

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <windows.h> //for void hidecursor(), only required on windows

// "kitten printers"

int in = 0;
int objects = 30; //0-29 are objects, 30 is kitten, 31 is robot
int x[31]; int y[31]; //int x[(objects + 2)]; int y[(objects + 2)]; the + 2 makes
room for kitten and robot, but these should be excluded from for loops
char object_symbol[31]; //char object_symbol[(objects + 2)];
int object_color[31]; //int object_color[(objects + 2)];
int object_brightness[31]; //int object_brightness[(objects + 2)];
int object_text[29]; //int object_text[(objects + 2)]; //kitten doesnt have text
because a cutscene plays and robot doesnt have text cause you cant colide with it
int populate_stop = 1; int addrobot_stop = 1; int addkitten_stop = 1; int move_stop
= 1; int text_stop = 1; //stop variables
int texts = 64; char text[64][128] = { //the text for colliding with nkos, might
want to minimize this because it will get big
    {"Not kitten."},
    {"Not a kitten."},
    {"A pigeon with a TCP/IP packet taped to it's leg."}, //i
like this line too much not to use it
    {"Deez."},
    {"The Internet, lying in a puddle on the floor."},
    {"Your mother's house."},
    {"What is it? I don't know."},
    {"That one thing. You know what thing I mean."},
    {"A magazine full of bullets... it's not the right type of
magazine, though."},
    {"Someone spilled their drink here!"},
    {"Your Advanced Placement Computer Science Principles
classroom."},
    {"text 11"},
    {"This brick wall is missing a brick! Oh, nevermind, it's
just been replaced with a Nokia 3310."},
    {"A kitten."},
    {"A penguin using an operating system you've never seen
before."},
    {"An open window shining on a computer with an entirely
blue screen."}, //objectively bad //16
    {"This white apple is hooked up to a mouse, keyboard and
monitor... it isn't doing anything."},
    {"A pufferfish using an operating system you've never seen
before... the pufferfish is puffed up."},
    {"A copy of 'The C Programming Language'."},
    {"George (the cat)."},
    {"An analog clock running on UNIX time."},
    {"The table that you always hit your knee on."},
    {"A calendar with some classic cars on it for 2022."},
    {"Justice, finally!"},
    {"A newspaper dated December 6th, 1941... it seems like it
was a good day."},
    {"Poland."},
    {"An old map of Europe... it looks more like pasta than
borders."},
    {"The RWin key."},
    {"It appears to be a... oh, nevermind."},

```



```

        {"      0--0--0,      "},
};
time_t t;

void hidecursor() { //i hate windows, and you do too
    HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_CURSOR_INFO info;
    info.dwSize = 100;
    info.bVisible = FALSE;
    SetConsoleCursorInfo(consoleHandle, &info);
}

void populate() {
    for(int i = 0; i < objects; i++) { //initial position randomization
        x[i] = (rand() % 128);
        y[i] = (rand() % 64);
        object_symbol[i] = ((rand() % 93) + 32); //ansi 33-126
        if (rand() % 2) { //picks between the top or bottom 8 colors
            object_color[i] = ((rand() % 8) + 31); //color 31-38, 30 is excluded
because the background is black and the character would be invisible
        } else {
            object_color[i] = ((rand() % 8) + 90); //color 90-98
        }
        object_text[i] = ((rand() % texts)); //text 0-127
    }
    while(populate_stop) { //check if multiple positions are the same, if so then
re-randomize those positions
        for(int i = 0; i < objects; i++) {
            for(int j = 0; j < objects; j++) {
                if((x[i] == x[j]) && (y[i] == y[j])) {
                    x[i] = (rand() % 128);
                    y[i] = (rand() % 64);
                } else {
                    populate_stop = 0;
                }
            }
        }
    }
    while(text_stop) {
        for(int i = 0; i < objects; i++) {
            for(int j = 0; j < objects; j++) {
                if(object_text[i] == object_text[j]) {
                    object_text[i] = ((rand() % texts));
                } else {
                    text_stop = 0;
                }
            }
        }
    }
    object_text[30] = 0; //kitten always gets 2 for some reason so im making it 0
manually for niceness
    for (int i = 0; i < objects; i++) {
        printf("\033[%d;%dH\033[%dm%c\033[0m", (y[i] + 3), (x[i] +
1), object_color[i], object_symbol[i]);
    }
}

void addkitten() {
    x[30] = (rand() % 128);

```

```

y[30] = (rand() % 64);
object_symbol[30] = ((rand() % 93) + 32); //ansi 33-126
if (rand() % 2) { //picks between the top or bottom 8 colors
    object_color[30] = ((rand() % 8) + 31); //color 31-38, 30 is excluded
because the background is black and the character would be invisible
} else {
    object_color[30] = ((rand() % 8) + 90); //color 90-98
}
while(addkitten_stop) { //check if multiple positions are the same, if so then
re-randomize those positions
    for(int i = 0; i < objects; i++) {
        for(int j = 0; j < objects; j++) {
            if((x[30] == x[j]) && (y[30] == y[j])) {
                x[30] = (rand() % 128);
                y[30] = (rand() % 64);
            } else {
                addkitten_stop = 0;
            }
        }
    }
}
//the kitten printer
printf("\033[%d;%dH\033[%dm%c\033[0m", (y[30] + 3), (x[30] +
1), object_color[30], object_symbol[30]);
}

void addrobot() {
    x[31] = (rand() % 128);
    y[31] = (rand() % 64);
    object_symbol[31] = 35; //robot is always a bright white #
    object_color[31] = 97;
    while(addrobot_stop) {
        for(int i = 0; i < objects; i++) {
            for(int j = 0; j < objects; j++) {
                if((x[31] == x[j]) && (y[31] == y[j])) {
                    x[31] = (rand() % 128);
                    y[31] = (rand() % 64);
                } else {
                    addrobot_stop = 0;
                }
            }
        }
    }
    printf("\033[%d;%dH\033[%dm%c\033[0m", (y[31] + 3), (x[31] + 1), object_color[31],
(object_symbol[31]));
}

void initialize() {
    hidecursor(); //hides cursor on windows
    srand(time(t));
    system("mode con: cols=128 lines=66");
    printf("\033[0;rfk\x1b\x5c"); //set conlse title to rfk
    printf("\033[8;40HYou are Robot, and your goal is to find Kitten.\033[9;36HThis
is complicated by the exsistance of other objects.\033[10;42HYou must touch an
object to see what it is.\033[12;39HUse the arrow keys and press any key to
continue.");
    getch();
    system("cls");
    printf("\

```

```

033[2;1H-----
-----");
    populate();
    addkitten();
    addrobot();
}

void kitscene() { //kitten cutscene, kitscene - if you will
    system("cls");
    printf("\
033[2;1H-----
-----");
    for (int i = 0; i < 39; i++) {
        printf("\033[28;%dH", (8 + i)); //kitten section
        for (int j = 0; j < 7; j++) { //kitsceen is 7 characters tall
            for (int l = 0; l < 21; l++) { //kitsceen is 19 characters long,
including \0 //maybe not anymroe
                switch(kitscenetext[j][l]) {
                    case '\ ':
                        printf("\033[33m");
                        break;
                    case '. ':
                        printf("\033[32m");
                        break;
                    case ', ':
                        printf("\033[0m");
                        break;
                    default:
                        printf("%c", kitscenetext[j][l]);
                        break;
                }
            }
        }
        printf("\033[18D \033[B");
    }
    printf("\033[28;%dH", (103 - i)); //robot section
    for (int j = 0; j < 7; j++) { //kitsceen is 7 characters tall
        for (int l = 0; l < 19; l++) { //kitsceen is 19 characters long,
including \0
            switch(kitscenerobot[j][l]) {
                case '\ ':
                    printf("\033[90m");
                    break;
                case '. ':
                    printf("\033[31m");
                    break;
                case ', ':
                    printf("\033[0m");
                    break;
                default:
                    printf("%c", kitscenerobot[j][l]);
                    break;
            }
        }
    }
    printf("\033[18D\033[B");
}
    Sleep(50); //windows only, fix cross-platfrom with defiens
}

printf("\033[1;1H

```

```

\033[1;1H"); //clear line manually because windwos termina lis crap
printf("\033[1;1HAgain? [y/n]");
sleep(1);
int exitchoice = getch();
while(1) { //the stuff inthe while loop make it so only y and n work, instead
of arrow keys and such working too      probably not the best wayof doing it but its
fine for v1
    if(exitchoice == 110) {
        printf("\033[1;1H
\033[1;1H"); //clear line manually because windwos termina lis crap
        printf("\033[1;1HYeah, cause DOS is so much better...");
        getch();
        exit(0);
    } else if(exitchoice == 121) {
        initialize();
        return;
    }
    exitchoice = getch();
}

}

void objectcollidetext(int i) {
    if (i == 30) { //if the object is kitten
        kitscene();
        return;
    } else {
        printf("\033[1;1H
\033[1;1H"); //clear line manually because windwos termina lis crap
        for(int j = 0; j < 128; j++) {
            printf("%c",text[(object_text[i])][j]);
        }
        return;
    }
}

void moverobot(int direction) {
    switch(direction) {
        case 72 : //up
            move_stop = 0; //for all objects including kitten
            printf("\033[%d;%dH ",(y[31] + 3),(x[31] + 1)); //if robot coords + 1
in wahatever direction equal the coords of something else then dont move, otherwise
set the coods to that position
            for(int i = 0; i < (objects + 1); i++) { //objects + 1 to account for
kitten
                if((y[31] - 1 == (-1))) { //if collided with a wall, dont do the
text/kitten check because otherwise it will return the text of object 0
                    printf("\033[%d;%dH\033[%d;1m%c\033[0m",(y[31] + 3),(x[31] +
1),object_color[31],(object_symbol[31]));
                    return;
                } else if(((x[31] == x[i]) && ((y[31] - 1) == y[i]))) { //trying to
have same coords as an object, or go out of bounds
                    objectcollidetext(i);
                    printf("\033[%d;%dH\033[%d;1m%c\033[0m",(y[31] + 3),(x[31] +
1),object_color[31],(object_symbol[31]));
                    return;
                }
            }
            y[31] = y[31] - 1;
            printf("\033[%d;%dH\033[%d;1m%c\033[0m",(y[31] + 3),(x[31] +

```

```

1),object_color[31],(object_symbol[31]));
    return;
    case 80 : //down //for all objects including kitten
        printf("\033[%d;%dH ",(y[31] + 3),(x[31] + 1)); //if robot coords + 1
in wahatever direction equal the coords of something else then dont move, otherwise
set the coods to that position
        for(int i = 0; i < (objects + 1); i++) { //objects + 1 to account for
kitten
            if((y[31] + 1 == 64)) { //if collided with a wall, dont do the
text/kitten check because otherwise it will return the text of object 0
                printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1),object_color[31],(object_symbol[31]));
                return;
            } else if((x[31] == x[i]) && ((y[31] + 1) == y[i])) { //trying to
have same coords as an object, or go out of bounds
                objectcollidetext(i);
                printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1),object_color[31],(object_symbol[31]));
                return;
            }
        }
        y[31] = y[31] + 1;
        printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1),object_color[31],(object_symbol[31]));
        return;
    case 75 : //left //for all objects including kitten
        printf("\033[%d;%dH ",(y[31] + 3),(x[31] + 1)); //if robot coords + 1
in wahatever direction equal the coords of something else then dont move, otherwise
set the coods to that position
        for(int i = 0; i < (objects + 1); i++) { //objects + 1 to account for
kitten
            if((x[31] - 1 == (-1))) { //if collided with a wall, dont do the
text/kitten check because otherwise it will return the text of object 0
                printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1),object_color[31],(object_symbol[31]));
                return;
            } else if(((x[31] - 1) == x[i]) && (y[31] == y[i])) { //trying to
have same coords as an object, or go out of bounds
                objectcollidetext(i);
                printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1),object_color[31],(object_symbol[31]));
                return;
            }
        }
        x[31] = x[31] - 1;
        printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1),object_color[31],(object_symbol[31]));
        return;
    case 77 : //right //for all objects including kitten
        printf("\033[%d;%dH ",(y[31] + 3),(x[31] + 1)); //if robot coords + 1
in wahatever direction equal the coords of something else then dont move, otherwise
set the coods to that position
        for(int i = 0; i < (objects + 1); i++) { //objects + 1 to account for
kitten
            if((x[31] + 1 == 128)) { //if collided with a wall, dont do the
text/kitten check because otherwise it will return the text of object 0
                printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1),object_color[31],(object_symbol[31]));
                return;
            }
        }

```

```

        } else if(((x[31] + 1) == x[i]) && (y[31] == y[i])) { //trying to
have same coords as an object, or go out of bounds
        objectcollidetext(i);
        printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1), object_color[31], (object_symbol[31]));
        return;
    }
}
x[31] = x[31] + 1;
printf("\033[%d;%dH\033[%d;1m%c\033[0m", (y[31] + 3), (x[31] +
1), object_color[31], (object_symbol[31]));
return;
}
}

int main() {
    initialize();
    while(1) {
        in = getch();
        if (in == 0 || in == 0xE0) in = getch(); //dang ol windows makes you call
functions and arrows keys twice to get the input
        switch(in) { //this switch statement is kinda goofy because it only really
calls moverobot() but thats got a switch statement in it right away, but now theres
other buttons too so its good now
            case 72 : //up
                moverobot(in);
                break;
            case 80 : //down
                moverobot(in);
                break;
            case 75 : //left
                moverobot(in);
                break;
            case 77 : //right
                moverobot(in);
                break;
            case 27 : //quit with prompt //esc
                printf("\033[1;1H
\033[1;1HQuit? [y/n]");
                if(getch() == 121) {
                    exit(0);
                } else {
                    printf("\033[1;1H          "); //clear the quit dialog
                    break;
                }
            case 113 : //quit with prompt //q
                printf("\033[1;1H
\033[1;1HQuit? [y/n]");
                if(getch() == 121) {
                    exit(0);
                } else {
                    printf("\033[1;1H          "); //clear the quit dialog
                    break;
                }
            default :
                printf("\a"); //bell, possibly annoying
        }
    }
}

```