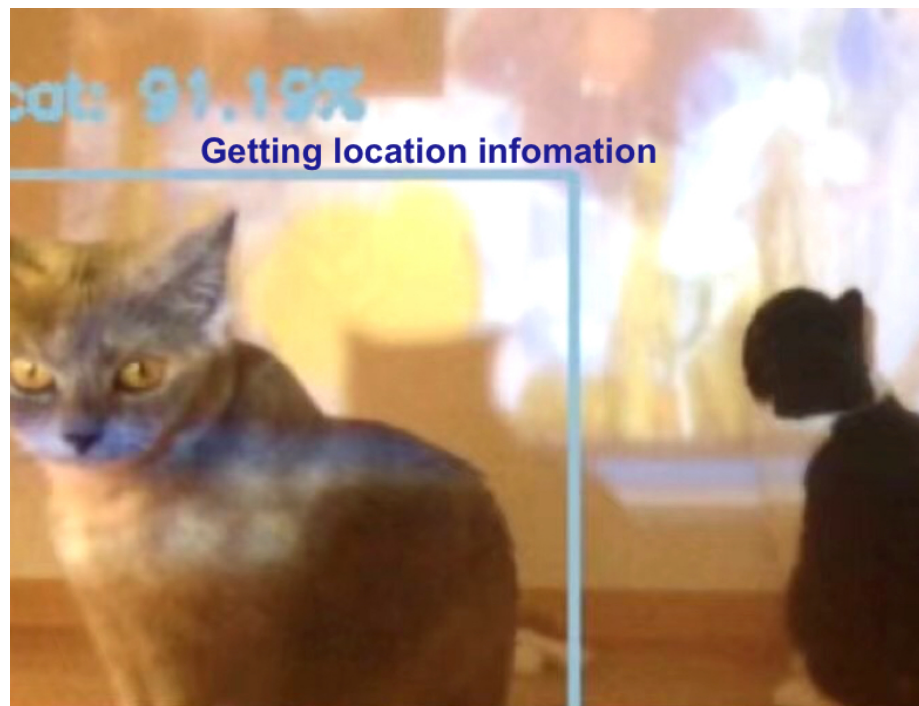
```
19 except ValueError:
20     pass
21     dirs.reverse()
22     dirs = dirs + [dirs[-1]]*100
23     sprites = []
24     for i in dirs:
25         image = pygame.image.load(filepath + str(i))
26         sprites.append(pygame.transform.rotozoom(image, 0, 0.14))
27     return sprites
28
29
30 class Player(pygame.sprite.Sprite):
31     def __init__(self, pos_x, pos_y, image_folder, xia = True):
32         super().__init__()
33         self.appear_animation = False
34         self.pause = False
35         self.t0 = None
36         self.sprites = append_sprite(image_folder)
37         self.current_sprite = 0
38         self.image = self.sprites[self.current_sprite]
39         self.rect = self.image.get_rect()
40         self.xia = xia
41         if self.xia == True:
42             self.rect.bottomLeft = (pos_x, pos_y)
43         else:
44             self.rect.topLeft = (pos_x, pos_y)
45
46 def appear(self):
47     self.appear_animation = True
48
49 def update(self, speed):
50     if self.appear_animation == True:
51         self.current_sprite += speed
```

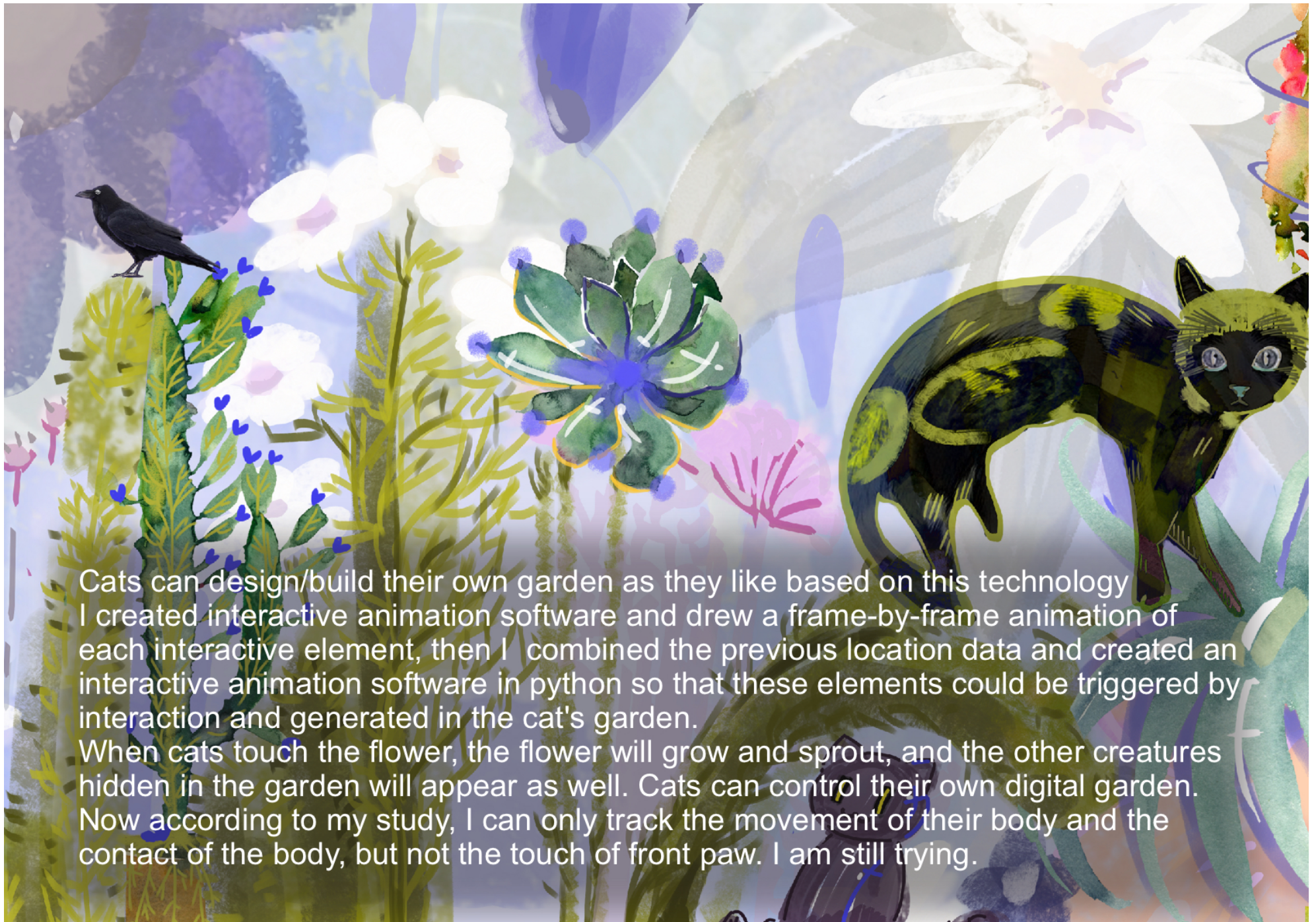
TERMINAL

```
found it
found it
found it
found it
found it
found it
File "real_time_object_detection.py", line 62, in <module>
  detections = net.forward()
KeyboardInterrupt
roz:Real-Time-Object-Detection-master roz
```



When cats touch the wall, the location information will be detected.





Cats can design/build their own garden as they like based on this technology. I created interactive animation software and drew a frame-by-frame animation of each interactive element, then I combined the previous location data and created an interactive animation software in python so that these elements could be triggered by interaction and generated in the cat's garden. When cats touch the flower, the flower will grow and sprout, and the other creatures hidden in the garden will appear as well. Cats can control their own digital garden. Now according to my study, I can only track the movement of their body and the contact of the body, but not the touch of front paw. I am still trying.

Problem

The first thing was that I wanted to do too much interactive design, but I **drew too slowly**. This was the most troublesome thing for me.

on the other hand, In the process of proceeding, I encountered technical difficulties when passing real-time tracking location data to the interactive animation program. This is because the flowers in the garden needed the cat to actually interact with the wall projection before they could bloom. To achieve such an effect, it was

necessary to control the **video detection range**.

For example

- If the cat could be detected when it was lying on the ground or not close to the wall, the interaction was lost (too easy to be detected)
- The cat could only be detected when most of its body appeared, so the detection range could not be too small. Because if only the head/paw that entered the projection was detected, the detector could not determine that the target object was a cat. Only when most of the cat's body was seen, the system could automatically identify the target as a cat and thus output the position data. So body parts that did not enter the projected wall also needed to be detected.

So the lowest position of the camera detection area was experimentally determined to be X cm above the ground.

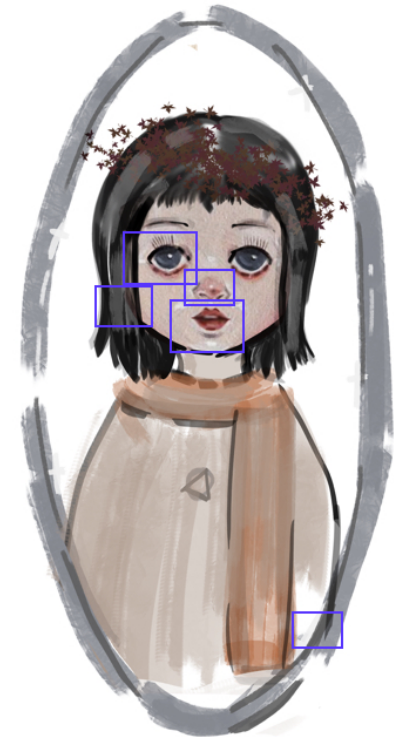
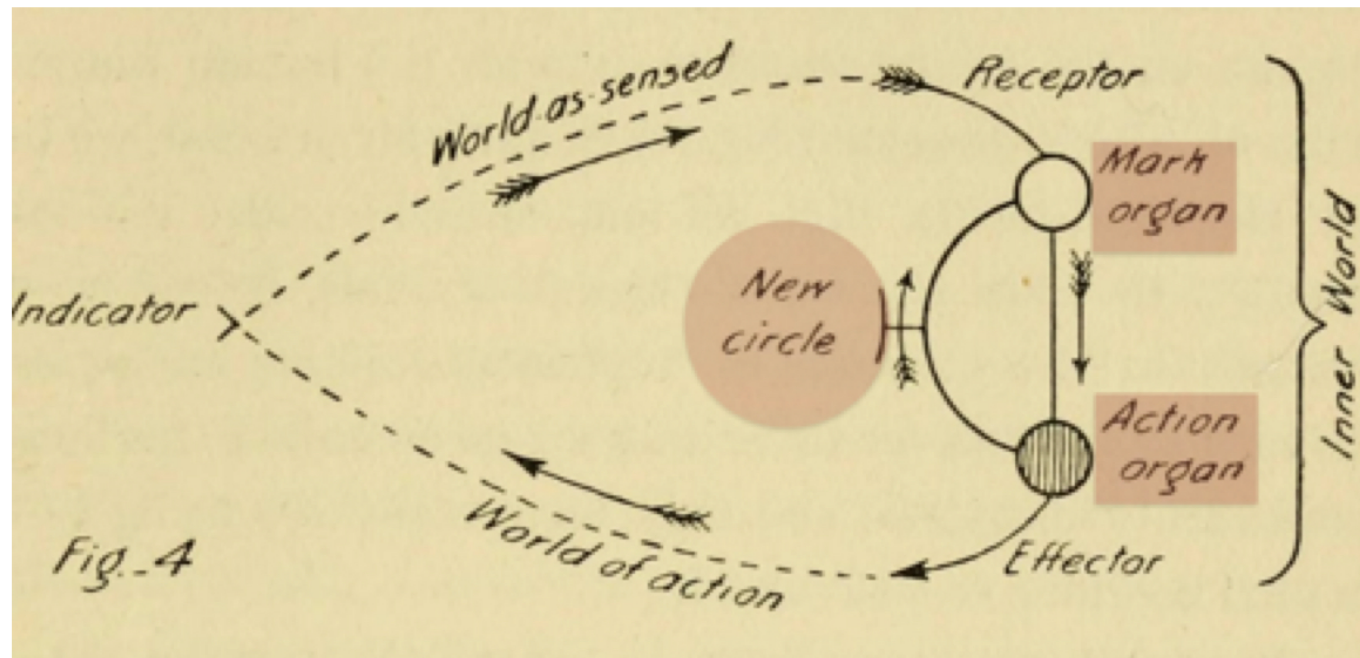
To choose the library for programming is difficult. After a lot of trying I choose OpenCV for detecting and Pygame for interactive animation.



Interactive Plant Growing an interactive computer installation

1992, Christa Sommerer & Laurent Mignonneau

Explanation of cats interaction with Uexkülls Biosemantic Feedback Circle



Later the development of my installation is according to the senses of cats. From the feedback circle we find out that the inner world of cats is similar to a lot of animals.

The action organs of cats are tails, legs, and mouths.

The mark organs are eyes, ears, noses, tongues, skin and beards.

The key element for building the new circle is: The movement of legs and tails will allocate the new position of cats and get new sense of touch; beards will feel the new position, and the skin will feel the stimulus from the outside world; the tongues will taste new flavors. Depending of these feelings their action organs will give reaction again.