Go Cart Racer

On your mark, get set, GO! Zoom around the tracks as you try to beat your best time. If you crash not only will you lose control of your car and lose valuable time, but your acceleration will drop too.

Use the following keys to control your car:

Arrow Keys..... Move
] .....Speed Up
[..... Slow Down

Maps are encoded using the same technique as found in the decripter.c program. It is therefore possible, if you are familiar with that program, to write your own tracks.

Go-Cart races is written by Joseph Larson.

/* gocart.c linsting begins: */

```
#include <curses.h>
#include <time.h>
#define XSIZE 79
#define YSIZE 24

// Start Track Data
#define NUMLEVELS 3
char* track[NUMLEVELS] = {
"3!-?!*?!*?!(1!,?!!\"~}$~r!!!/o!!!+7!\"3!`1!%~3P3!&~o/70~~0$7P~y!0O/3!-1+z!!$7"
"-7!!/s(7!!Ps(?!-3+t03l~v't03h~u%r/3!!!-3!!P!H!!/7#?!!`7&q!!l1",

"3)1+?!*?&1+?!(1@!`3!!O*~}$tlv!0O'p0x!,?)pnw&3+7*plw&3+7*plw.3-3`1,q,r.3-3`1,v"
"+7Py.7*{&7,3`1,~1/3\"7*~o/3\"7*~o/3#?)~o/3`z'~O/3hz'~O/3`y%~703!!!-3!-1!'3!.7"
"!'?!.O!(q!00",

"P~x!*7l~!1\"1v&1!!.1!@&1!*1!@!!+1*1/1+1!'~p.7(1h~!0~~~r07!l~{01!*~u!'1&{!!!4!"
",1!2!2!@*1\"1$~1(1!-~v!&1n~}!L0~~~q(1!-1!!*1!P)1!@!+1/7-1!01!0"
};
int sta[NUMLEVELS][3] = {{41, 2, 1}, {41, 2, 1}, {3, 1, 2}};
int fin[NUMLEVELS][4] = {{36,1,36,3}, {36,1,36,3}, {78, 20, 78, 23}};
// End Track Data

int car[4];
char buf[YSIZE][XSIZE];

int init() { // Setup Curses the way we want it.
  initscr ();
#ifdef PDCURSES
  PDC_set_title("Cymon's Games - Go Cart Racer by Joe Larson");
#endif
  if ((LINES < YSIZE) || (COLS < XSIZE)) {
    endwin();
    return (1);
  }
  raw (); nodelay(stdscr,1); noecho(); curs_set(0); nonl(); keypad(stdscr,1);
  start_color();
  init_pair (1, COLOR_RED, COLOR_WHITE); // Track
  init_pair (2, COLOR_BLACK, COLOR_YELLOW);// Finish
  init_pair (3, COLOR_BLUE, COLOR_WHITE); // Car
  init_pair (4, COLOR_RED, COLOR_BLACK); // Red Light
  init_pair (5, COLOR_YELLOW, COLOR_BLACK); // Yellow Light
  init_pair (6, COLOR_GREEN, COLOR_BLACK); // Green Light
  car[0] = ACS_UARROW;
  car[1] = ACS_RARROW;
  car[2] = ACS_DARROW;
  car[3] = ACS_LARROW;
  return 0;
}

void draw_track (int lvl) {
  int x, y, c, pl, cur, tr;

  for (x = 0; x <= XSIZE ; x++) mvaddch(0, x, ACS_BOARD|COLOR_PAIR(1));
  c = pl = 0; cur = track[lvl][0] - 33;
  for (y = 1; y < YSIZE; y++) {
    mvaddch (y, 0, ACS_BOARD|COLOR_PAIR(1));
    for (x = 1; x < XSIZE; x++) {
      if (pl == 6) {cur = track[lvl][++c] - 33; pl = 0;}
      if (cur < 15) {cur++; tr = 0;}
      else if (cur > 78) {cur--; tr = 1;}
      else tr = ((cur - 15) & (1 << pl++));
```

/* Listing continued on next page…*/

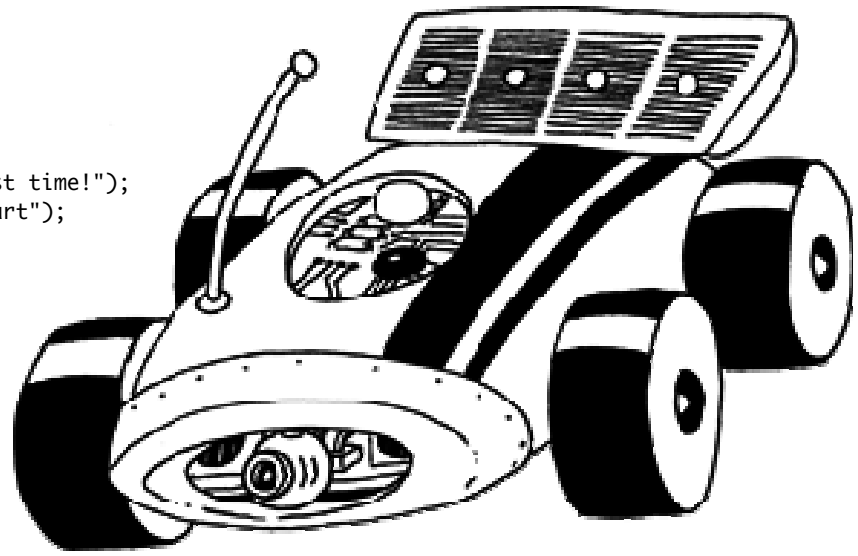
/* Listing continued from previous page */

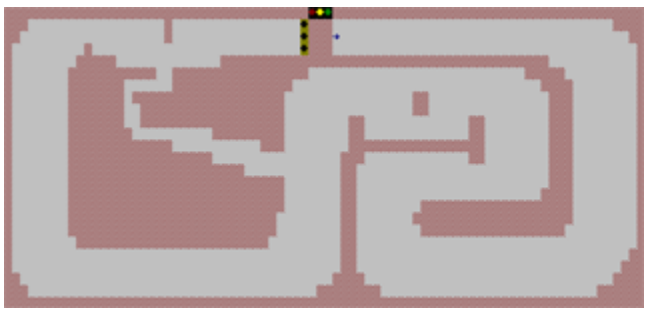
```
      buf[y][x] = !tr;
      tr = (tr) ? ACS_BOARD : ' ';
      mvaddch (y, x, tr | COLOR_PAIR (1));
      refresh();
    }
    mvaddch (y, XSIZE, ACS_BOARD|COLOR_PAIR (1));
  }
  for (x = 0; x <= XSIZE; x++) mvaddch(YSIZE, x, ACS_BOARD|COLOR_PAIR(1));
  for (y = fin[lvl][1]; y <= fin[lvl][3]; y++)
    for (x = fin[lvl][0]; x <= fin[lvl][2]; x++)
      mvaddch(y, x, ACS_DIAMOND|COLOR_PAIR(2));
}

void info () {
  mvaddstr(8, 33, "Go Cart Racer");
  mvaddstr(9, 33, "by Joe Larson");
  mvaddstr(11, 33, "Instructions:");
  mvaddstr(12, 31, "Arrow Keys - Move");
  mvaddstr(13, 33, "] - Speed Up");
        mvaddstr(14, 33, "[ - Slow Down");
        mvaddstr(15, 31, "Press 'Q' to quit");
        mvaddstr(17, 26, "Try to beat your best time!");
        mvaddstr(18, 28, "Press any key to start");
        refresh ();
        nodelay(stdscr, 0);
  getch ();
  nodelay(stdscr, 1);
}

void play () {
  int x, y, ch, dir, crash, speed, level;
  double secs, totsecs;
  time_t start, timer;

  totsecs = 0;
  for (level = 0; level < NUMLEVELS; level++) {
    draw_track (level);
    x = sta[level][0], y = sta[level][1]; dir = sta[level][2];
    mvaddch (y, x, car[dir]|COLOR_PAIR(3));
    refresh ();
    for (ch = 0; ch < 3; ch++)
      mvaddch(0, 38 + ch, ACS_DIAMOND|COLOR_PAIR(4 + ch));
    refresh();
    for (ch = 0; ch < 3; ch++) {
      napms (1000);
      mvaddch(0, 38 + ch, ACS_DIAMOND|COLOR_PAIR(4 + ch)|A_BOLD); refresh();
      beep();
      if (ch < 2) mvaddch(0, 38 + ch, ACS_DIAMOND|COLOR_PAIR(4 + ch));
    }
    time(&start); crash = 0; speed = 100;
    do {
      time(&timer);
      secs = difftime(timer, start);
      mvprintw (YSIZE, 37, " %.0f Seconds ", secs);
      mvaddch (y, x, ' '|COLOR_PAIR(1));
      if (crash) {dir++; dir %= 4; crash--; speed = 100;}
      else {
        ch = getch();
        switch (ch) {
          case KEY_UP    : dir = 0; break;
          case KEY_RIGHT : dir = 1; break;
          case KEY_DOWN  : dir = 2; break;
          case KEY_LEFT  : dir = 3; break;
```





/* Listing continued on next page…*/

/* Listing continued from previous page */

```
          case '[' : speed += 10; break;
          case ']' : speed -= 5; if (speed < 0) speed = 0; break;
          case 'q' :
          case 'Q' : return;
        }
        switch (dir) {
          case 0 : y--; if (!buf[y][x]) {crash = 4; y++;} break;
          case 1 : x++; if (!buf[y][x]) {crash = 4; x--;} break;
          case 2 : y++; if (!buf[y][x]) {crash = 4; y--;} break;
          case 3 : x--; if (!buf[y][x]) {crash = 4; x++;} break;
        }
      }
      mvaddch (y, x, car[dir]|COLOR_PAIR(3));
      refresh ();
      napms (speed);
    } while (!((x >= fin[level][0]) && (x <= fin[level][2])
           && (y >= fin[level][1]) && (y <= fin[level][3])));
    totsecs += secs;
  }
  mvaddstr (10, 35, "You WIN!");
  mvaddstr (12, 34, "Total Time:");
  mvprintw (13, 34, "%.0f Seconds", totsecs);
}

int main () {
  int ch;

  if (init()) return 0;
  draw_track(0);
  info ();
  do {
    play();
    mvaddstr(15, 31, "Press 'R' to replay");
    mvaddstr(16, 31, "Press 'Q' to quit");
    do {ch = getch();}
    while ((ch != 'q') && (ch != 'Q') && (ch != 'r') && (ch != 'R'));
  } while ((ch != 'q') && (ch != 'Q'));
  endwin();
  return 0;
}
```