**Lab Assignment #03**

**Compiler construction**



**COMSATS UNIVERSITY**

**ATTOCK CAMPUS**

**Submitted By**

SAFWANA

**Registration No**

SP21-BCS-028

**Date**

05-24-2024

**Code**

#include <iostream>

#include <vector>

#include <ncurses.h>

#include <unistd.h>

#include <cstdlib>

#include <ctime>

#include <algorithm> // For remove\_if

#define WIDTH 40

#define HEIGHT 20

#define CRIMINAL\_SPEED 1

using namespace std;

class Entity {

public:

int x, y;

char symbol;

Entity(int x, int y, char symbol) : x(x), y(y), symbol(symbol) {}

virtual void move() = 0;

};

class CopCar : public Entity {

public:

CopCar(int x, int y) : Entity(x, y, 'C') {}

void move(int dx, int dy) {

x += dx;

y += dy;

if (x < 0) x = 0;

if (x >= WIDTH) x = WIDTH - 1;

if (y < 0) y = 0;

if (y >= HEIGHT) y = HEIGHT - 1;

}

void move() override {}

};

class Bullet : public Entity {

public:

Bullet(int x, int y) : Entity(x, y, '|') {}

void move() override {

y -= 1;

}

};

class Criminal : public Entity {

public:

Criminal(int x, int y) : Entity(x, y, 'X') {}

void move() override {

y += CRIMINAL\_SPEED;

}

};

vector<Criminal> criminals;

vector<Bullet> bullets;

CopCar cop(WIDTH / 2, HEIGHT - 2);

bool game\_over = false;

void draw() {

clear();

mvaddch(cop.y, cop.x, cop.symbol);

for (auto &bullet : bullets) {

mvaddch(bullet.y, bullet.x, bullet.symbol);

}

for (auto &criminal : criminals) {

mvaddch(criminal.y, criminal.x, criminal.symbol);

}

refresh();

}

void update() {

for (auto &bullet : bullets) {

bullet.move();

}

for (auto &criminal : criminals) {

criminal.move();

if (criminal.y >= HEIGHT) {

game\_over = true;

}

}

bullets.erase(remove\_if(bullets.begin(), bullets.end(), [](Bullet &b) { return b.y < 0; }), bullets.end());

criminals.erase(remove\_if(criminals.begin(), criminals.end(), [](Criminal &c) { return c.y >= HEIGHT; }), criminals.end());

for (auto &bullet : bullets) {

for (auto &criminal : criminals) {

if (bullet.x == criminal.x && bullet.y == criminal.y) {

bullet.y = -1;

criminal.y = HEIGHT;

}

}

}

}

void spawn\_criminal() {

int x = rand() % WIDTH;

criminals.push\_back(Criminal(x, 0));

}

int main() {

srand(time(0));

initscr();

noecho();

cbreak();

timeout(100);

keypad(stdscr, TRUE);

while (!game\_over) {

int ch = getch();

switch (ch) {

case KEY\_UP:

cop.move(0, -1);

break;

case KEY\_DOWN:

cop.move(0, 1);

break;

case KEY\_LEFT:

cop.move(-1, 0);

break;

case KEY\_RIGHT:

cop.move(1, 0);

break;

case ' ':

bullets.push\_back(Bullet(cop.x, cop.y - 1));

break;

}

if (rand() % 10 == 0) {

spawn\_criminal();

}

update();

draw();

usleep(50000); // Sleep for 50ms

}

endwin();

cout << "Game Over!" << endl;

return 0;

}