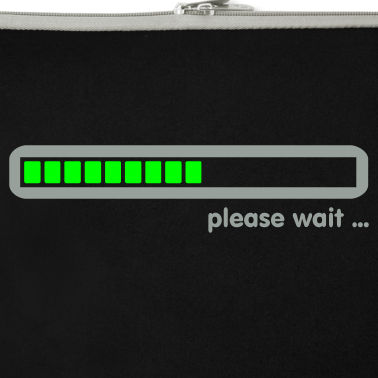


**FUNCTIONS USED**

* Void intro() – There is a page of loading like the one given below



There is also a page of progress bar like:



* Void ud() – It is used to draw the background for our game that is the line dividing objects and arrow. It also includes the arc from which a user can change the direction of the arrow to hit the desired object.

Functions like setcolor() and cleardevice() has been used in it.

* Void obj() – It has been used to initialize different objects for our game of different shapes.

Circles,rectangles,ellipse,diamond,lines, etc. has been used to make the best use of different figures available easily in graphics.

* Void score() – It is used to print the score corresponding to each object and the score flies and get set to the variable aim.

It also initializes a screen showing the aim of the game and no. of trials being available to the user.

* Void mix() – It includes the mixture of 3 functions as the name suggests i.e ud(),obj(),score().

So it reduces the code.



* Void move() – This function is the main head as the arrow pointing to the desired object is set only through this function.

User can use ‘a’ and ‘d’ to move left and right resp. and ‘s’ to move the arrow pointing to the object in the given direction to grab it.

* Void grasp() – This function has capability to grab an object.

It calls the move() function which returns the direction in which the arrow has to move.

According to the cases, the arrow follows a certain path as set and if the object hinders its path it carries along with it and the time taken is depending on the heaviness of the object. If no object is detected then it moves and comes back.

A user have the three chances to use the arrow in which he has to make a score of atleast 200$ to win.

Finally the object disappears as it is carried by line and screen is thus. Available for further palying.



<GRAPHICS.H> was used which includes the following functions which were used in the project “GOLD MINER” to give interesting graphics.

1. **Cleardevice()** - erasess the ebtire graphics and move the current position to home(0,0).
2. **Circle()** – draws a circle by taking three coordinates ,2 for coordinates and 1 for radius.

**Syntax**- gotoxy(int cordinate x,int c cordinate y)

1. **Arc()** - draws an arc by taking starting and ending angles besides taking coordinates and radius

**Syntax** – arc(int cordinate x,int cordinate y,stangle,endangle, int radius cordinate ).

1. **Gotoxy()** – moves the current position of ptr to a desired coordinates .

**Syntax** - gotoxy(int cordinate x, int cordinate y).

1. **Settextxy()** – moves the current position to the desired coordinates and also print the text.

**Syntax** – settextxy(int cordinate x,int coordinate y,”TEXT”).

1. **Ellipse()** – draws an ellipse by taking coordinates,starting and ending angles and width/height.

**Syntax** – ellipse(int cordinate x,int coordinate y,int stangle,int endangle,int radius).

1. **Line()** – draws a line by taking 4 coordinates for the start and end.

**Syntax**  - line(int cordinate x1,int coordinate y1, int cordinate x2,int coordinate y2).

1. **Rectangle()** – draws a rectangle by taking 4 parameters .

**Syntax** – rectangle(int cordinate x1,int coordinate x2, int cordinate y1,int coordinate y2).

1. **Setcolor()** – sets the desired colour for the text aur figure so given after this function.

**Syntax** – setcolor(colour/no.).

1. **Setbkcolor()** – sets the background colour as per the colour indicated inside it,it can either be a no. aur colour in caps.

**Syntax** – setbkcolor(colour/no.).

1. **Settextstyle()** – takes three parameters and is used for formatting d text aur figure.

**Syntax** – settextstyle(int font,int direction,int size).