

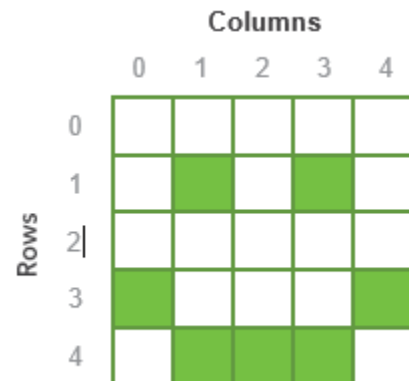
Colour By Pixel

Would you enjoy playing games on a computer that didn't have a screen with graphics? Probably not. Computer graphics and animation have become an important part of how we interact with computers today.

Computer programs are like lists of instructions that we give to a computer. Computers are not very capable on their own -- they need good instructions to tell them what to do and how to do it. Programmers can create instructions for computers by writing computer code that tells the computer exactly what to do.

But how do programs draw pictures onto the screen? Computers break each picture down into tiny blocks of color, called pixels, that are organized in a grid across the screen. Each pixel contains one single color. By combining thousands and thousands of these tiny pixels, programs can draw complicated pictures onto the screen.

The image to the right is a very simple example of using pixels to draw a basic image. Each box in the grid is a single pixel. In this case, some of the pixels have been colored to make a smiley face. We can label the rows and columns of the image to make it easier to pick out specific pixels:



Now that we have row and column numbers, we can select individual pixels. For example, the smiley face's right eye is located in column 3, row 1 (notice that we start counting at zero). We can communicate this more quickly by writing all the pixel coordinates in this format: (column, row).

We'll write the color of the pixel rightly after the column and row position. So to color the right eye, we can write: **(3, 1, green)**.

Using these coordinates, we can paint the smiley face above with the following program: **1. (1, 1, green)**

2. (3, 1, green)

3. (0, 3, green)

4. (4, 3, green)

5. (1, 4, green)

6. (2, 4, green)

7. (3, 4, green)



	0	1	2	3	4
0					
1					
2					
3					
4					

Program Instructions

- 1. (0, 0, yellow)
- 2. (2, 0, yellow)
- 3. (4, 0, yellow)
- 4. (0, 1, yellow)
- 5. (2, 1, yellow)
- 6. (4, 1, yellow)
- 7. (0, 2, yellow)
- 8. (1, 2, yellow)
- 9. (2, 2, yellow)
- 10. (4, 2, yellow)
- 11. (0, 3, yellow)
- 12. (2, 3, yellow)
- 13. (4, 3, yellow)
- 14. (0, 4, yellow)
- 15. (2, 4, yellow)
- 16. (4, 4, yellow)



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0															
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															

PROGRAM INSTRUCTIONS

- | | | | |
|--------------------|--------------------|--------------------|--------------------|
| 1. (3, 1, black) | 17. (5, 3, red) | 33. (2, 5, red) | 50. (9, 6, red) |
| 2. (4, 1, black) | 18. (6, 3, black) | 34. (3, 5, red) | 51. (10, 6, black) |
| 3. (8, 1, black) | 19. (7, 3, red) | 35. (4, 5, red) | 52. (3, 7, black) |
| 4. (9, 1, black) | 20. (10, 3, red) | 36. (5, 5, red) | 53. (4, 7, red) |
| 5. (2, 2, black) | 21. (11, 3, black) | 37. (6, 5, red) | 54. (5, 7, red) |
| 6. (3, 2, red) | 22. (1, 4, black) | 38. (7, 5, red) | 55. (6, 7, red) |
| 7. (4, 2, red) | 23. (2, 4, red) | 39. (8, 5, red) | 56. (7, 7, red) |
| 8. (5, 2, black) | 24. (3, 4, red) | 40. (9, 5, red) | 57. (8, 7, red) |
| 9. (7, 2, black) | 25. (4, 4, red) | 41. (10, 5, red) | 58. (9, 7, black) |
| 10. (8, 2, red) | 26. (5, 4, red) | 42. (11, 5, black) | 59. (4, 8, black) |
| 11. (9, 2, red) | 27. (6, 4, red) | 43. (2, 6, black) | 60. (5, 8, red) |
| 12. (10, 2, black) | 28. (7, 4, red) | 44. (3, 6, red) | 61. (6, 8, red) |
| 13. (1, 3, black) | 29. (8, 4, red) | 45. (4, 6, red) | 62. (7, 8, red) |
| 14. (2, 3, red) | 30. (10, 4, red) | 46. (5, 6, red) | 63. (8, 8, black) |
| 15. (3, 3, red) | 31. (11, 4, black) | 47. (6, 6, red) | 64. (5, 9, black) |
| 16. (4, 3, red) | 32. (1, 5, black) | 48. (7, 6, red) | 65. (6, 9, red) |
| | | 49. (8, 6, red) | 66. (7, 9, black) |
| | | | 67. (6, 10, black) |

