

# Appendix 5: Interview Questions

Questions for Artists	408
Questions for Programmers	409

## Questions for Artists

1. Background/History
  - a. How long have you been making art?
  - b. Why do you like doing it?
  - c. How long have you been using computers?
  - d. How long have you been making artwork using computers?
  - e. What first made you interested in using computers for art?
  - f. Why are computers important in art?
  - g. How much do you think you understand the potential (technological) capabilities of computers?
  - h. Do you program?
    - i. What sort of programs are they?
    - ii. Copied from books? Did you understand them? When do you think you reached the stage where you feel you could figure out how to understand any given program?

2. How many of your art projects have used computers?
  - a. At what stage of your process do you think “hey, we could use a computer for this?”
  - b. Could you give me a list, off the top of your head?
  - c. Did you work with a programmer for all of these?
  - d. Favourites or not favourites? Why?
  - e. different models of collaboration: equal partnership, assistant, teacher, contracted consultant—which role do you prefer your technology collaborators to take? Why?
3. What’s the hardest thing about working with programmers?
  - a. Could you describe in your own words the process you go through with the programmer to create the art?
  - b. How interested are you in the programming process?
  - c. How involved do you become in the programming process? Is this enough, or too much? Can you think of anything that might make the right amount of involvement easier?
  - d. Do you have a clear idea in your imagination of what you expect the finished work to look like? (if no:) How do you decide when it’s “finished”? (experimentation) If you don’t program, how do you experiment with the program? Does the programmer make special controls for you?
4. Do you think the programmer is able to influence the art or you as an artist?  
When? In what ways?
  - b. What do you think about that?

## Questions for Programmers

1. Background/History
  - a. How long have you been using computers? What made you interested in them?

- b. When did you start to program?
    - i. What sort of programs were they?
    - ii. Copied from books? Did you understand them? When do you think you reached the stage where you feel you could figure out how to understand any given program?
  - c. Why do you like programming?
  - d. What languages are you most comfortable in? i. Why did you learn those?
2. How many art projects have you programmed?
- a. Could you give me a list, off the top of your head?
  - b. Favourites or not favourites?
  - c. What motivates?
  - d. Candy/Mamykina paper, different models of collaboration—which role do you see yourself taking?
3. Collaborating with other programmers
- a. What kind?
  - b. Did you stick to different tasks or both work on the same tasks?
    - i. Did you use the same language?

The term ‘Artists’ in the following is too general. So choose one or several specific examples if you think they’re useful.

4. What’s the hardest thing about programming for artists?
- a. How do you choose tools?
  - b. Broadly describe your development process?
  - c. Do you show them what the problem is?

- d. Have you tried to show artists the code? Did they understand it?
    - i. Have they ever changed it?
  - e. Do you find that artists understand what you do? Are they interested?
  - f. Do you add things to the program to make it easier for the artists to engage with the program?
  - g. Is there anything you think that the tools could do to help support you communicating with artist?
  - h. Do you use conventional software development processes to make art? (Like documentation, UML diagrams)
    - i. Might you in future? When?
5. Do you think you are able to influence the art or artist as a programmer?
- a. When would you? In what ways?
  - b. What does the artist think?
6. If working with artist who doesn't really understand the tools or the power general-purpose computing, how might you convey the idea?
- a. Demos? Diagrams
  - b. Have you ever tried? Do you think it worked?
7. Open-ended discussion: When I learnt programming, I had a kind of object-oriented 'epiphany', where I got my head round the things you could do with OO that you could not with procedural, and it's kind of a mathematical, abstract way of looking at things. So some artists exploit that; some are kind of stuck in the 'if a, then b' mindset—it's hard to nail down though. Do you think it helps in art?