ELE00107M C Programming for MSc **Coursework Assessment**

2021/22

Exam number: Y3900784

Date: 30/1/2022

# Requirements

In this experiment we need to implement a simple computer game, a javelin throwing game, using C language. The player controls the movement of the stick-person and the action of throwing the javelin with a mouse or keys. The betting speed of the javelin can be chosen, and when the javelin is thrown, an arc is drawn and the score is calculated and output to the screen when it lands. We need to add multiple players mode or computer player, and we can also add some music to the game.

The programming environment we need to use for this experiment is cold blocks, developed in a windows environment. In the implementation of the program, we use the graphics library provided by the university to implement some of the functions

# Analysis

The target audience for this program can be anyone older than 6 years old, with or without computer skills, who should be able to quickly understand the rules and operation of the game and get up and running playing.

**Aim**: This program should allow the user to control the movement of the stick person, the selection of the throwing angle and the throwing of the javelin using the buttons. When the javelin is thrown, the user can select three different initial velocities. (Divided into three small sections, when the user throws the javelin in the first section the velocity speed on the x-coordinate is V1, when in the second section the velocity speed of the javelin on the x-coordinate is V2, when in the third section the velocity speed of the javelin is V3.) When the javelin is thrown, the arc trajectory of the movement is shown on the screen. When the javelin lands the score at this point is calculated, and then the score is printed on the upper left corner of the screen.

First of all, this game has different modes, single-player mode and multi-player mode, so it is better to develop a user menu at the start of the game. The player can select which game mode they want to play, and they also can quit the game at this stage. And it would be great to have a “Game description” for the new player. Below is the algorithm of the menu.

Exit game

Key = ‘3’

Key = ‘4’ key = ‘2’

Start menu

Key = ‘1’

Insturction

Multi-player mode

Single player mode

2.1

**Calculate the angle of the javelin:**

The requirement let us need to control the angle of the javelin, so it is important to think about how to calculate the angle and the algorithm we need to use.

The lab script has given us the formula to calculate the velocity using the angle of the javelin. We can use this formula to calculate the angle.

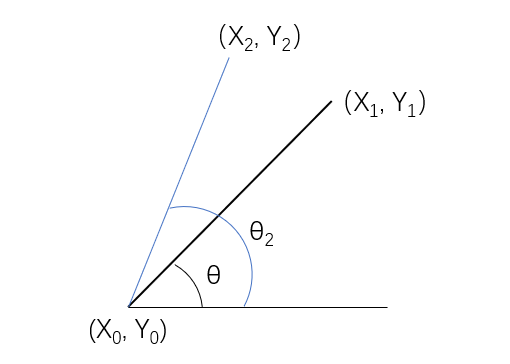
*X = L\*cos α*

*Y = L\* cos α*

*2.2*

2.2 is the transformed form of the formula on the lab script, α is the angle of the javelin, L is the length of the javelin, X and Y are the coordinates of the javelin’s tip.

The x and y positions of the javelin have been known after drawing the graphic on the screen and the initial angle of the javelin is 45 degrees. Then we can calculate the length of the javelin by using the Trigonometric functions. If the user wants to adjust the angle, the value of ***α*** will be changed, and then using the formula in 2.2 we can calculate the new coordinates (x, y) of the javelin’s tip. Then connect the new coordinate to the origin position of the javelin.



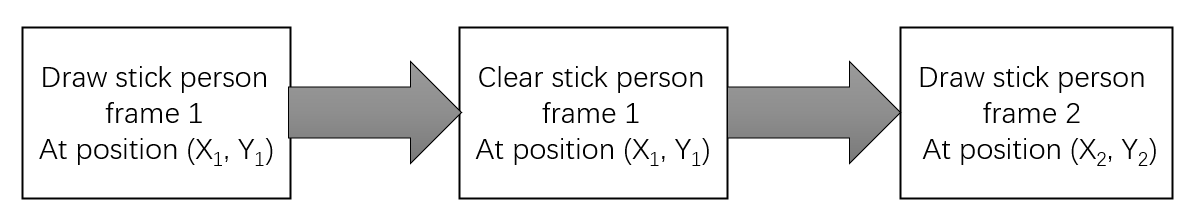
2.3

**Control the stick person:**

The player could control the stick person to move and control when to throw the javelin.

Use the character ‘D’ on the keyboard to control the stick person to move forward. The program will increase the x position value of the stick person when the ‘D’ key is pressed, then the program draws a new stick person at the new x position.

And according to the hints in the lab script, we need to clear the previous frame when we control the stick person to move forward (set the colour to black to cover the previous frame).



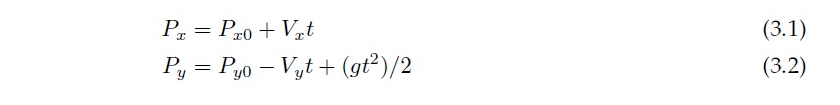
2.4

**Determine whether the foul**: The stick person running stage must be limited, when the stick person moves to the right and exceeds the red line (this means the x position of the stick person is greater than the red line), the program will determine that the player violates the rules and pop-up violation tips.

**Control the javelin:**

The up and down keys control the angle of the javelin, and the ‘W’ key throws the javelin.

The program first has to output the basic graphic elements such as stick person, javelin, ground, and score marker. The player's score is output in the upper left corner. The gravity factor needs to be considered during the javelin’s fall. We can calculate the trajectory of the javelin falling type by Newton's formula of gravity, the formula is



2.5

Which is shown in the lab script.

The three optional initial speeds of the javelin in this game are 30, 80 and 150, and the player's assist range in the assist phase is 0~100 in phase 1 (initial speeds 30), 100~200 in phase 2 (initial speeds 80), and 200~300 in phase 3 (initial speeds 150).

**Calculate the score:**

The program will calculate the position of the javelin falls when thrown the javelin by using the formula in **2.5**. There has a range for each section to obtain scores (See figure 2.6).

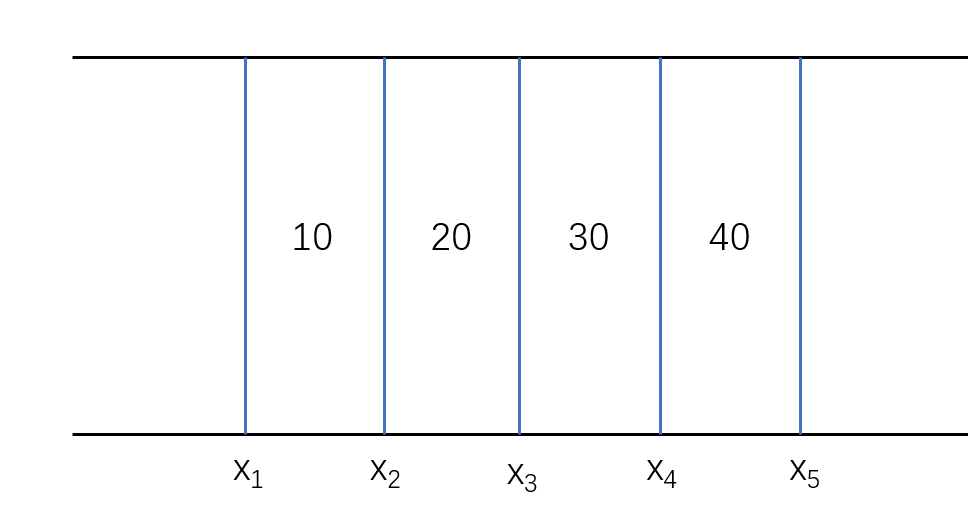


Figure 2.6

The program will compare the x position on the ground with the position of the javelin falls, to determine how much score the player got.

# Specification

What the program must do: Print a graphical interface and display the matchmaker, the javelin, the ground, the score marker, the trajectory of the javelin and the score in the interface. The player can control at least the stick person's movement and the javelin's throwing, and the angle of the javelin could be adjusted. The game should have a multi-player mode for 2 players, and they can compete with each other.

* A start menu
* An instruction for the new player about how to play this game
* Single-player mode
* Multi-player mode
* Control to move the stick person
* Adjust the velocity speed and angle of the javelin
* Calculate scores and print the best score. If the game is in multiplayer mode, the program will calculate the sum of the score for each player and determine who is the winner.

# Design

## User Interface design

At the beginning of the game, the program will start with a user menu and the player could choose which one they want to choose. The following text should be displayed.

*[1] Play game -> Single-player mode*

*[2] Play game -> multi-player mode*

*[3] Instructions for new players*

*[4] Exit*

The program will wait for the key press. When the user press ‘1’, it will enter the single-player mode, one stick person and ground and target will appear on the screen. When the user press ‘2’, it will enter the multi-player mode, there are 2 stick persons that will print on the screen with targets and ground. When the user press ‘3’, the program will display the instructions on a new screen, and press ‘q’ to return to the menu. When the user press ‘4’, quit the game and close the window.

The player is required to provide control actions (movement of the stick person, javelin throwing action, etc.). In this programming, draw a circle on the screen with a radius of 10 and a fill colour of yellow, and then its body. Its body is made of straight lines with a width of 2.5 mm.

The second step starts drawing the javelin, which is a straight line diagonally across the pixel range (140, 300) to (60, 380).

The third step starts with drawing the ground, which is a straight yellow line parallel to the x-axis. The pixel range is (80, 520) to (1000, 450), (200, 200) to (1120, 200), (80, 840) to (1000, 840). The red line (500, 200) to (500, 840) is the boundary the stick person can move

|  |  |  |
| --- | --- | --- |
| **symbol** | **Value** | **description** |
| X\_position | 100 | The initial x position of the stick person. |
| Y\_position | 340 | The initial y position of the stick person. |
| x\_position\_2 | 100 | The initial x position of the second stick person.  (In the multi-player mode.) |
| y\_position\_2 | 740 | The initial y position of the second stick person.  (In the multi-player mode.) |
| pos\_x | 140 | Initial x-coordinate of the javelin’s tip |
| pos\_y | 380 | Initial y-coordinate of the javelin’s tip |
| pos\_x\_2 | 140 | Initial x-coordinate of the javelin’s tip  (multi-player mode) |
| pos\_y\_2 | 700 | Initial y-coordinate of the javelin’s tip  (multi-player mode) |

|  |  |  |
| --- | --- | --- |
| javelin\_long | 84.85 | The length of the javelin. |
| radians | 45 in degrees | The initial degree of the javelin in radians |
| gravity | 9.81 | Earth acceleration |
|  |  |  |

Global variable



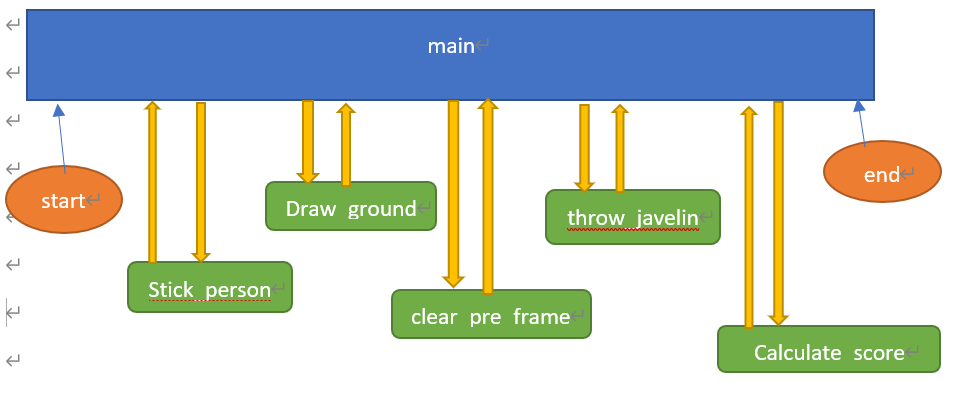
Text display: The text will be displayed by the function outtextxy(x, y, “Text”).

|  |  |  |
| --- | --- | --- |
| **Text** | **Position** |  |
| [1]Play game --> single-player mode" | (200, 150) | On the start menu |
| [2] Play game --> multi-player mode | (200, 250) | On the start menu |
| [3] Instructions for new players | (200, 350) | On the start menu |
| [4] Exit game | (200, 450) | On the start menu |
| Best Score | (200, 100) | After the player throws the javelin |
| Current Score | (200, 130) | After the player throws the javelin |
| player 1 scores | (200, 100) | After the player throws the javelin |
| player 2 scores | (200, 130) | After the player throws the javelin |
| game over | (600, 300) | illegal |
| You have reached the maximum number of attempts! | (460, 320) | After 6 times throwing |
| Your best score is: | (460, 340) | After 6 times throwing and calculating the best score |
| Now is the player 2 turn! | (860, 140) | After player 1 throws 6 times |
| Player 1 is the winner! | (460, 140) | After player 2 throws 6 times |
| Player 2 is the winner! | (460, 140) | After player 2 throws 6 times |
| Press 'q' to go back to the menu | (460, 160) | After player 2 throws 6 times |
| Same score! | (460, 140) | After player 2 throws 6 times |

**Acceptable user input:**

User\_control: ‘w’ ‘a’ ‘d’ ‘j’ ‘l’ ‘I’ ‘↑’ ‘↓’ ‘1’ ‘2’ ‘3’ ‘4’

## Structure of program in terms of functions



|  |  |  |
| --- | --- | --- |
| **Function** | **Returns** | **Description** |
| main | Default value (integer) | Driving function, calls all others, always returns 0 |
| User\_menu | None | Display the start menu |
| Instruction | None | Show the instruction of the game |
| play\_SinglePlayer\_mode/  play\_MultiPlayer\_mode | none | Single-player mode/Multi-player mode |
| Stick\_person | None | Draw the stick person |
| stick\_person\_2 | none | Draw the stick person 2 |
| control\_stick\_person | none | Control the movement of the stick person |
| Draw\_ground | None | Draw a ground |
| clear\_pre\_frame | None | Clear the path the stick person previously passed |
| set\_angle | None | Adjust the angle of the javelin |
| javelin | None | Draw the javelin on the current position |
| throw\_javelin | None | Action of throw |
| Calculate\_score | None | Calculate the score |

## Pseudo Code Description

**Function main**

Call user\_menu function

Begin loop

call ‘getch’ to get a key from the user, and put the result in key

if key is ‘1’

close the menu and start the single-player mode

End if

If key is ‘2’

close the menu and start the multiplayer mode

End if

If key is ‘3’

Show the instruction of the game

End if

If key is ‘4’

Quite game and close the screen

End if

End loop

End function main

**Single-player mode**

Call stick\_person function

Call draw\_ground function

Call target function

Begin while loop

call ‘getch’ to get a key from the user, and put the result in key

call control\_stick\_person function

call set\_angle function

call javelin() function

call throw\_javelin function

call game\_over function

end loop

Multi-player mode

Call stick\_person function

Call stick\_person\_2 function

Call draw\_ground\_multiPlayer\_mode function

Call draw\_target\_multiPlayer\_mode function

Begin while loop

call ‘getch’ to get a key from the user, and put the result in key

call control\_stick\_person function

call set\_angle function

call javelin() function

call throw\_javelin\_person\_1 function

call game\_over\_player\_1 function

if player\_2\_start

call control\_stick\_person\_2 function

call set\_angle function

call javelin\_2() function

call throw\_javelin\_person\_2 function

call game\_over\_player\_2 function

end if

# Implementation:

#include <stdio.h>

#include <math.h>

#include <graphics\_lib.h>

/\* Declare two variables for the x and y positions \*/

int x\_position**,** y\_position**,** x\_position\_2**,** y\_position\_2**;**

int initial\_pos\_x**,** initial\_pos\_y**,** initial\_pos\_x\_2**,** initial\_pos\_y\_2**;**//coordinate of the javelin tip when it was thrown

int o\_pos\_x\_2**,** o\_pos\_y\_2**,** pos\_x\_2**,** pos\_y\_2**;**

int o\_pos\_x**,** o\_pos\_y**,** pos\_x**,** pos\_y**,** Amount\_of\_change\_y**,** Amount\_of\_change\_x**;**//Initial coordinate of the javelin tip

int javelin\_long **=** 84.85**;**// the length of the javelin

double radians **=** **(**45**\***M\_PI**)/**180**;**

double t**,** vel\_x**,** vel\_y**,** gravity **=** 9.81**;**//javelin

char user\_control**,** user\_2\_control**;**//user press keyboard value

int score**,** best\_score **=** 0**;;**

int sum\_of\_score\_player\_1 **=** 0**,** sum\_of\_score\_player\_2 **=** 0**;**

int player\_2\_start**;**

int scores**[**10**]** **=** **{**10**,** 20**,** 30**,** 40**,** 50**,** 40**,** 30 **,**20**,** 10**,** 40**};**

char scores\_label**[**3**],** player\_1\_scores\_label**[**3**],** player\_2\_scores\_label**[**3**],** player\_1\_sum\_of\_scoures\_label**[**3**],** player\_2\_sum\_of\_scoures\_label**[**3**];**

char player\_scores\_label**[**3**],** player\_BestScores\_label**[**3**];**

//The main program

int main**(**void**)**

**{**

/\*Open the start menu \*/

user\_menu**();**

**while(**1**)**

**{**

user\_control **=** getch**();**

**switch(**user\_control**)**

**{**

**case** '1'**:**

closegraph**();**

/\*start single-player mode\*/

play\_SinglePlayer\_mode**();**

**break;**

**case** '2'**:**

closegraph**();**

/\*start multi-player mode\*/

play\_MultiPlayer\_mode**();**

**break;**

**case** '3'**:**

closegraph**();**

/\*open the instruction\*/

Instruction**();**

**break;**

**case** 'q'**:**

initwindow**(**640**,** 480**);**

/\*back to the start menu\*/

user\_menu**();**

**}**

**}**

**return** 0**;**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Below are the functions declaration\*/

/\* the game start menu\*/

void user\_menu**(**void**)**

**{**

initwindow**(**640**,** 580**);**

initfont**();**

setcolor**(**GREEN**);**

outtextxy**(**200**,** 150**,** "[1] Play game --> single-player mode"**);**

outtextxy**(**200**,** 250**,** "[2] Play game --> multi-player mode"**);**

outtextxy**(**200**,** 350**,** "[3] Instructions for new players"**);**

outtextxy**(**200**,** 450**,** "[4] Exit game"**);**

update\_display**();**

**}**

/\*display the instruction of the game \*/

void Instruction**(**void**)**

**{**

initwindow**(**840**,** 580**);**

initfont**();**

setcolor**(**GREEN**);**

outtextxy**(**200**,** 150**,** "single-player mode: Press 'd' to move the stick person"**);**

outtextxy**(**200**,**170**,** "Press 'w' to throw the javelin"**);**

outtextxy**(**200**,**190**,** "Press up and down keys to adjust the angle of javelin"**);**

outtextxy**(**200**,**210**,** "After throwing the javelin you will get a score, you have 6 times attempt"**);**

outtextxy**(**200**,** 250**,** "multi-player mode: press 'd' to move the stick person 1"**);**

outtextxy**(**200**,** 270**,** "press 'w' to throw the javelin 1"**);**

outtextxy**(**200**,** 290**,** "press 'l' to move the stick person 2"**);**

outtextxy**(**200**,** 310**,** "press 'i' to throw the javelin 2"**);**

outtextxy**(**200**,** 330**,** "Press up and down keys to adjust the angle of javelin 1&2"**);**

outtextxy**(**200**,** 350**,** "The program will calculate the sum of the score of your all attempt"**);**

outtextxy**(**200**,** 370**,** "And determine who is the winner"**);**

outtextxy**(**200**,** 400**,** "Press 'q' to return the menu"**);**

**if** **(**user\_control **==** 'q'**)**

**{**

closegraph**();**

**}**

update\_display**();**

**}**

/\* Play the single-player mode\*/

void play\_SinglePlayer\_mode**(**void**)**

**{**

initwindow**(**1280**,** 960**);**

/\* Set up some coordinates \*/

x\_position **=** 100**;**

y\_position **=** 340**;**

/\* draw the person\*/

stick\_person**();**

/\* draw the ground\*/

draw\_ground**();**

draw\_target**();**//draw and display the target

update\_display**();**

user\_control **=** ' '**;**

user\_2\_control **=** ' '**;**

**while(**1**)**

**{**

user\_control **=** getch**();**

/\*control to move the stick person\*/

control\_stick\_person**(**user\_control**);**

/\* set the angle of the javelin\*/

set\_angle**(**user\_control**);**

/\* drawing the javelin\*/

javelin**();**

/\* control to throw the javelin\*/

throw\_javelin**(**user\_control**);**

game\_over**();**

**}**

**}**

void play\_MultiPlayer\_mode**(**void**)**

**{**

initwindow**(**1280**,** 960**);**

/\* Set up some coordinates \*/

/\* the first player\*/

x\_position **=** 100**;**

y\_position **=** 340**;**

/\* the second player\*/

x\_position\_2 **=** 100**;**

y\_position\_2 **=** 740**;**

stick\_person**();**//draw the person

stick\_person\_2**();**

draw\_ground\_multiPlayer\_mode**();**

draw\_target\_multiPlayer\_mode**();**//draw and display the target

update\_display**();**

user\_control **=** ' '**;**

user\_2\_control **=** ' '**;**

**while(**1**)**

**{**

user\_control **=** getch**();**

/\*control to move the stick person\*/

control\_stick\_person**(**user\_control**);**

/\* set the angle of the javelin\*/

set\_angle**(**user\_control**);**

/\* drawing the javelin\*/

javelin**();**

/\* player 1 to control to throw the javelin\*/

throw\_javelin\_person\_1**(**user\_control**);**

game\_over\_Player\_1**();**

/\* Determine if the player 2 start the game\*/

**if(**player\_2\_start **==** 1**)**

**{**

user\_2\_control **=** getch**();**

/\*control to move the stick person\*/

control\_stick\_person\_2**(**user\_2\_control**);**

/\* set the angle of the javelin\*/

set\_angle**(**user\_2\_control**);**

/\* drawing the javelin\*/

javelin\_2**();**

/\* player 2 to control to throw the javelin\*/

throw\_javelin\_person\_2**(**user\_2\_control**);**

game\_over\_Player\_2**();**

**}**

**else**

**{**

user\_2\_control **=** ' '**;**

**}**

/\* exit th game\*/

**if(**user\_2\_control **==** 'q' **||** user\_control **==** 'q'**)**

**{**

closegraph**();**

**}**

**}**

**}**

//draw and display the target

void draw\_target**(**void**)**

**{**

int i**;**

int target\_position\_x **=** 600**;**

int target\_position\_y **=** 200**;**

initfont**();**

/\* display the text\*/

outtextxy**(**200**,** 100**,** "Best Score: " **);**

outtextxy**(**200**,** 130**,** "Current Score: " **);**

**for(**i **=** 0**;** i**<** 10**;** i**++)**

**{**

setcolor**(**BLUE**);**

/\* Creating scoring areas\*/

line**(**target\_position\_x**,** target\_position\_y**,** target\_position\_x**,** target\_position\_y **+** 320**,** 2**);**

/\* print the score\*/

sprintf**(**scores\_label**,** "%d"**,** scores**[**i**]);**

setcolor**(**RED**);**

outtextxy**(**target\_position\_x**,** target\_position\_y**,** scores\_label**);**

/\* the red line\*/

target\_position\_x **=** target\_position\_x **+** 50**;**

**}**

**}**

void draw\_target\_multiPlayer\_mode**(**void**)**

**{**

int i**;**

int target\_position\_x **=** 600**;**

int target\_position\_y **=** 200**;**

initfont**();**

outtextxy**(**200**,** 100**,** "player 1 scores: " **);**

outtextxy**(**200**,** 130**,** "player 2 scores: " **);**

**for(**i **=** 0**;** i**<** 10**;** i**++)**

**{**

setcolor**(**BLUE**);**

/\* Creating scoring areas\*/

line**(**target\_position\_x**,** target\_position\_y**,** target\_position\_x**,** target\_position\_y **+** 640**,** 2**);**

/\* print the score\*/

sprintf**(**scores\_label**,** "%d"**,** scores**[**i**]);**

setcolor**(**RED**);**

outtextxy**(**target\_position\_x**,** target\_position\_y**,** scores\_label**);**

/\* the red line\*/

target\_position\_x **=** target\_position\_x **+** 50**;**

**}**

**}**

//Draw the ground

void draw\_ground**(**void**)**

**{**

setcolor**(**YELLOW**);**

line**(**80**,** 520**,** 1000**,** 520**,** 2**);**//draw the bottom ground

line**(**200**,** 200**,** 1120**,** 200**,** 2**);**//draw the top ground

setcolor**(**RED**);**

line**(**500**,** 200**,** 500**,** 520**,** 5**);**//draw a boundary

**}**

/\* Drawing ground, multi-player mode\*/

void draw\_ground\_multiPlayer\_mode**(**void**)**

**{**

setcolor**(**YELLOW**);**

line**(**80**,** 520**,** 1000**,** 520**,** 2**);**//draw the bottom ground

line**(**200**,** 200**,** 1120**,** 200**,** 2**);**//draw the top ground

line**(**80**,** 840**,** 1000**,** 840**,** 2**);**

setcolor**(**RED**);**

line**(**500**,** 200**,** 500**,** 840**,** 5**);**//draw a boundary

**}**

/\*draw and drive the stick person\*/

void stick\_person**(**void**)**

**{**

/\* draw a circle on the screen buffer

at x\_position, y\_position

with radius 10 and line thickness 2 \*/

setcolor**(**YELLOW**);**

filled\_circle**(**x\_position**,** y\_position**,** 10**,** YELLOW**);**//stick person's head

//body of the stick person

line**(**x\_position**,** y\_position **+** 10**,** x\_position**,** y\_position **+** 60**,** 2**);**

line**(**x\_position **-** 20**,** y\_position **+** 20**,** x\_position **+** 20**,** y\_position **+** 20**,** 2**);**

line**(**x\_position**,** y\_position **+** 60**,** x\_position **-**20**,** y\_position **+** 80**,** 2**);**

line**(**x\_position**,** y\_position **+** 60**,** x\_position **+** 20**,** y\_position **+** 80**,** 2**);**

javelin**();**

update\_display**();**

**}**

/\*draw and drive the stick person, player 2\*/

void stick\_person\_2**(**void**)**

**{**

setcolor**(**RED**);**

filled\_circle**(**x\_position\_2**,** y\_position\_2**,** 10**,** YELLOW**);**//stick person's head

//body of the stick person 2

line**(**x\_position\_2**,** y\_position\_2 **+** 10**,** x\_position\_2**,** y\_position\_2 **+** 60**,** 2**);**

line**(**x\_position\_2 **-** 20**,** y\_position\_2 **+** 20**,** x\_position\_2 **+** 20**,** y\_position\_2 **+** 20**,** 2**);**

line**(**x\_position\_2**,** y\_position\_2 **+** 60**,** x\_position\_2 **-**20**,** y\_position\_2 **+** 80**,** 2**);**

line**(**x\_position\_2**,** y\_position\_2 **+** 60**,** x\_position\_2 **+** 20**,** y\_position\_2 **+** 80**,** 2**);**

javelin\_2**();**

update\_display**();**

**}**

/\* control to move the stick person\*/

void control\_stick\_person**(**char user\_control**)**

**{**

/\* clear the previous frame \*/

clear\_person\_1\_pre\_frame**();**

/\* player 1 move forward\*/

**if(**user\_control **==** 'd'**){**

x\_position **=** x\_position **+** 3**;**

/\* Increase the speed of the stick person \*/

**if(**x\_position **>**200 **&&** x\_position **<** 300**)**

**{**

x\_position **=** x\_position **+** 5**;**

**}**

/\* Increase the speed of the stick person \*/

**if(**x\_position **>**300**)**

**{**

x\_position **=** x\_position **+** 8**;**

**}**

**}**

**else**

**if(**user\_control **==** 'a'**){**

x\_position **=** x\_position **-** 3**;**

**}**

/\* draw a circle on the screen buffer

at x\_position, y\_position

with radius 10 and line thickness 2 \*/

setcolor**(**YELLOW**);**

filled\_circle**(**x\_position**,** y\_position**,** 10**,** YELLOW**);**//stick person's head

//body of the stick person

line**(**x\_position**,** y\_position **+** 10**,** x\_position**,** y\_position **+** 60**,** 2**);**

line**(**x\_position **-** 20**,** y\_position **+** 20**,** x\_position **+** 20**,** y\_position **+** 20**,** 2**);**

line**(**x\_position**,** y\_position **+** 60**,** x\_position **-**20**,** y\_position **+** 80**,** 2**);**

line**(**x\_position**,** y\_position **+** 60**,** x\_position **+** 20**,** y\_position **+** 80**,** 2**);**

//javelin();

update\_display**();**

**}**

/\* control to move the stick person, player 2\*/

void control\_stick\_person\_2**(**char user\_2\_control**)**

**{**

/\* clear the previous frame \*/

clear\_person\_2\_pre\_frame**();**

/\* player 2 move forward\*/

**if(**user\_2\_control **==** 'l'**){**

x\_position\_2 **=** x\_position\_2 **+** 3**;**

/\* Increase the speed of the stick person \*/

**if(**x\_position\_2 **>**200 **&&** x\_position\_2 **<** 300**)**

**{**

x\_position\_2 **=** x\_position\_2 **+** 5**;**

**}**

/\* Increase the speed of the stick person \*/

**if(**x\_position\_2 **>**300**)**

**{**

x\_position\_2 **=** x\_position\_2 **+** 8**;**

**}**

**}**

**else**

**if(**user\_2\_control **==** 'j'**){**

x\_position\_2 **=** x\_position\_2 **-** 3**;**

**}**

/\* draw a circle on the screen buffer

at x\_position\_2, y\_position\_2

with radius 10 and line thickness 2 \*/

setcolor**(**RED**);**

filled\_circle**(**x\_position\_2**,** y\_position\_2**,** 10**,** RED**);**//stick person's head

//body of the stick person

line**(**x\_position\_2**,** y\_position\_2 **+** 10**,** x\_position\_2**,** y\_position\_2 **+** 60**,** 2**);**

line**(**x\_position\_2 **-** 20**,** y\_position\_2 **+** 20**,** x\_position\_2 **+** 20**,** y\_position\_2 **+** 20**,** 2**);**

line**(**x\_position\_2**,** y\_position\_2 **+** 60**,** x\_position\_2 **-**20**,** y\_position\_2 **+** 80**,** 2**);**

line**(**x\_position\_2**,** y\_position\_2 **+** 60**,** x\_position\_2 **+** 20**,** y\_position\_2 **+** 80**,** 2**);**

//javelin();

update\_display**();**

**}**

/\* clear the stick person 1 previous frame\*/

void clear\_person\_1\_pre\_frame**(**void**)**

**{**

setcolor**(**BLACK**);**

filled\_circle**(**x\_position**,** y\_position**,** 10**,** BLACK**);**

line**(**x\_position**,**y\_position **+** 10**,** x\_position**,** y\_position **+** 60**,** 2**);**

line**(**x\_position **-** 20**,** y\_position **+** 20**,** x\_position **+** 20**,** y\_position **+** 20**,** 2**);**

line**(**x\_position**,** y\_position **+** 60**,** x\_position **-** 20**,** y\_position **+** 80**,** 2**);**

line**(**x\_position**,** y\_position **+** 60**,** x\_position **+** 20**,** y\_position **+** 80**,** 2**);**

javelin\_clear**();**

update\_display**();**

**}**

/\* clear the stick person 2 previous frame\*/

void clear\_person\_2\_pre\_frame**(**void**)**

**{**

setcolor**(**BLACK**);**

filled\_circle**(**x\_position\_2**,** y\_position\_2**,** 10**,** BLACK**);**

line**(**x\_position\_2**,** y\_position\_2 **+** 10**,** x\_position\_2**,** y\_position\_2 **+** 60**,** 2**);**

line**(**x\_position\_2 **-** 20**,** y\_position\_2 **+** 20**,** x\_position\_2 **+** 20**,** y\_position\_2 **+** 20**,** 2**);**

line**(**x\_position\_2**,** y\_position\_2 **+** 60**,** x\_position\_2 **-** 20**,** y\_position\_2 **+** 80**,** 2**);**

line**(**x\_position\_2**,** y\_position\_2 **+** 60**,** x\_position\_2 **+** 20**,** y\_position\_2 **+** 80**,** 2**);**

javelin\_clear\_2**();**

update\_display**();**

**}**

int long\_x**,** long\_y**;**//The base and height of a triangle

//draw the javelin

//long of the javelin = 84.85

void javelin**(**void**)**

**{**

o\_pos\_x **=** x\_position **+** 40 **;**

o\_pos\_y **=** y\_position **-** 40 **;**

pos\_x **=** o\_pos\_x **+** Amount\_of\_change\_x**;** //initial x = 140

pos\_y **=** o\_pos\_y **+** Amount\_of\_change\_y**;** //initial y = 380

setcolor**(**GREEN**);**

line**(**pos\_x**,** pos\_y**,** o\_pos\_x **-** 60**,** o\_pos\_y **+** 60**,** 2**);**

update\_display**();**

**}**

void javelin\_2**(**void**)**

**{**

o\_pos\_x\_2 **=** x\_position\_2 **+** 40 **;**

o\_pos\_y\_2 **=** y\_position\_2 **-** 40 **;**

pos\_x\_2 **=** x\_position\_2 **+** 40 **;** //initial x = 140

pos\_y\_2 **=** y\_position\_2 **-** 40 **;** //initial y = 380

setcolor**(**GREEN**);**

line**(**pos\_x\_2**,** pos\_y\_2**,** o\_pos\_x\_2 **-** 60**,** o\_pos\_y\_2 **+** 60**,** 2**);**

update\_display**();**

**}**

/\*clear the previous frame of the javelin 1\*/

void javelin\_clear**(**void**)**

**{**

o\_pos\_x **=** x\_position **+** 40 **;**

o\_pos\_y **=** y\_position **-** 40 **;**

pos\_x **=** x\_position **+** 40 **+** Amount\_of\_change\_x**;**

pos\_y **=** y\_position **-** 40 **+** Amount\_of\_change\_y**;**

setcolor**(**BLACK**);**

line**(**pos\_x**,** pos\_y**,** o\_pos\_x **-** 60**,** o\_pos\_y **+** 60**,** 2**);**

update\_display**();**

**}**

/\*clear the previous frame of the javelin 2\*/

void javelin\_clear\_2**(**void**)**

**{**

o\_pos\_x\_2 **=** x\_position\_2 **+** 40 **;**

o\_pos\_y\_2 **=** y\_position\_2 **-** 40 **;**

pos\_x\_2 **=** x\_position\_2 **+** 40 **;**

pos\_y\_2 **=** y\_position\_2 **-** 40 **;**

setcolor**(**BLACK**);**

line**(**pos\_x\_2**,** pos\_y\_2**,** o\_pos\_x\_2 **-** 60**,** o\_pos\_y\_2 **+** 60**,** 2**);**

update\_display**();**

**}**

//user control the angle

//angle degree is in radians

void set\_angle**(**char user\_control**)**

**{**

**if(**user\_control **==** 72**){**

javelin\_clear**();**

radians **=** radians **+** M\_PI**/**180**;**

long\_x **=** javelin\_long **\*** cos**(**radians**);**

long\_y **=** javelin\_long **\*** sin**(**radians**);**

**if(**radians **<** 1.57 **&&** radians **>** 0**)**

**{**

Amount\_of\_change\_y **=** **-** long\_y **;**

Amount\_of\_change\_x **=** **-** long\_x**;**

**}**

**else**

**{**

Amount\_of\_change\_y **=** 0 **;**

Amount\_of\_change\_x **=** 0**;**

**}**

**}**

**else**

**if(**user\_control **==** 80**){**

javelin\_clear**();**

radians **=** radians **-** M\_PI**/**180**;**

long\_x **=** javelin\_long **\*** cos**(**radians**);**

long\_y **=** javelin\_long **\*** sin**(**radians**);**

**if(**radians **<** 1.57 **&&** radians **>** 0**)**

**{**

Amount\_of\_change\_y **=** long\_y**;**

Amount\_of\_change\_x **=** long\_x**;**

**}**

**else**

**{**

Amount\_of\_change\_y **=** 0 **;**

Amount\_of\_change\_x **=** 0**;**

**}**

**}**

**}**

void set\_angle\_2**(**char user\_2\_control**)**

**{**

**if(**user\_2\_control **==** 72**){**

javelin\_clear\_2**();**

radians **=** radians **+** M\_PI**/**180**;**

long\_x **=** javelin\_long **\*** cos**(**radians**);**

long\_y **=** javelin\_long **\*** sin**(**radians**);**

**if(**radians **<** 1.57 **&&** radians **>** 0**)**

**{**

Amount\_of\_change\_y **=** **-** long\_y **;**

Amount\_of\_change\_x **=** **-** long\_x**;**

**}**

**else**

**{**

Amount\_of\_change\_y **=** 0 **;**

Amount\_of\_change\_x **=** 0**;**

**}**

**}**

**else**

**if(**user\_2\_control **==** 80**){**

javelin\_clear\_2**();**

radians **=** radians **-** M\_PI**/**180**;**

long\_x **=** javelin\_long **\*** cos**(**radians**);**

long\_y **=** javelin\_long **\*** sin**(**radians**);**

**if(**radians **<** 1.57 **&&** radians **>** 0**)**

**{**

Amount\_of\_change\_y **=** long\_y**;**

Amount\_of\_change\_x **=** long\_x**;**

**}**

**else**

**{**

Amount\_of\_change\_y **=** 0 **;**

Amount\_of\_change\_x **=** 0**;**

**}**

**}**

**}**

int i **=** 0**;**

/\* calculate the score\*/

void calculate\_score**(**void**)**

**{**

int gap **=** 30**;**

**if(**pos\_x **<** 600**)**

**{**

score **=** 0**;**

printf**(**" Your score is: %d\n" **,**score**);**

**}**

**if(**pos\_x **>=** 600 **&&** pos\_x **<** 650**)**

**{**

score **=** 10**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=**650 **&&** pos\_x **<** 700**)**

**{**

score **=** 20**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=** 700 **&&** pos\_x **<** 750**)**

**{**

score **=** 30**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=** 750 **&&** pos\_x **<** 800**)**

**{**

score **=** 40**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=** 800 **&&** pos\_x **<** 850**)**

**{**

score **=** 50**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=** 850 **&&** pos\_x **<** 900**)**

**{**

score **=** 40**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=** 900 **&&** pos\_x **<** 950**)**

**{**

score **=** 30**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=** 950 **&&** pos\_x **<** 1000**)**

**{**

score **=** 20**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=**1000 **&&** pos\_x **<** 1500**)**

**{**

score **=** 10**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

**else**

**if(**pos\_x **>=**1500 **&&** pos\_x **<** 2000**)**

**{**

score **=** 40**;**

printf**(**" Your score is: %d\n" **,**score**);**

**if(**best\_score **<** score**)**

**{**

best\_score **=** score**;**

**}**

**}**

/\*print the score on the graphic window \*/

/\*If the player reach the maximum attempts of the game, then game over\*/

i **=** i **+** 1**;**

**if(**i **==** 6**){**

cleardevice**();**

outtextxy**(**600**,** 300**,** "game over!"**);**

outtextxy**(**460**,** 320**,** "You have reached the maximum number of attempts!"**);**

outtextxy**(**460**,** 340**,** "Your best score is: "**);**

outtextxy**(**640**,** 340**,** player\_BestScores\_label**);**

**}**

**else{**

sprintf**(**player\_scores\_label**,** "%d"**,** score**);**

sprintf**(**player\_BestScores\_label**,** "%d"**,** best\_score**);**

outtextxy**(**300 **+** i**\***gap**,** 130**,** player\_scores\_label**);**

outtextxy**(**300 **+** i**\***gap**,** 100**,** player\_BestScores\_label**);**

update\_display**();**

**}**

**}**

void calculate\_score\_Player\_1**(**void**)**

**{**

**if(**pos\_x **<** 600**)**

**{**

score **=** 0**;**

**}**

**if(**pos\_x **>=** 600 **&&** pos\_x **<** 650**)**

**{**

score **=** 10**;**

**}**

**else**

**if(**pos\_x **>=**650 **&&** pos\_x **<** 700**)**

**{**

score **=** 20**;**

**}**

**else**

**if(**pos\_x **>=** 700 **&&** pos\_x **<** 750**)**

**{**

score **=** 30**;**

**}**

**else**

**if(**pos\_x **>=** 750 **&&** pos\_x **<** 800**)**

**{**

score **=** 40**;**

**}**

**else**

**if(**pos\_x **>=** 800 **&&** pos\_x **<** 850**)**

**{**

score **=** 50**;**

**}**

**else**

**if(**pos\_x **>=** 850 **&&** pos\_x **<** 900**)**

**{**

score **=** 40**;**

**}**

**else**

**if(**pos\_x **>=** 900 **&&** pos\_x **<** 950**)**

**{**

score **=** 30**;**

**}**

**else**

**if(**pos\_x **>=** 950 **&&** pos\_x **<** 1000**)**

**{**

score **=** 20**;**

**}**

**else**

**if(**pos\_x **>=**1000 **&&** pos\_x **<** 1500**)**

**{**

score **=** 10**;**

**}**

**else**

**if(**pos\_x **>=**1500 **&&** pos\_x **<** 2000**)**

**{**

score **=** 40**;**

**}**

sum\_of\_score\_player\_1 **=** sum\_of\_score\_player\_1 **+** score**;**

**}**

void calculate\_score\_Player\_2**(**void**)**

**{**

**if(**pos\_x\_2 **<** 600**)**

**{**

score **=** 0**;**

**}**

**if(**pos\_x\_2 **>=** 600 **&&** pos\_x\_2 **<** 650**)**

**{**

score **=** 10**;**

**}**

**else**

**if(**pos\_x\_2 **>=**650 **&&** pos\_x\_2 **<** 700**)**

**{**

score **=** 20**;**

**}**

**else**

**if(**pos\_x\_2 **>=** 700 **&&** pos\_x\_2 **<** 750**)**

**{**

score **=** 30**;**

**}**

**else**

**if(**pos\_x\_2 **>=** 750 **&&** pos\_x\_2 **<** 800**)**

**{**

score **=** 40**;**

**}**

**else**

**if(**pos\_x\_2 **>=** 800 **&&** pos\_x\_2 **<** 850**)**

**{**

score **=** 50**;**

**}**

**else**

**if(**pos\_x\_2 **>=** 850 **&&** pos\_x\_2 **<** 900**)**

**{**

score **=** 40**;**

**}**

**else**

**if(**pos\_x\_2 **>=** 900 **&&** pos\_x\_2 **<** 950**)**

**{**

score **=** 30**;**

**}**

**else**

**if(**pos\_x\_2 **>=** 950 **&&** pos\_x\_2 **<** 1000**)**

**{**

score **=** 20**;**

**}**

**else**

**if(**pos\_x\_2 **>=**1000 **&&** pos\_x\_2 **<** 1500**)**

**{**

score **=** 10**;**

**}**

**else**

**if(**pos\_x\_2 **>=**1500 **&&** pos\_x\_2 **<** 2000**)**

**{**

score **=** 40**;**

**}**

sum\_of\_score\_player\_2 **=** sum\_of\_score\_player\_2 **+** score**;**

**}**

/\*print the score on the graphic window \*/

/\*If the player reach the maximum attempts of the game, then game over\*/

void print\_player1\_score**(**void**)**

**{**

int gap\_1 **=** 30**;**

i **=** i **+** 1**;**

**if(**i **==** 6**)**

**{**

player\_2\_start **=** 1**;**

outtextxy**(**860**,** 100**,** "You have reached the maximum number of attempts!"**);**

outtextxy**(**860**,** 120**,** "Your best score is: "**);**

outtextxy**(**1040**,** 100**,** player\_1\_sum\_of\_scoures\_label**);**

outtextxy**(**860**,** 140**,** "Now is the player 2 turn!"**);**

**}**

**else**

**{**

sprintf**(**player\_1\_scores\_label**,** "%d"**,** score**);**

sprintf**(**player\_1\_sum\_of\_scoures\_label**,** "%d"**,** sum\_of\_score\_player\_1**);**

outtextxy**(**300 **+** i**\***gap\_1**,** 100**,** player\_1\_sum\_of\_scoures\_label**);**

update\_display**();**

**}**

**}**

int j **=** 0**;**

void print\_player2\_score**(**void**)**

**{**

int gap\_2 **=** 30**;**

j **=** j **+** 1**;**

**if(**j **==** 6**)**

**{**

cleardevice**();**

outtextxy**(**460**,** 100**,** "You have reached the maximum number of attempts!"**);**

outtextxy**(**460**,** 120**,** "Your best score is: "**);**

outtextxy**(**640**,** 100**,** player\_2\_sum\_of\_scoures\_label**);**

**if(**sum\_of\_score\_player\_1 **>** sum\_of\_score\_player\_2**)**

**{**

outtextxy**(**460**,** 140**,** "The player 1 is winner!"**);**

outtextxy**(**460**,** 160**,** "press 'q' to go back the menu"**);**

**}**

**else**

**if(**sum\_of\_score\_player\_1 **<** sum\_of\_score\_player\_2**)**

**{**

outtextxy**(**460**,** 140**,** "The player 2 is winner!"**);**

outtextxy**(**460**,** 160**,** "press 'q' to go back the menu"**);**

**}**

**else**

**if(**sum\_of\_score\_player\_1 **==** sum\_of\_score\_player\_2**)**

**{**

outtextxy**(**460**,** 140**,** "Same score!"**);**

outtextxy**(**460**,** 160**,** "press 'q' to go back the menu"**);**

**}**

**}**

**else**

**{**

sprintf**(**player\_2\_scores\_label**,** "%d"**,** score**);**

sprintf**(**player\_2\_sum\_of\_scoures\_label**,** "%d"**,** sum\_of\_score\_player\_2**);**

outtextxy**(**300 **+** j**\***gap\_2**,** 130**,** player\_2\_sum\_of\_scoures\_label**);**

**}**

**}**

/\* control to throw the javelin\*/

void throw\_javelin**(**char user\_control**)**

**{**

**if(**user\_control **==** 'w'**){**

initial\_pos\_x **=** pos\_x**;**

initial\_pos\_y **=** pos\_y**;**

moveto**(**initial\_pos\_x**,** initial\_pos\_y**);**

/\* set the velocity of the javelin\*/

**if(**x\_position **>** 100 **&&** x\_position **<** 200**)**

**{**

vel\_x **=** 15**;**

vel\_y **=** 15**;**

**}**

**else**

**if(**x\_position **>** 200 **&&** x\_position **<** 300**)**

**{**

vel\_x **=** 30**;**

vel\_y **=** 30**;**

**}**

**else**

**if(**x\_position **>** 300**)**

**{**

vel\_x **=** 45**;**

vel\_y **=** 45**;**

**}**

/\* draw the arc\*/

**do**

**{**

t **=** **(**pos\_x **-** initial\_pos\_x**)** **/** vel\_x**;**

pos\_y **=** **(**int**)(**initial\_pos\_y **-** vel\_y**\*** t **+** **(**gravity **\*** t **\*** t**)/**2**);**

lineto**(**pos\_x**,** pos\_y**,** 1**);**

update\_display**();**

pos\_x**++;**

**}**

**while(**pos\_y **<** 420**);**//reach the ground

calculate\_score**();**

/\* wait for 1 second\*/

Sleep**(**1000**);**

/\* clear frame\*/

clear\_person\_1\_pre\_frame**();**

x\_position **=** 100**;**

y\_position **=** 340**;**

stick\_person**();**

**}**

**}**

/\* control to throw the javelin, player 1\*/

void throw\_javelin\_person\_1**(**char user\_1\_control**)**

**{**

**if(**user\_control **==** 'w'**){**

initial\_pos\_x **=** pos\_x**;**

initial\_pos\_y **=** pos\_y**;**

moveto**(**initial\_pos\_x**,** initial\_pos\_y**);**

/\* set the velocity of the javelin\*/

**if(**x\_position **>** 100 **&&** x\_position **<** 200**)**

**{**

vel\_x **=** 15**;**

vel\_y **=** 15**;**

**}**

**else**

**if(**x\_position **>** 200 **&&** x\_position **<** 300**)**

**{**

vel\_x **=** 30**;**

vel\_y **=** 30**;**

**}**

**else**

**if(**x\_position **>** 300**)**

**{**

vel\_x **=** 45**;**

vel\_y **=** 45**;**

**}**

**do**

**{**

t **=** **(**pos\_x **-** initial\_pos\_x**)** **/** vel\_x**;**

pos\_y **=** **(**int**)(**initial\_pos\_y **-** vel\_y**\*** t **+** **(**gravity **\*** t **\*** t**)/**2**);**

lineto**(**pos\_x**,** pos\_y**,** 1**);**

update\_display**();**

pos\_x**++;**

**}**

**while(**pos\_y **<** 420**);**//reach the ground

calculate\_score\_Player\_1**();**

print\_player1\_score**();**

/\* wait for 1 second\*/

Sleep**(**1000**);**

clear\_person\_1\_pre\_frame**();**

x\_position **=** 100**;**

y\_position **=** 340**;**

stick\_person**();**

**}**

**}**

/\* control to throw the javelin, player 2\*/

void throw\_javelin\_person\_2**(**char user\_2\_control**)**

**{**

**if(**user\_2\_control **==** 'i'**){**

initial\_pos\_x\_2 **=** pos\_x\_2**;**

initial\_pos\_y\_2 **=** pos\_y\_2**;**

moveto**(**initial\_pos\_x\_2**,** initial\_pos\_y\_2**);**

/\* set the velocity of the javelin\*/

**if(**x\_position\_2 **>** 100 **&&** x\_position\_2 **<** 200**)**

**{**

vel\_x **=** 15**;**

vel\_y **=** 15**;**

**}**

**else**

**if(**x\_position\_2 **>** 200 **&&** x\_position\_2 **<** 300**)**

**{**

vel\_x **=** 30**;**

vel\_y **=** 30**;**

**}**

**else**

**if(**x\_position\_2 **>** 300**)**

**{**

vel\_x **=** 45**;**

vel\_y **=** 45**;**

**}**

**do**

**{**

t **=** **(**pos\_x\_2 **-** initial\_pos\_x\_2**)** **/** vel\_x**;**

pos\_y\_2 **=** **(**int**)(**initial\_pos\_y\_2 **-** vel\_y**\*** t **+** **(**gravity **\*** t **\*** t**)/**2**);**

lineto**(**pos\_x\_2**,** pos\_y\_2**,** 1**);**

update\_display**();**

pos\_x\_2**++;**

**}**

**while(**pos\_y\_2 **<** 820**);**//reach the ground

calculate\_score\_Player\_2**();**

print\_player2\_score**();**

Sleep**(**1000**);**

clear\_person\_2\_pre\_frame**();**

x\_position\_2 **=** 100**;**

y\_position\_2 **=** 740**;**

stick\_person\_2**();**

**}**

**}**

//illegal or game over

void game\_over**(**void**)**

**{**

initfont**();**

/\* If the stick person reach the red line, illegal\*/

**if(**x\_position **>=** 500**)**

**{**

cleardevice**();**

outtextxy**(**600**,** 300**,** "game over! Because of the illegal!"**);**

outtextxy**(**460**,** 320**,** "You are beyond the red line! Press 'q' to exist game."**);**

**if(**user\_control **==** 'q'**)**

**{**

closegraph**();**

**}**

**}**

**}**

/\*illegal or game over, in the multi-player mode\*/

void game\_over\_Player\_1**(**void**)**

**{**

initfont**();**

/\* If the stick person 1 reach the red line, illegal\*/

**if(**x\_position **>=** 500**)**

**{**

cleardevice**();**

outtextxy**(**860**,** 100**,** "game over! Because of the illegal!"**);**

outtextxy**(**860**,** 120**,** "You are beyond the red line! You have no choice."**);**

**}**

**}**

/\*illegal or game over, in the multi-player mode\*/

void game\_over\_Player\_2**(**void**)**

**{**

initfont**();**

/\* If the stick person 2 reach the red line, illegal\*/

**if(**x\_position\_2 **>=** 500**)**

**{**

cleardevice**();**

outtextxy**(**860**,** 100**,** "game over! Because of the illegal!"**);**

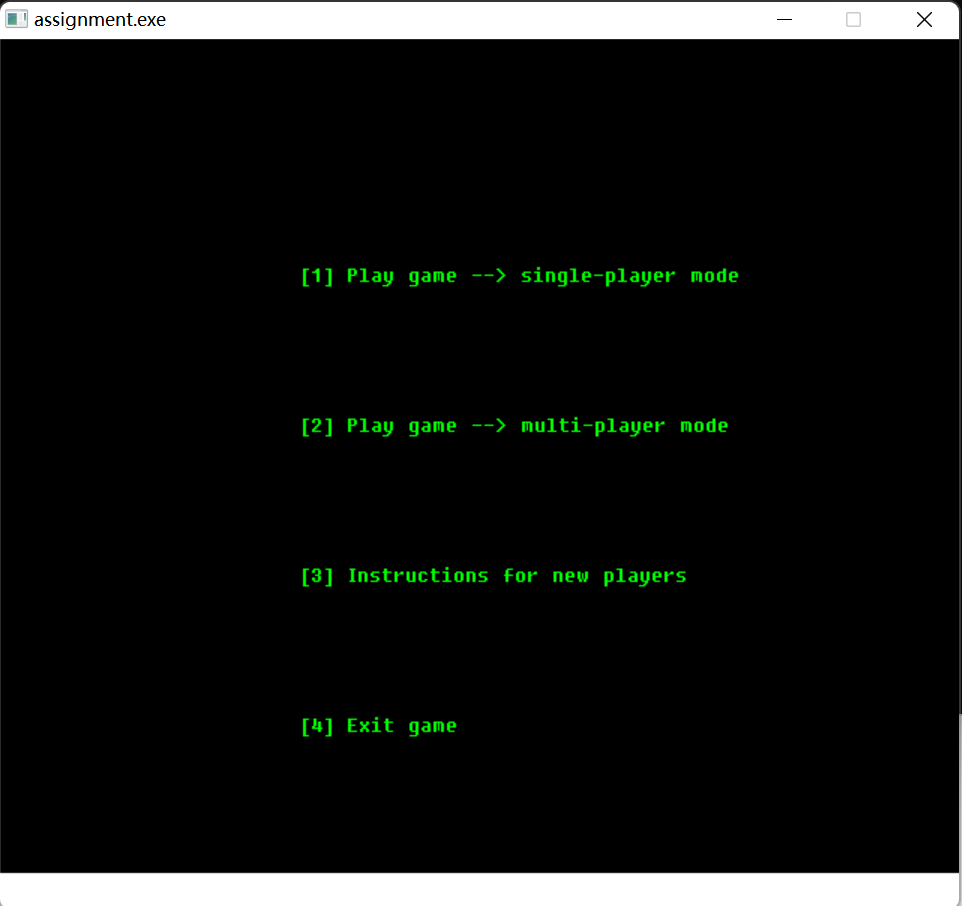
outtextxy**(**860**,** 120**,** "You are beyond the red line! You have no choice."**);**

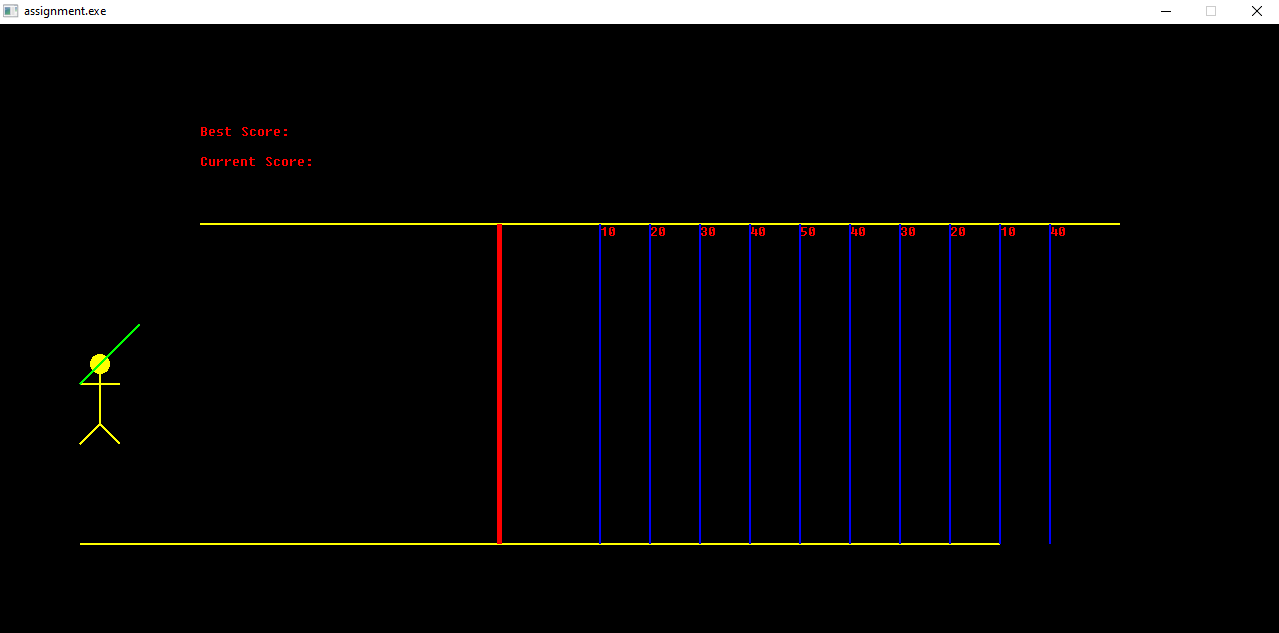
outtextxy**(**860**,** 140**,** "The winner is player 1."**);**

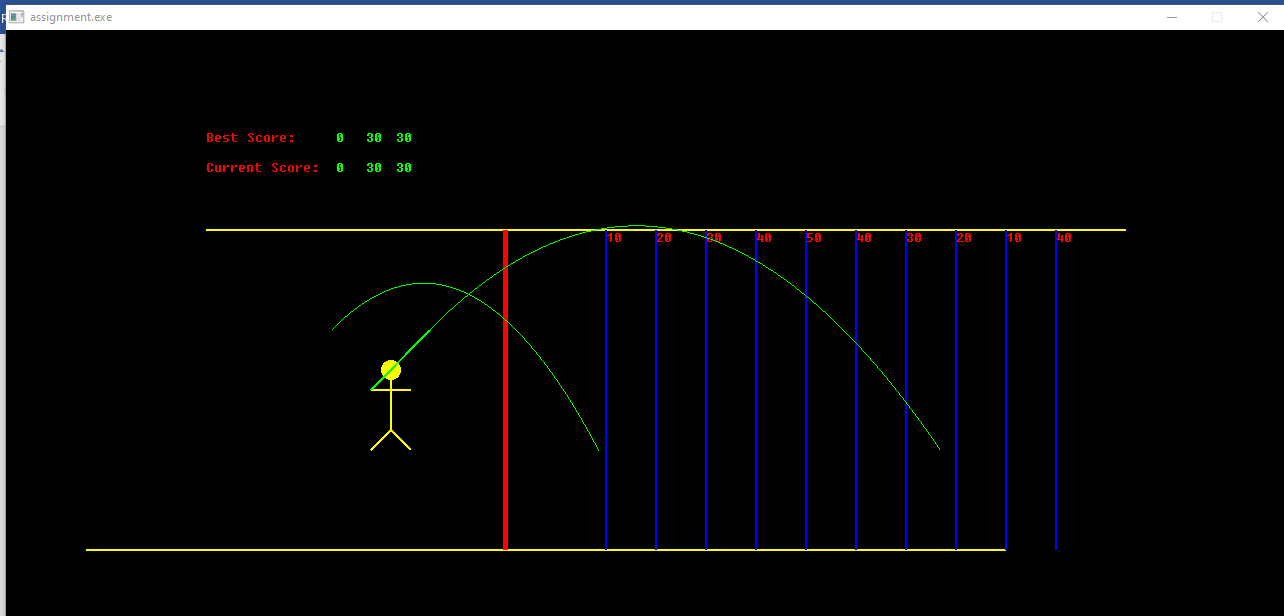
**}**

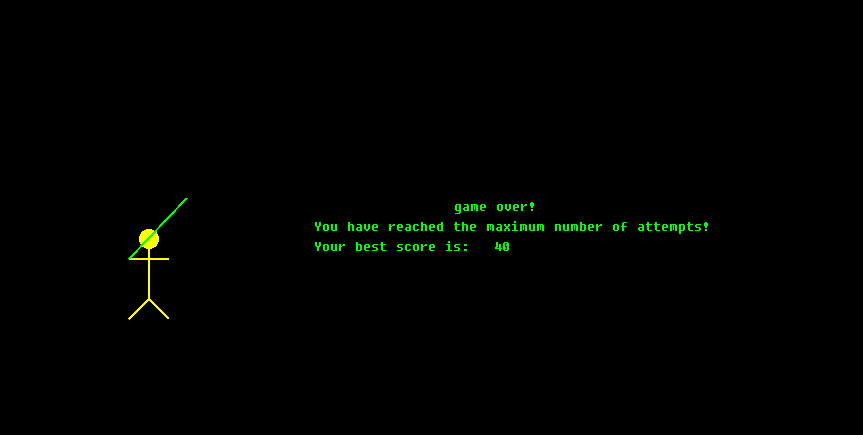
**}**

Start menu:

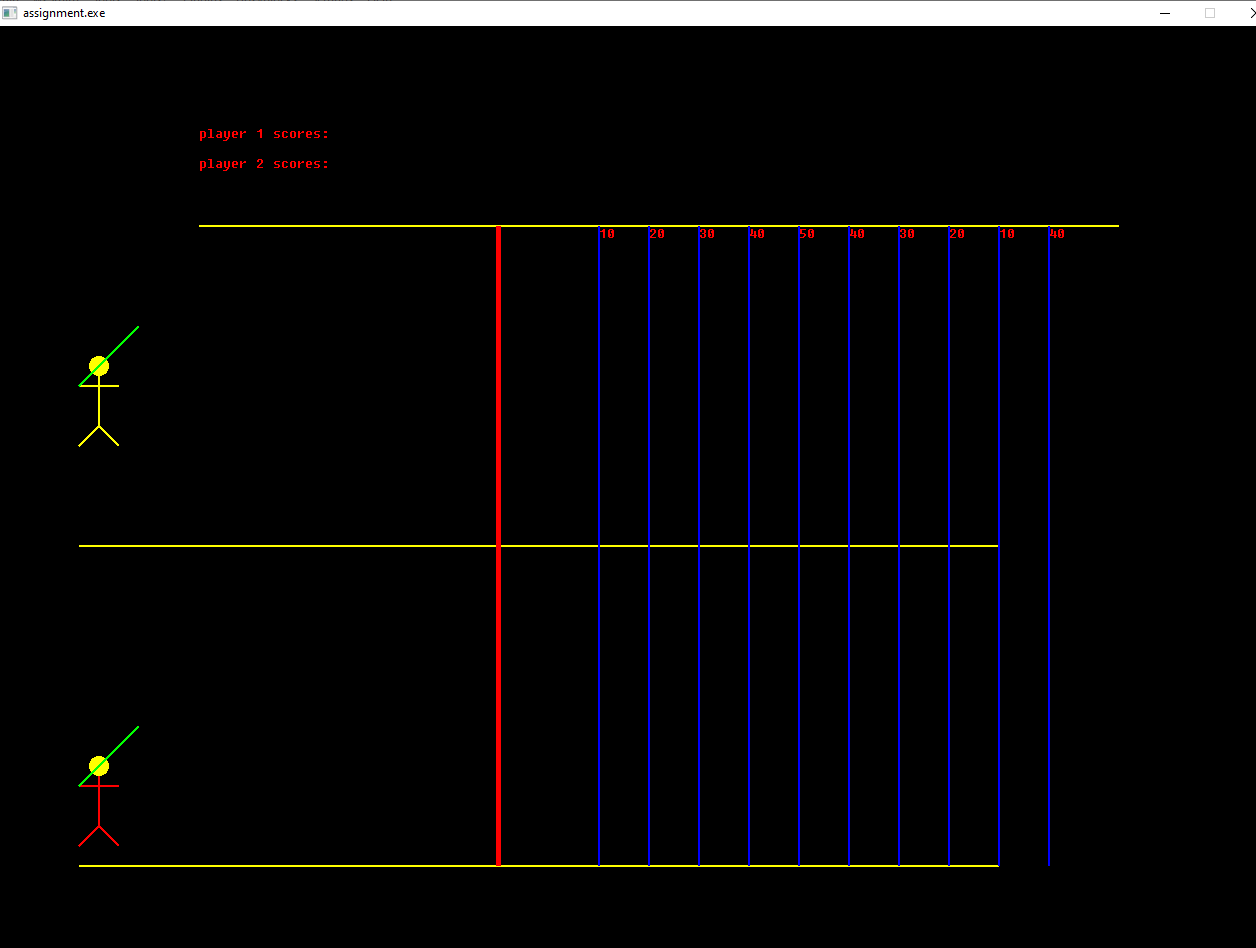




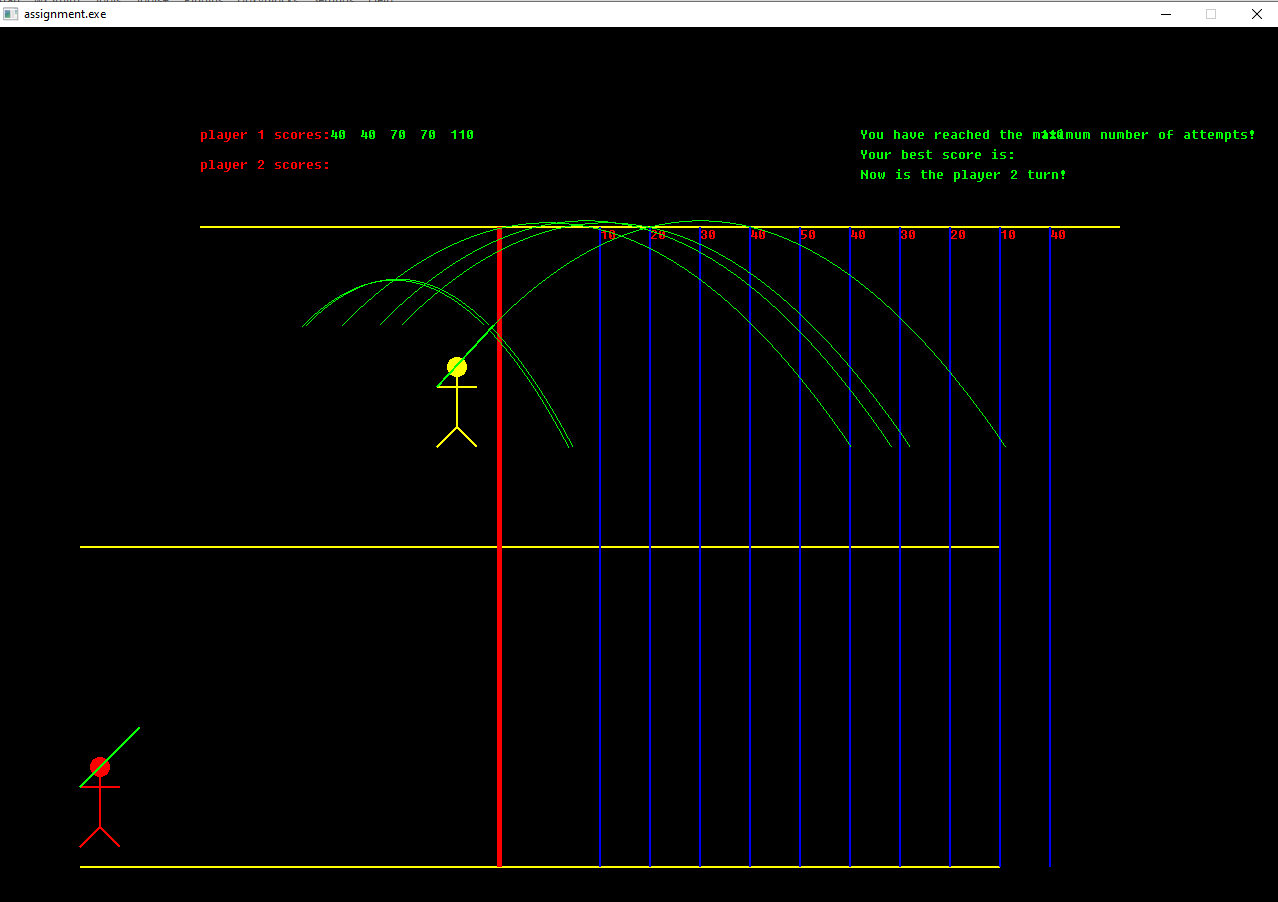




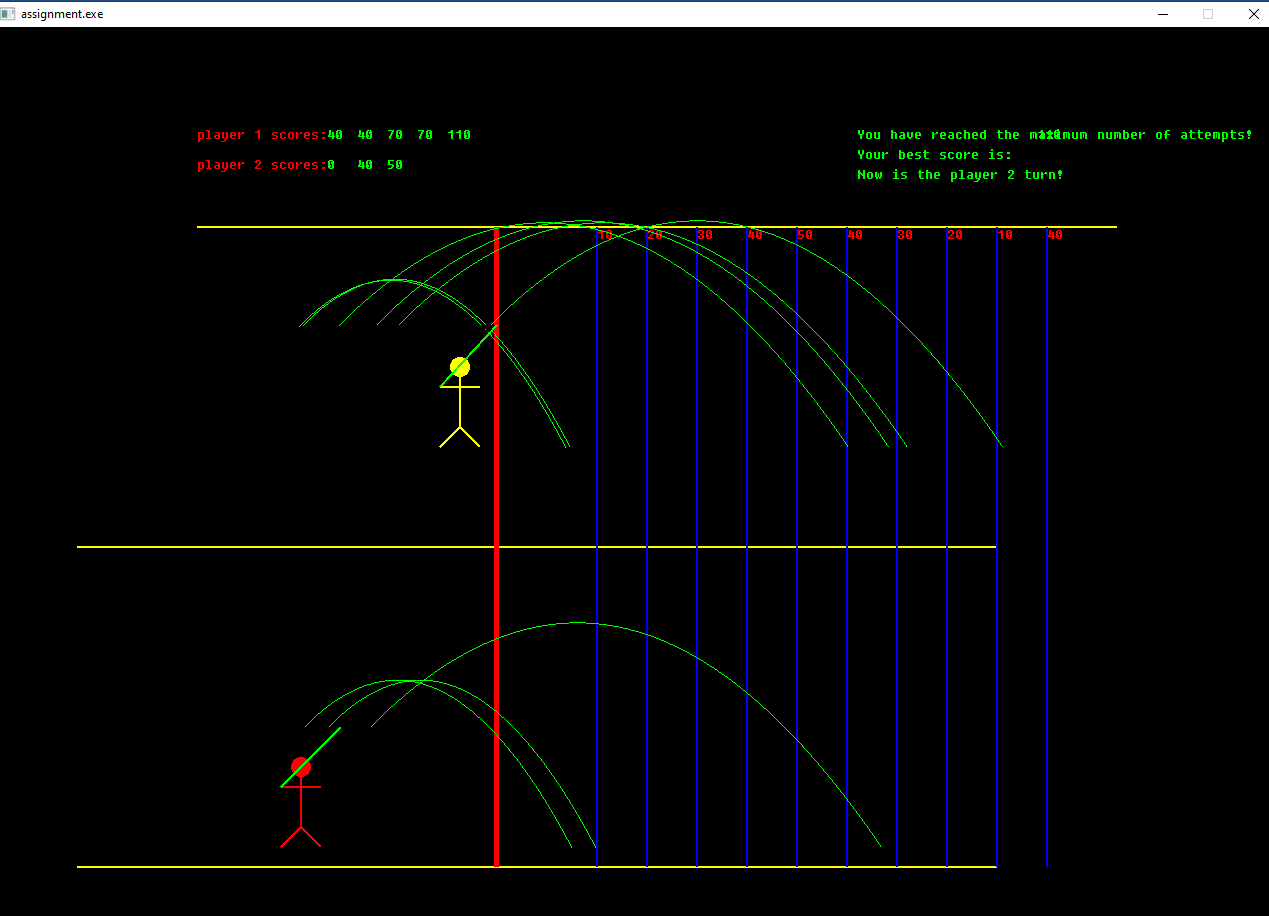
Multi-player mode:

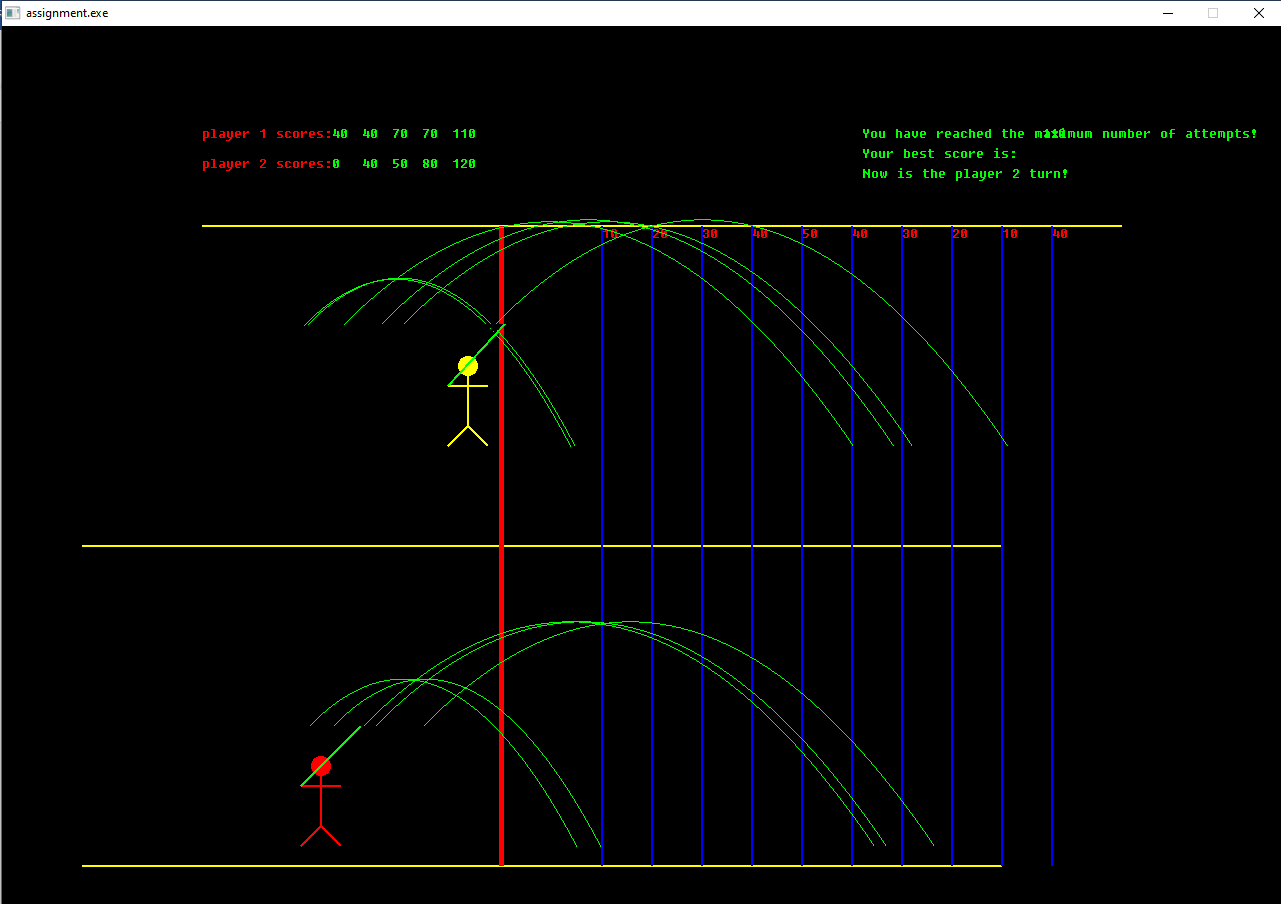
****

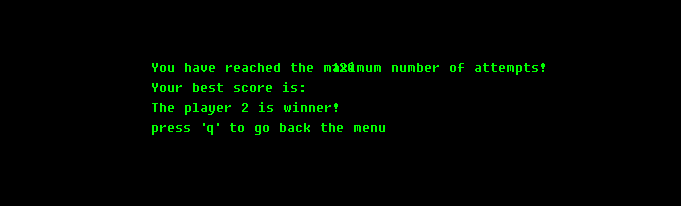
**First player:**

****

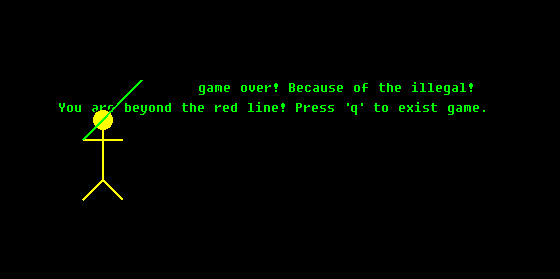
**Second player:**

****

****

****

If the stick man crosses the red line, it is a foul. (illegal)



# Testing and Verification

- Test strategy: When designing the program, I encapsulated the operations of drawing the image, throwing the javelin, and calculating the score into functions that can be called when executing the main program, which facilitates subsequent improvements and maintenance and enhances the readability of the code. This also increases the probability of success for the subsequent program to run properly. In the game, I am using a conditional judgment statement to determine whether a certain situation occurs, and if that condition is satisfied at the moment, then the operation is executed. By making such a judgment various conditions can be satisfied for the operation.

**Functional test:**

The program will be tested through functions and make sure every function works well. We can monitor the input and output of each function to verify whether this function outputs the result we want under specified conditions. By doing this, we can figure out which function has problems directly and fix it. The method we test each function is called “Black box testing”, as the name implies, is the simulation of the software testing environment as an invisible "black box". Data is input and output is observed to check that the software functions properly internally. When the test is launched, data is entered into the software and waited for the data to be output. If the data output is consistent with the expected data, then the software has passed the test.

**Unit test:**

Unit testing is the decomposition of the entire software into individual units and the subsequent testing of the units. This software testing basic approach is followed by the tester to test the unit of the program. It helps developers to know whether the individual unit of the code is working properly or not. We can divide different game modes as a different unit to test if it works well, or if it has some bugs.

# User Manual

The target audience for this program can be anyone older than 6 years old, with or without computer skills, who should be able to quickly understand the rules and operation of the game and get started playing.

**Installation instructions**: Copy this program (file name graphics1.c) to the Assignment-project->src folder and open the project file with Codeblocks. Press the Start button to run the program.

**System requirements**: The program is recommended for Windows systems (WIN7 and higher).

This game is intended for people aged six and above, regardless of whether you have good computer skills or not.

**Usage Instructions**: After opening the project file, you will see a game start menu, you can select which game mode you would like to play. If you in a game mode, use the ‘d’ or ‘l’ button to control the stick person’s movement, use the ‘w’ or ’ i’ button to control the throwing of the javelin, and the up and down buttons to control the angle of throwing. You will get a score when the javelin lands. Try to break your score.

# Demonstration/Summary

The implementation of the program is generally in line with the design specification because the program designed this time already has the most important functions, such as controlling the movement of the villain, throwing the javelin, drawing the running trajectory, and calculating the score. However, there are still some missing functions, such as using the mouse to control the game, controlling the angle of javelin throwing, background music, and two-player mode. In terms of program design, I only strictly follow the requirements given in the experiment to design the program, and there is not a lot of innovation, which I can continue to improve the game in the future to make it reach a higher level. In terms of the operation of the program, I can only say that sometimes it is not very good, although the program can achieve the main function, in the player control of the throwing javelin, the player needs to repeatedly press the W and D keys, after a while to throw the javelin. Therefore, this is the most important point that needs to be corrected in the subsequent improvements.