ATOM MAN

A

Mini Project Report

Submitted in partial fulfilment of the Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

Nakkala Rachel 1602-19-737-030
Bhaskara Supraja 1602-19-737-059

Department of Information Technology
Vasavi College of Engineering
(Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31

2020

Vasavi College of Engineering
(Autonomous)

(Affiliated to Osmania University)
Hyderabad-500 031
Department of Information Technology

DECLARATION

We,Nakkala Rachel and Bhaskara Venkata Supraja, bearing hall ticket number 1602-19-737-030 and 1602-19-737-059, hereby declare that the project report entitled "**ATOM MAN",**Department of Information Technology, Vasavi College of Engineering,Hyderabad is submitted in partial fulfilment of the requirement for the award of degree of **Bachelor of Engineering** in **Information Technology.**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Nakkala Rachel
1602-19-737-030
Bhaskara Supraja
1602-19-737-059

ACKNOWLEDGEMENT

Self-confidence,hard work,commitment and planning are essential to carry-out any task. Possessing these qualities is sheer waste ,if an opportunity does not exist .So,we whole-heartedly thank Dr.S.V.Ramana, Principal,and Mr.Ram Mohan Rao,Head of the Department of Information Technology for their encouragement ,support and guidance in carrying out the project.

We thank our Project Guide, Mrs. Leelavathi Ma'am for her valuable advice and suggestions from time to time and for being a constant source of inspiration and guiding us in completing our project successfully.

Finally, we would also like to express our sincere thanks to all staff of Information Technology for their kind cooperation and timely help during the course of our project.

ABSTRACT

ATOM MAN

As the internet culture took the world over, the popularity of the games has exponentially increased. Nowadays games have been catching people's attention in making money and many people made gaming as a part of their hobby as well. Gaming has become a fun-filled activity. They have become a healthy source of entertainment. Elon Musk, the chief designer of SpaceX and the architect of Tesla has started his career with a simple game. So, the main idea behind this project is to build an efficient game application named **"ATOM MAN"**. Basically, Atom Man is a student who needs to reach his school. The user has to help him reach the destination. This application contains all the domain features of an ideal game such as graphics, scoreboard, number of lives.

ABOUT THE GAME

1. The atom man has got three chances to fight with the demon and make his way to the destination by collecting 9 books.
2. For every 3 books he collects, a general knowledge question pops up which he needs to answer. If he chooses the wrong option, he loses the game.
3.When the atom man touches the demon, he loses his life. The player also gets to choose the demon he wants to fight with.
4. One thing to remember is if he loses his first life due to a demon, he has two more lives. But, if he fails to answer the question, he loses the entire game.

The whole project is built in C language using SDL, SDL. image libraries which also include window creation and render creation. Data structures like arrays and structures are used. Functions are also used to make it a structured tool.

INTRODUCTION

INFORMATION ABOUT PROBLEM

Our application is a game, where a boy has to reach his destination(school), by collecting books, fighting with demons on the way to his school. For every 3 books he collects, he has to answer a question. If it turns out to be the right answer, he is good to go forward, else, he loses the game. He has got 3 chances to fight with demons.

REQUIREMENTS OF THE PROBLEM

To build this application, one has to be perfect in the C language and one has to have a basic idea on C libraries like SDL, TTF, Image. The application only works in editors like Visual Studio Code 2019. As these libraries are not available by default, one has to download all the libraries mentioned above and should change the properties of the project so that project can adapt to all the libraries. The major requirement is that the user has to be able to handle technical and logical errors
To run the application/ to play the game, the user has to know how to operate the atom-man using left, right, top, bottom keys. He should be aware of general knowledge and basic computer related topics to answer the questions.

WHAT WE WANTED TO ACHIEVE THROUGH THIS PROJECT

Gaming has become a major attention-seeker today. Elon Musk, the famous engineer of all time has started his career by gaming. 98% of programmers believe that games can be developed easily using C++ or Java. Our intention is to break that myth and shed some light on development of games using C. There are libraries in C like SDL where one can proceed to develop a gaming application. We also wanted to replicate the salient features of games like Dev/Mario and additional features to make our application wholly different from existing games.

TECHNOLOGY

SOFTWARE REQUIREMENTS

1. A powerful source code editor like Visual Studio Code 2019 as it helps in downloading libraries, debugging. DO NOT choose editors like VI Editor/ Sublime as they do not provide any kind of facilities to download libraries which are not default.
2. Complete skill on C Language.
3. A basic idea on libraries like SDL.h, SDL_image, time.h, SDL_ttf.h
4. .png files of images (brick, man, demon, clouds, trees, book as per the application)
5. .ttf file of any kind of font available on the internet(as per the choice of user)
6. Downloaded versions of the following libraries for Windows 10, 64-bit operating Systems.
    1. SDL.h,
    2. SDL_image
    3. time.h
    4. SDL_ttf.h
7. Inclusion of default libraries like stdio.h, stdlib.h, time.h, string.h
8. Basic knowledge on rules and syntax permitted by Visual Studio Code 2019(For example, scanf is not permitted in it, we have to use scanf_s instead)

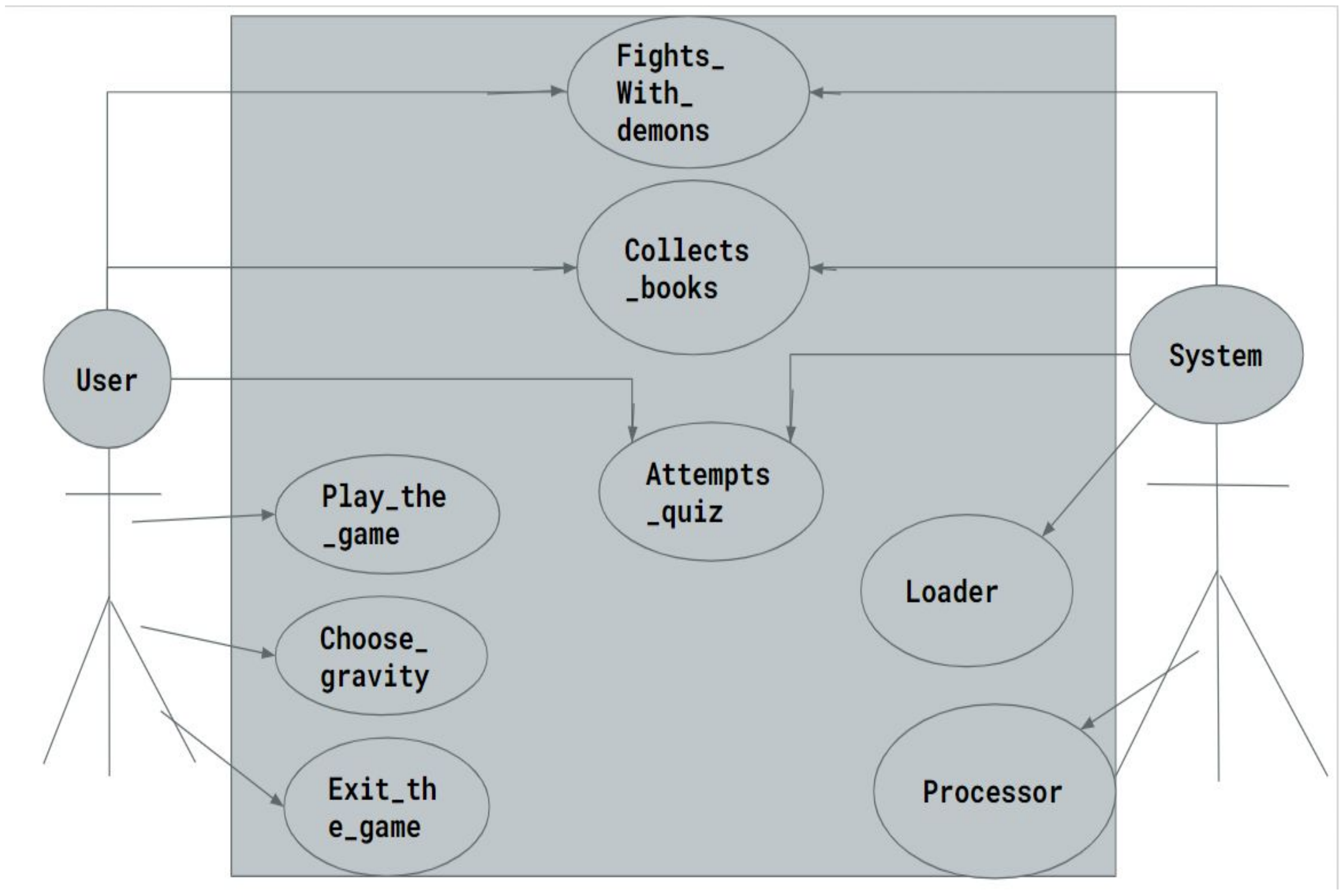
HARDWARE REQUIREMENTS

1. Personal Computer or a Laptop (prefer Windows version rather than Linux or Mac)

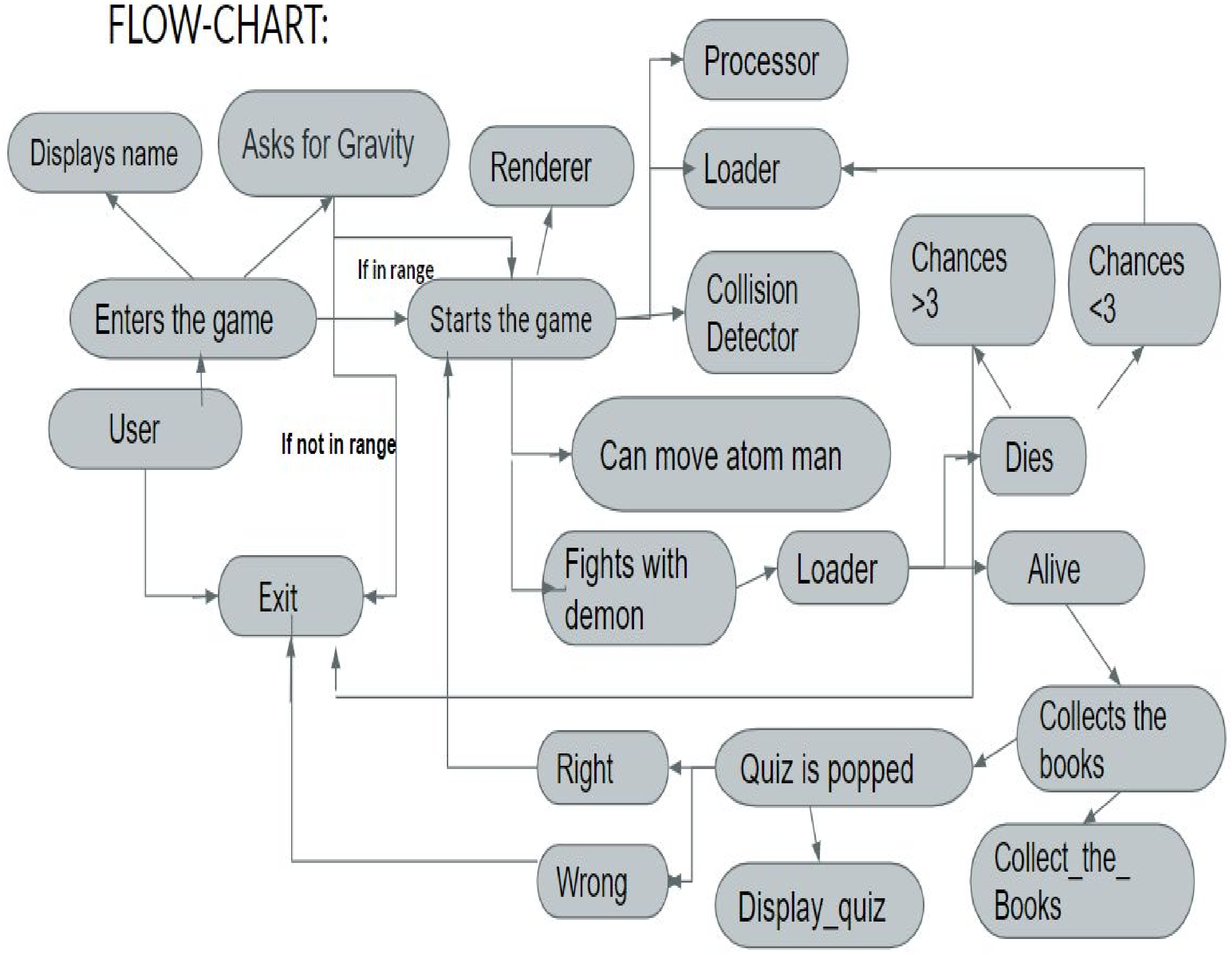
PROPOSED-WORK

DESIGN

USECASE DIAGRAM



FLOW-CHART:


Displays name
Asks for Gravity
Renderer
Processor
Loader
If in range
Enters the game
Starts the game
Collision Detector
Chances >3
Chances <3
User
If not in range
Can move atom man
Dies
Exit
Fights with demon
Loader
Alive
Collects the books
Right
Quiz is popped
Wrong
Display_quiz
Collect_the_ Books

IMPLEMENTATION

DESCRIPTION OF MAIN MODULES/COMPONENTS

1. init_status_lives

   It enables the introductory window prior to the game. It displays a blackcolored window with the name of the game on it

   CODE :

```
void init_status_lives(Gamestate* gameState)
{

   char str[100] = "";


   SDL_Colour white = { 255,255,255,255 };
            SDL_Surface*   tmp   =   TTF_RenderText_Blended(gameState->font,
"ATOM-MAN", white);
   gameState->label = SDL_CreateTextureFromSurface(gameState->renderer, tmp);
   SDL_FreeSurface(tmp);




}
```

2. Draw_status_lives

```
 void draw_status_lives(Gamestate* gameState)

{
   SDL_SetRenderDrawColor(gameState->renderer, 0, 0, 0, 255);
   SDL_RenderClear(gameState->renderer);

   //draw image:
  // SDL_Rect rect = { 320 - 40,240,48,48 };
     //SDL_RenderCopyEx(gameState->renderer,  gameState->manframes[0],  NULL,
&rect, 0, NULL,
    //  1);


   SDL_SetRenderDrawColor(gameState->renderer, 255, 255, 255, 255);

   SDL_Rect textRect = { 200,100,100,100 };
   SDL_RenderCopy(gameState->renderer, gameState->label, NULL, &textRect);
}
```

shut_down_lives

It destroys the enabled introductory window and lets the user play the game. It destroys the game after certain span of time(seconds)

3. Process

It controls the lives of the Atom-man

```
void process(Gamestate* game) {
   //add time
   game->time++;
   if (game->time > 120) {
      shutdown_status_lives(game);
      game->statusState = STATUS_STATE_GAME;
   }

   if (game->statusState == STATUS_STATE_GAME) {
       // if (game->man.isDead == 0) {
      Man* man = &game->man;
      man->y += man->dy;
      man->dy += GRAVITY;// makes atoman travel downwards



       game->scrollX = -game->man.x + 320;
      // }
      if (game->man.isDead && game->deathCountDown < 0) {
         game->deathCountDown = 120;
      }
      if (game->deathCountDown > 0) {
         game->deathCountDown--;
         if (game->deathCountDown < 0) {
            game->man.lives--;
            if (game->man.lives >= 0) {
               init_status_lives(game);
               game->statusState = STATUS_STATE_LIVES;
               game->time = 0;

                 //set it again
               game->man.isDead = 0;
               game->man.x = 100;
```

```
                    game->man.y = 240 - 40;
                    game->man.dx = 0;
                    game->man.dy = 0;
                    game->man.onLedge = 0;

                }

                else {
                    shutdown_status_lives(game);
                    game->statusState = STATUS_STATE_GAMEOVER;
                    game->time = 0;
                }
            }
        }
    }

}
```

4. collide2d

It checks if two certain objects have collided or not (Atom-man and demon, Atom-man and brick, Atom-man and books), returns True if collided, else, False.

CODE:

```
int collide2d(float x1, float y1, float x2, float y2, float wt1, float ht1,
   float wt2, float ht2) {
   return (!((x1 > (x2 + wt2)) || (x2 > (x1 + wt1)) || (y1 > (y2 + ht2)) || (y2 > (y1 + ht1))));
}
```

5. Dorernder2d

```
void dorender2(SDL_Renderer* renderer, Gamestate* game) {
  SDL_Rect rect = { game->scrollX + game->man.x,game->man.y,10,10 };
  SDL_RenderCopyEx(renderer, game->booktouch, NULL, &rect, 0, NULL, (game->time
% 20 < 10));
}
```

6.collissionDetect

Calls collide2d function to check the collision between two objects (Atom-man and demon, Atom-man and brick, Atom-man and books) and also makes the count of books and it pops a quiz if the count matches with the condition .

```
void collissionDetect(Gamestate* game, SDL_Renderer* renderer, SDL_Window*
window ) {

   //check the collission of atom-man and demon
   for (int i = 0; i < 100; i++) {
      if (game->man.freqdead[i] == 0)
      if (collide2d(game->man.x, game->man.y, game->stars[i].x, game->stars[i].y
         , 48, 48, 64, 64)) {
         game->man.isDead = 1;//if they collide (man and demon)
        // printf("Look around!");
         game->man.countdead = game->man.countdead + 1;
         printf("\n%d MORE LIVES LEFT", 3 - game->man.countdead);
         game->man.freqdead[i]++;

         if (game->man.countdead == 3) {
            printf("\nYou lost!");
           // delay(2);
            SDL_Quit();
            exit(1);
         }


      }
   }

   for (int i = 0; i < 100; i++) {
      if (game->man.freq[i] == 0) {
                    if (collide2d(game->man.x, game->man.y, game->books[i].x,
game->books[i].y,
            48, 48, 70, 70)) {
                  //SDL_Window* window = SDL_CreateWindow("Getting Started",
SDL_WINDOWPOS_UNDEFINED,
             //  SDL_WINDOWPOS_UNDEFINED, 640, 480, 0);
                     //SDL_Renderer* renderer = SDL_CreateRenderer(window, -1,
SDL_RENDERER_ACCELERATED);
            //doRender2(renderer, &game);
            game->man.isTouch = 1;
            // printf("keep going buddy!");
            game->man.county = game->man.county + 1;
            //SDL_Rect rect = { game->scrollX + 200,200,10,10 };
            //SDL_RenderFillRect(renderer, &rect);
                  //SDL_RenderCopyEx(renderer, game->booktouch, NULL, &rect, 0,
NULL, (game->time % 20 < 10));
```

```
printf("\nBOOK COUNT: %d", game->man.county);

game->man.freq[i]++;

if ((game->man.county % 3 == 0) && (game->man.county != 0)) {
   int a;
   int r;
   if (game->man.county == 3) {
      r = rand() % 3;
      switch (r) {
      case 0:
            printf("\nHow many numbers are binary numbers made of ?");
            printf("\n 1. 1 2.2 3.3 4.4");
            printf("\n Please enter your response");

            scanf_s("%d", &a);
            if (a != 2) {
               printf("Sorry, try again");
               SDL_Quit();
               exit(1);

            }
            break;

      case 1:
            printf("\nWhat is the national game of India ?");
            printf("\n 1. Hockey 2.Basketball ");
            printf("\n 3. Cricket 4. Kabbaddi");
            printf("\n Please enter your response");

            scanf_s("%d", &a);
            if (a != 1) {
               printf("Sorry, try again");
               SDL_Quit();
               exit(1);

            }
            break;

      case 2:
            printf("\nPortable computer is also called as ?");
            printf("\n 1. Laptop 2.Computer ");
            printf("\n 3. Adapter 4. Monitor");
            printf("\n Please enter your response");
```

```
                scanf_s("%d", &a);
                if (a != 1) {
                    printf("Sorry, try again");
                    SDL_Quit();
                    exit(1);

                }
                break;


        }
    }
    if (game->man.county == 6) {
        r = rand() % 2;
        switch (r) {
        case 0 :
                printf("\nWho invented computer ?");
                printf("\n 1. Charles Babbage 2.GrahamBell");
                printf("\n 3. Marconi 4. Tesla");
                printf("\n Please enter your response");

                scanf_s("%d", &a);
                if (a != 1) {
                    printf("Sorry, try again");
                    SDL_Quit();
                    exit(1);

                }
                break;

        case 1:
                    printf("\nThe display screen image that is used to provide visual
output from a computer is?");
                printf("\n 1. Monitor 2.CPU");
                printf("\n 3. Mouse 4.  AC adapter");
                printf("\n Please enter your response");

                scanf_s("%d", &a);
                if (a != 1) {
                    printf("Sorry, try again");
                    SDL_Quit();
                    exit(1);
```

```
            }
            break;

    }

}

if (game->man.county == 9) {
    r = rand() % 2;
    switch (r) {
    case 0:
        printf("\nWhat is the main circuit board of computer ?");
        printf("\n 1. Mother board 2. Chipset");
        printf("\n Please enter your response");

        scanf_s("%d", &a);
        if (a != 1) {
            printf("Sorry, try again");
            SDL_Quit();
            exit(1);

        }
        break;
    case 1:
        printf("\nWhat is the national bird of India ?");
        printf("\n 1. Penguin 2. Peacock");
        printf("\n 3. Parrot 4. Sparrow");
        printf("\n Please enter your response");

        scanf_s("%d", &a);
        if (a != 2) {
            printf("Sorry, try again");
            SDL_Quit();
            exit(1);

        }
        break;


    }
}
```

```
                if (game->man.county == 9) {
                    printf("CONGRATS YOU WON!!");
                    SDL_Quit();
                    exit(1);
                }
            }









        }
    }
}

//To ensure that the man does not go inside the brick
for (int i = 0; i < 100; i++) {
    float mw = 49, mh = 48;//man's width and height
    float mx = game->man.x;
    float my = game->man.y;
    float bx = game->ledges[i].x;
    float by = game->ledges[i].y;
    float bw = game->ledges[i].w;
    float bh = game->ledges[i].h;

    if ((my + mh) > by && my < (by + bh)) {
        //if man touches the right edge:
        if (mx < (bx + bw) && (mx + mw) >(bx + bw)) {
            game->man.x = bx + bw;
            //bring the man to the correct position
            mx = bx + bw;

        }

        //check for left edge
        else if ((mx + mw) > bx && mx < bx) {
            //correct the coordinate x
            game->man.x = bx - mw;
            mx = bx - mw;
        }
    }
```

```
            if ((mx + mw) > bx && mx < (bx + bw)) {

                if (my < (by + bh) && my>by) {
                    game->man.y = by + bh;
                    //correct y
                    mx = bx + bw;

                    //if bumps the brick, make the velocity of man 0
                    game->man.dy = 0;
                    game->man.onLedge = 1;
                }

                //check if we are landing on the edge
                else if ((my + mh) > by && my < by) {
                    //correct the coordinate x
                    game->man.y = by - mh;
                    //if landed, stop the jump velocity
                    game->man.dy = 0;
                    game->man.onLedge = 1;
                }
            }



        }
    }
```

6. Loadgame

Loads the game, calls function: init_status_lives to load the game, keeps track of lives of Atom-man, number of books he has collected, loads the images of trees, books, Atom-man, clouds, demons on the gaming window.

```
void Loadgame(Gamestate* game) {
    int j;
    //loading fonts:
    game->font = TTF_OpenFont("OpenSans-Bold.ttf", 50);
    if (!game->font) {
        printf("Font not available");
        SDL_Quit();
        exit(1);
}
SDL_Surface* starsurface = NULL;
starsurface = IMG_Load("star.png");
if (starsurface == NULL) {
   printf("could not load image");
   SDL_Quit();
   exit(1);
}
game->man.x = 320 - 80;
game->man.y = 320 - 20;//the lesser the y cor, the more upwards he will be
game->man.dy = 0;
game->man.onLedge = 0;
game->man.isDead = 0;
game->time = 0;
game->statusState = STATUS_STATE_LIVES;
game->label = NULL;
game->man.lives = 3;
game->deathCountDown = -1; //since we are at the start of the game!
game->man.isTouch = 0;
game->man.count = 0;
game->man.county = 0;
game->man.countdead = 0;

for (int i = 0; i < 100; i++) {
   game->man.freq[i] = 0;
}

for (int i = 0; i < 100; i++) {
   game->man.freqdead[i] = 0;
}



init_status_lives(game);

game->star = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);

starsurface = IMG_Load("man_walk.png");
if (starsurface == NULL) {
   printf("could not load image man_walk.png");
   SDL_Quit();
   exit(1);
}
```

```
game->manframes[0] = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);

starsurface = IMG_Load("man_ltb.png");
if (starsurface == NULL) {
   printf("could not load image man_ltb.png");
   SDL_Quit();
   exit(1);
}

game->manframes[1] = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);

starsurface = IMG_Load("brick.png");
game->brick = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);

game->scrollX = 0;

starsurface = IMG_Load("tree.png");
game->tree = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);

starsurface = IMG_Load("book.png");
game->book = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);

starsurface = IMG_Load("bush.png");
game->bush = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);

//if (game->man.isDead == 1)
//{
starsurface = IMG_Load("fire.png");
game->fire = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);

//}

starsurface = IMG_Load("book2.png");
game->booktouch = SDL_CreateTextureFromSurface(game->renderer, starsurface);
SDL_FreeSurface(starsurface);
```

```
//initialise stars
for (int i = 0; i < 100; i++) {
   game->stars[i].x = i * 500;//random number between 0 and 4000
   game->stars[i].y = rand() % 400;
}

//initialize the ledges
for (int i = 0; i < 100; i++) {
   game->ledges[i].w = 256;
   game->ledges[i].h = 64;
   game->ledges[i].x = i * 256;
   game->ledges[i].y = 400;
}

game->ledges[99].x = 350;
game->ledges[99].y = 200;

//initialise clouds
for (int i = 0; i < 100; i++) {
   game->trees[i].x = rand() % 6000;
   game->trees[i].y = 0;
}

for (int i = 99, j = 1; i > 50; i--, j++) {
   game->ledges[i].x = j * 550;
   game->ledges[i].y = 200 - rand() % 100;
}


//initialise books
for (int i = 0; i < 100; i++) {
   game->books[i].x = i * 400;
   game->books[i].y = rand() % 600;
}

//initilase bushes
for (int i = 0; i < 20; i++) {
   game->bushes[i].x = rand() % 6000;
   game->bushes[i].y = 235;
}
```

```
  //for (int i = 0; i < 20; i++) {
     //if (game->man.isDead == 1) {
   //      game->fires[i].x = game->man.x;
     //    game->fires[i].y = game->man.y;
     //}

 // }


}
```

7. ProcessEvent

Controls the movement of Atom-man(left, right, top, bottom)

```
//To make sure that window appears on the screen until we click escape key or any button,
//we use while loop here.
int ProcessEvent(SDL_Window* window, Gamestate* game)//passing reference of window
as an argument
{
  SDL_Event event;
  int done = 0;

  while (SDL_PollEvent(&event))
  {
    switch (event.type) {
    case SDL_WINDOWEVENT_CLOSE:
      // case SDL_WINDOWEVENT_CLOSE:
    {
      if (window)
      {
        SDL_DestroyWindow(window);
        window = NULL;
        done = 1;
      }
    }
    break;

    case SDL_KEYDOWN:
    {
          switch (event.key.keysym.sym)
          {
          case SDLK_ESCAPE:
          {
             done = 1;
          }break;
          case SDLK_RIGHT:
             game->man.x += 10;
             break;
          case SDLK_LEFT:
             game->man.x -= 10;
             break;
          case SDLK_UP:
             if (game->man.onLedge) {//he can jump only if he is on the ledge
                game->man.dy = -5;
                game->man.onLedge = 0;//once he jumps, he is not on ledge anymore..
             }

             //game->man.y -= 10;
             break;
          case SDLK_DOWN:

             game->man.y += 10;
             break;
          }
       }
       break;

       case SDL_QUIT:
       {
          done = 1;
       }
       break;
       }
    }
    //below code can be used instead of switch left and right keys.
    //state is basically an array. *state = is equal to state[] =
    //const Uint8* state = SDL_GetKeyboardState(NULL);
    //if (state[SDL_SCANCODE_RETURN]) {
     //  printf("RETURN IS PRESSED");
    //}

    return done;
}
```

8. doRender

It displays the images of ledges ,atom man ,demons ,books,clouds,trees on the game window at the specified position.

```
void doRender(SDL_Renderer* renderer, Gamestate* game)
{
  int h = 0;

  if (game->statusState == STATUS_STATE_LIVES) {
    draw_status_lives(game);
  }

  else if (game->statusState == STATUS_STATE_GAME) {
    //Set the draw color of renderer to blue
    SDL_SetRenderDrawColor(renderer, 0, 0, 100, 100);

    SDL_RenderClear(renderer);

    //set the color of the renderer to white
    SDL_SetRenderDrawColor(renderer, 510, 510, 510, 510);

    for (int i = 0; i < 100; i++) {
                              SDL_Rect ledgeRect = { game->scrollX +
game->ledges[i].x,game->ledges[i].y,game->ledges[i].w,game->ledges[i].h };
      SDL_RenderCopy(renderer, game->brick, NULL, &ledgeRect);
    }


    //Draw a rectangle at man's position
    SDL_Rect rect = { game->scrollX + game->man.x,game->man.y,76,48 };
    //SDL_RenderFillRect(renderer, &rect);
     SDL_RenderCopyEx(renderer, game->manframes[0], NULL, &rect, 0, NULL,
0);


    //Drawing the images of stars
    for (int i = 0; i < 100; i++) {
                               SDL_Rect starrect = { game->scrollX +
game->stars[i].x,game->stars[i].y,64,64 };
      SDL_RenderCopy(renderer, game->star, NULL, &starrect);
    }

    for (int i = 0; i < 100; i++) {
      //size of clouds
                              SDL_Rect   starrect   =   {   game->scrollX   +
game->trees[i].x,game->trees[i].y,70,70 };
      SDL_RenderCopy(renderer, game->tree, NULL, &starrect);

    }

    for (int i = 0; i < 100; i++) {
      //size of books
                              SDL_Rect   starrect   =   {   game->scrollX   +
game->books[i].x,game->books[i].y,70,70 };
      SDL_RenderCopy(renderer, game->book, NULL, &starrect);

    }

    for (int i = 0; i < 20; i++) {
      //size of bushes
                              SDL_Rect   starrect   =   {   game->scrollX   +
game->bushes[i].x,game->bushes[i].y,200,200 };
      SDL_RenderCopy(renderer, game->bush, NULL, &starrect);

    }

   /* if (game->man.isDead == 1)
    {
      SDL_Rect rect = { game->scrollX + game->man.x,game->man.y,48,48 };
      //SDL_RenderFillRect(renderer, &rect);
            SDL_RenderCopyEx(renderer,  game->fire,  NULL,  &rect,  0,  NULL,
(game->time%20<10));

      SDL_Delay(1000);
      // printf("look around!");
      // init_status_lives(game);

       //  SDL_DestroyTexture(game->fire);
      SDL_Quit();
      exit(1);
      // printf("look around!");

    }*/
    //SDL_DestroyTexture(game->fire);
    if (game->man.isTouch == 1 ) {
      //  h = 0;
        //SDL_Rect rect = { game->scrollX + game->man.x,game->man.y,10,10 };
        SDL_Rect rect = { game->scrollX +200,200,10,10 };
        //SDL_RenderFillRect(renderer, &rect);
          SDL_RenderCopyEx(renderer, game->booktouch, NULL, &rect, 0, NULL,
(game->time % 20 < 10));
        // printf("keep going buddy");

       // h++;
      //  game->man.count++;
        init_status_lives(game);
    }

    /*for (int i = 0; i < 20; i++) {
        //size of bushes
                                        SDL_Rect    starrect    =    {    game->scrollX    +
game->fires[i].x,game->fires[i].y,10,10 };
        SDL_RenderCopy(renderer, game->fire, NULL, &starrect);

    }*/




    //Clear the renderer with the draw color
    //SDL_RenderPresent(renderer);

        //Update the renderer which will show the renderer cleared by the draw color
which is green
    //When we are done with drawing, this function presents us the screen
  }
  SDL_RenderPresent(renderer);
 // return game->man.count;

}
```

9. main

Calls all the above functions along with displaying quizzes when needed(when the number of collected books are multiples of 3 except 0), enables the user to choose the gravity(the acceleration with which the Atom-man can move up), lets the user answer the question prior to the game. If the answer chosen turns out to be an incorrect one, he loses the game.

```
int main() {
   //int number;
   int a;
   int choice;
   printf("**********************************************");
   printf("\nHello, press 1 to enter the game, any other key to exit the game!");
   scanf_s("%d", &choice);

   if (choice == 1)
   {

      printf("\nHello, welcome to the game");

      printf("\nChoose the gravity in the range 0.04f and 0.06f:");


      float g;
      scanf_s("%f", &g);
      if ((g > 0.06) || (g < 0.04)) {
         printf("\n Please select a valid value for the gravity");
      }
      else {

         printf("Answer the question correctly to enter into the game!");
         printf("\nAnswer the question below to enter the game: ");
         printf("\n What is the Capital city of India?");
         printf("\n1. Mumbai 2. Delhi");
         printf("\n3. Karnataka 4. Chennai");
         printf("\nPlease enter your option: ");
         scanf_s("%d", &a);



         int c;
         //scanf_s("%d", &a);
         if (a == 2) {
            void print();
            SDL_Init(SDL_INIT_VIDEO);//initialise SDL2
           // printf("Enter an integer: ");

            // reads and stores input
            //scanf_s("%d", &number);
```

```
        // displays output
        //printf("You entered: %d", number);
              SDL_Window* window = SDL_CreateWindow("Getting Started",
SDL_WINDOWPOS_UNDEFINED,
          SDL_WINDOWPOS_UNDEFINED, 640, 480, 0);

        Gamestate gamestate;

        //Initializing Random number generator
            srand(time(0)); //for the coordinates of stars., to be type casted into int
datatype

           //Create a renderer for the window created above, with the first display
driver present
        //and with no additional settings
                 SDL_Renderer* renderer = SDL_CreateRenderer(window, -1,
SDL_RENDERER_ACCELERATED);
        int done = 0;

        gamestate.renderer = renderer;

        //initialise the font system
        TTF_Init();
        //init_status_lives(&gamestate);
        Loadgame(&gamestate);

        Man man;
        //initialising man's components as same as rectangle
        man.x = 220;
        man.y = 140;

        // printf("hello");





        while (!done)
        {

          //check for process events
          done = ProcessEvent(window, &gamestate);

          process(&gamestate);
```

```
//collission detection
collissionDetect(&gamestate, renderer, window);

//Render Display
doRender(renderer, &gamestate);
/* if (c == 1) {
    int g;
    int h, i, l, m, o;
    printf("\nHow many continents are there in the world? ");
    printf("\n1. 6 2. 7 3. 5 4. 9");
    scanf_s("%d", &g);


    printf("\nHow many countries are there in India ");
    printf("\n1. 22 2. 28 3. 29 4. 27");
    scanf_s("%d", &h);


    printf("\nWho wrote Ramayana? ");
    printf("\n1. Valmiki 2. Potana 3. Vedavyasa ");
    scanf_s("%d", &i);




    if ((g != 2) || (h != 3) || (i != 1)) {
        printf("Sorry, wrong answer");
        SDL_Quit();
        exit(1);
    }
 }*/
 /* else   if (c == 2) {
     int g;
     printf("1+1?");
     scanf_s("%d", &g);
     if (g != 2) {
         printf("wrong ans");
         SDL_Quit();
         exit(1);
     }
  }*/

  //Pause for 3 seconds (or 3000 milliseconds)
```

```
                SDL_Delay(10);

            }

            //Destroy the created texture
            SDL_DestroyTexture(gamestate.star);

            if (gamestate.label != NULL) {
                SDL_DestroyTexture(gamestate.label);
            }
            TTF_CloseFont(gamestate.font);

            //Destroy the renderer created above
            SDL_DestroyRenderer(renderer);

            //Destroy the window created above
            SDL_DestroyWindow(window);

            TTF_Quit();

            //Close all the systems of SDL initialized at the top
            SDL_Quit();

            return 0;
        }
        else {
            printf("Sorry, Try again");
        }
    }
  }


  else {
  printf("Exited from the game successfully!");
}
}
```

GITHUB LINKS:
https://github.com/RachelNakkala/C-Project,
https://github.com/Suprajabhaskara2903/GAME_ATOM-MAN

LOGIC TO BE HIGHLIGHTED

We have used "Frequency Arrays" for books to ensure that the count of books doesn't go wrong when collided with atom man. We have used structures to represent Atom-man, trees, books etc and we have included libraries like sdl, sld_image, sdl_ttf

```
#include <stdio.h>
#include <string.h>
#include <SDL.h>
#include <SDL_image.h>
#include <time.h>
#include <SDL_ttf.h>
#include <stdlib.h>



#define STATUS_STATE_LIVES 0
#define STATUS_STATE_GAME 1
#define STATUS_STATE_GAMEOVER 2

#define GRAVITY 0.04f //gravity constant
//the lesser the gravity constant, the slower will be the falling of the man
#define ACC 0.04f
typedef struct {
   float x, y;
   int lives;
   char* name;
   float dy;
   float dx;
   int onLedge;
   int isDead;
   int isTouch;
   int count;
   int freq[100];
   int county;
```

```
    int freqdead[100];
    int countdead;
}Man; //name of the structure is man

typedef struct {
    int x, y;

 }Reward;

typedef struct {
    int x, y;
}Star;

typedef struct {
    int x, y, w, h;

}Ledge;

typedef struct {
    int x, y;
}Book;

typedef struct {
    int x, y;
}Tree;

typedef struct {
    int x, y;
}Bush;

//typedef struct {
  //  int x, y;
//}Fire;


typedef struct {


    float scrollX;
    //creating players (man)
    Man man;

    //stars
    Star stars[100];
    //adding ledges
    Ledge ledges[100];

    Tree trees[100];

    Book books[100];

    Bush bushes[20];

    Reward rewards[100];

    //Fire fires[20];

     //for adding images
    SDL_Texture* star;

    //Images
    SDL_Texture* manframes[2];
    SDL_Texture* brick;

    SDL_Texture* label;//for text
    SDL_Texture* tree;
    SDL_Texture* book;
    SDL_Texture* bush;
    SDL_Texture* fire;
    SDL_Texture* booktouch;
    SDL_Texture* reward;

    TTF_Font* font;

    int time;
    int statusState;
    int deathCountDown;

    SDL_Renderer* renderer;
}Gamestate;
```

TESTING

TestCase:1
If the gravity chosen by the user crosses the range specified we are not allowed to play the game.

TestCase:2
If we choose to play the game with selecting a gravity in the given range and the system displays a quiz question if the user answers it wrong then he/she is out of the game.

TestCase:3
If we choose to play the game with selecting a gravity in the given range and the system pops a quiz question which need to be answered correctly then we can continue playing ,in the play if the atom man collides with demon he loses a life (of given three
lives) and also the system displays the left over lives,if all the life lifes get over, the user will be out from the game,even though he answers the quiz question correctly.

TestCase:4
If we choose to play the game with selecting a gravity in the given range and then the system pops a quiz question which need to be answered correctly then we continue playing ,in the play if the atom man collides with demon he loses a life (of given three lives) and also the system displays the left over lives,and if the atom man collides with books the book count will be displayed ,if the book count reaches three then again a question will be displayed ,if the selected answer is wrong then the user loses the whole game.

TestCase:5
If we choose to play the game with selecting a gravity in the given range and then the system pops a quiz question which need to be answered correctly then we continue playing ,in the play if the atom man collides with demon he loses a life (of given three lives) and also the system displays the left over lives ,and if the atom man collides with books the book count will be displayed ,if the book count reaches three then again a question will be displayed (general and computer based questions) ,if the user answers it correctly then he can continue playing and when again the count increases by another multiple of three the quiz question will be displayed and if he selects an incorrect option, he loses the game.

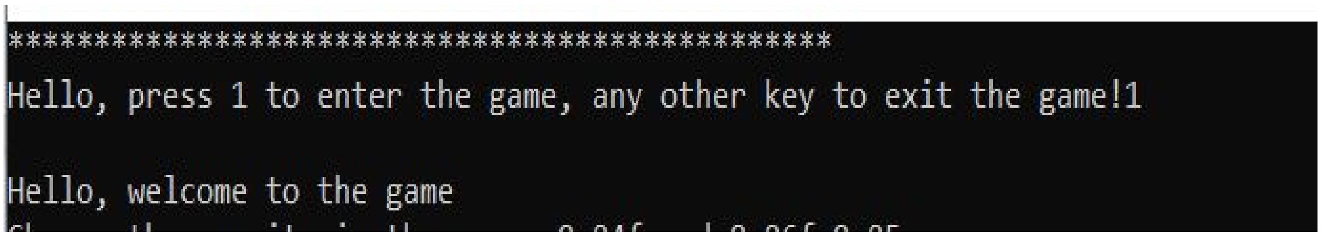
TestCase:6

If we choose to play the game with selecting a gravity in the given range and then the system pops a quiz question which need to be answered correctly then we can continue playing ,in the play if the atom man collides with demon he loses a life (of given three lives) and also the system displays the left over lives ,and if the atom man collides with books the book count will be displayed ,if the book count reaches three then again a quiz question will be displayed (general and computer based questions) ,if the user answers it correctly then he can continue playing and when again the count increases by another three the quiz question will be displayed and this continues for another multiple of three books(9) ,in all the cases if the user answers the quiz correctly he/she will win the game. He also has to remember that he should not lose more than 2 lives to win the game.

RESULTS

The game application provides the user to choose the gravity(the acceleration with which the user can operate the Atom-man to move upwards unlike in any other video games. The user will also be tested regarding his/her general knowledge and basic computer knowledge along with fighting with demons which isn't present in other games.

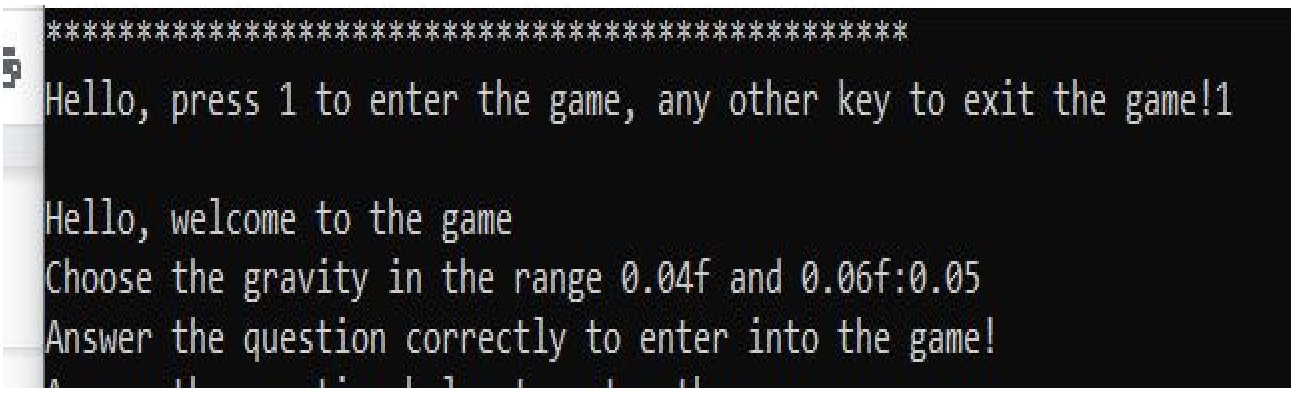
1. Entering the game



```
************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
```

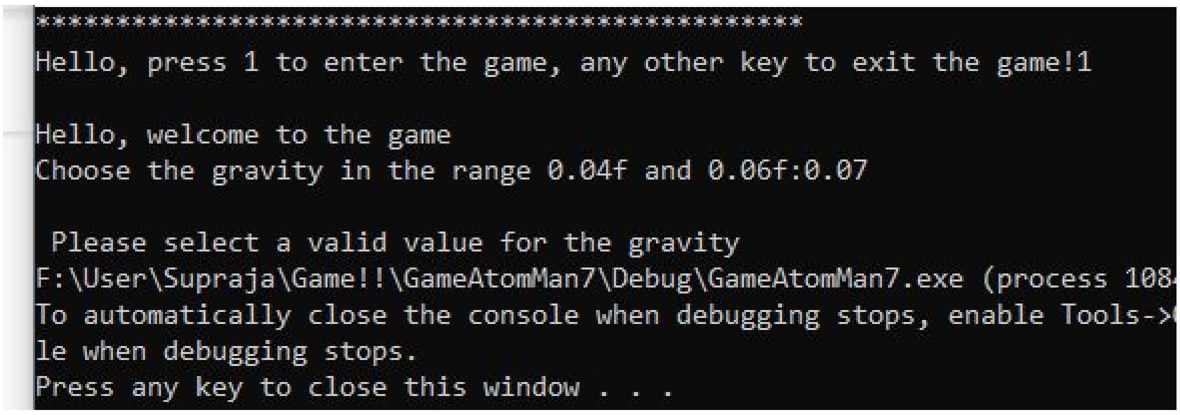
2.Choosing the correct value of gravity



```
**************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.05
Answer the question correctly to enter into the game!
```

3.Choosing the incorrect value of the gravity

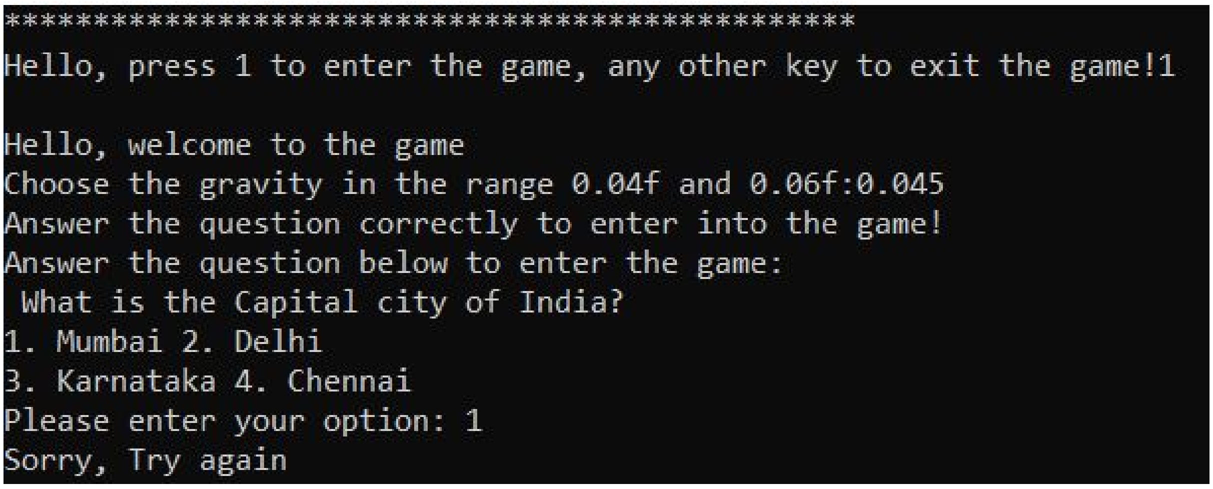


```
************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.07

 Please select a valid value for the gravity
F:\User\Supraja\Game!!\GameAtomMan7\Debug\GameAtomMan7.exe (process 108
To automatically close the console when debugging stops, enable Tools->
le when debugging stops.
Press any key to close this window . . .
```

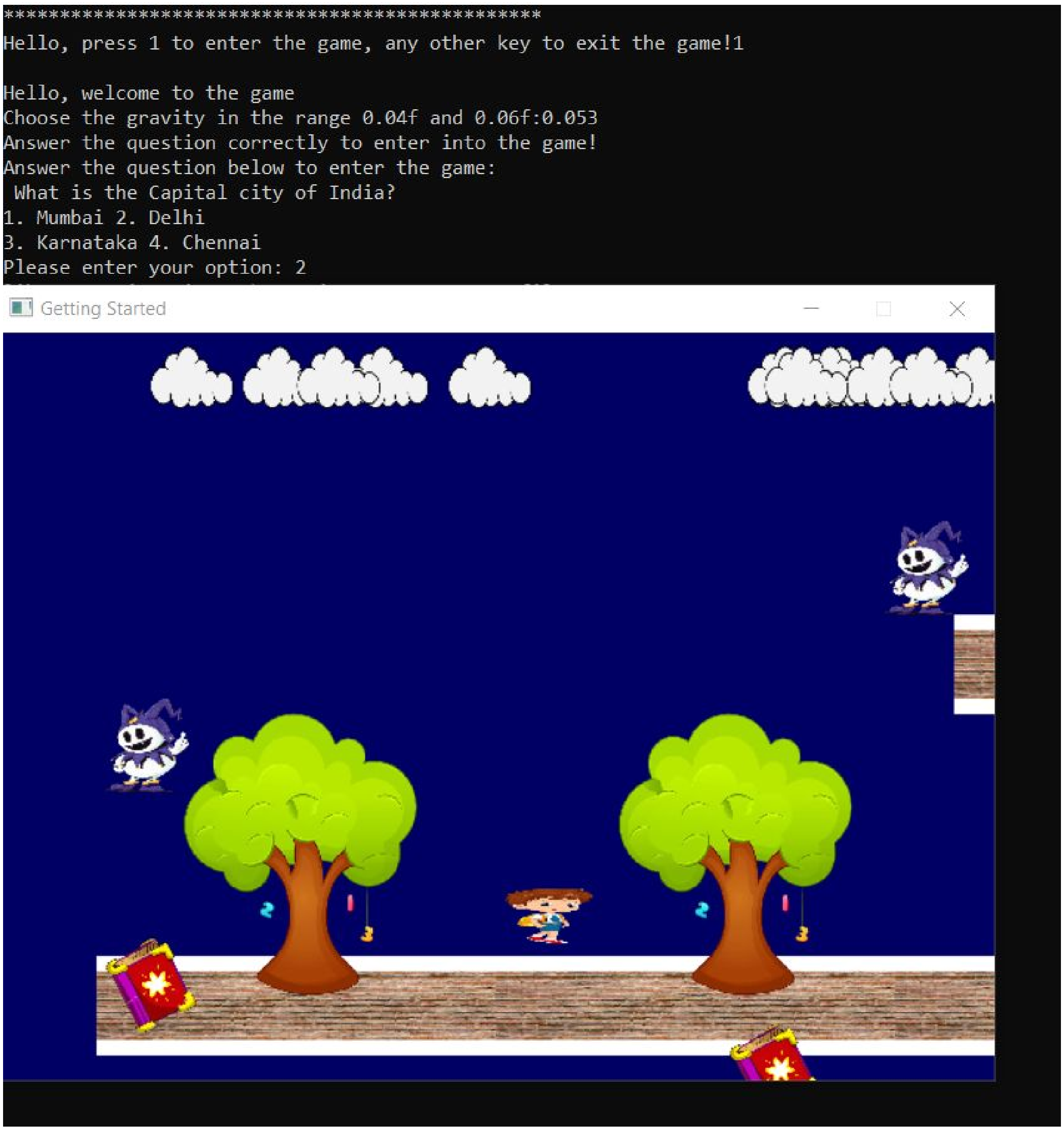
4.Choosing the incorrect option for prerequisite question



```
************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.045
Answer the question correctly to enter into the game!
Answer the question below to enter the game:
 What is the Capital city of India?
1. Mumbai 2. Delhi
3. Karnataka 4. Chennai
Please enter your option: 1
Sorry, Try again
```

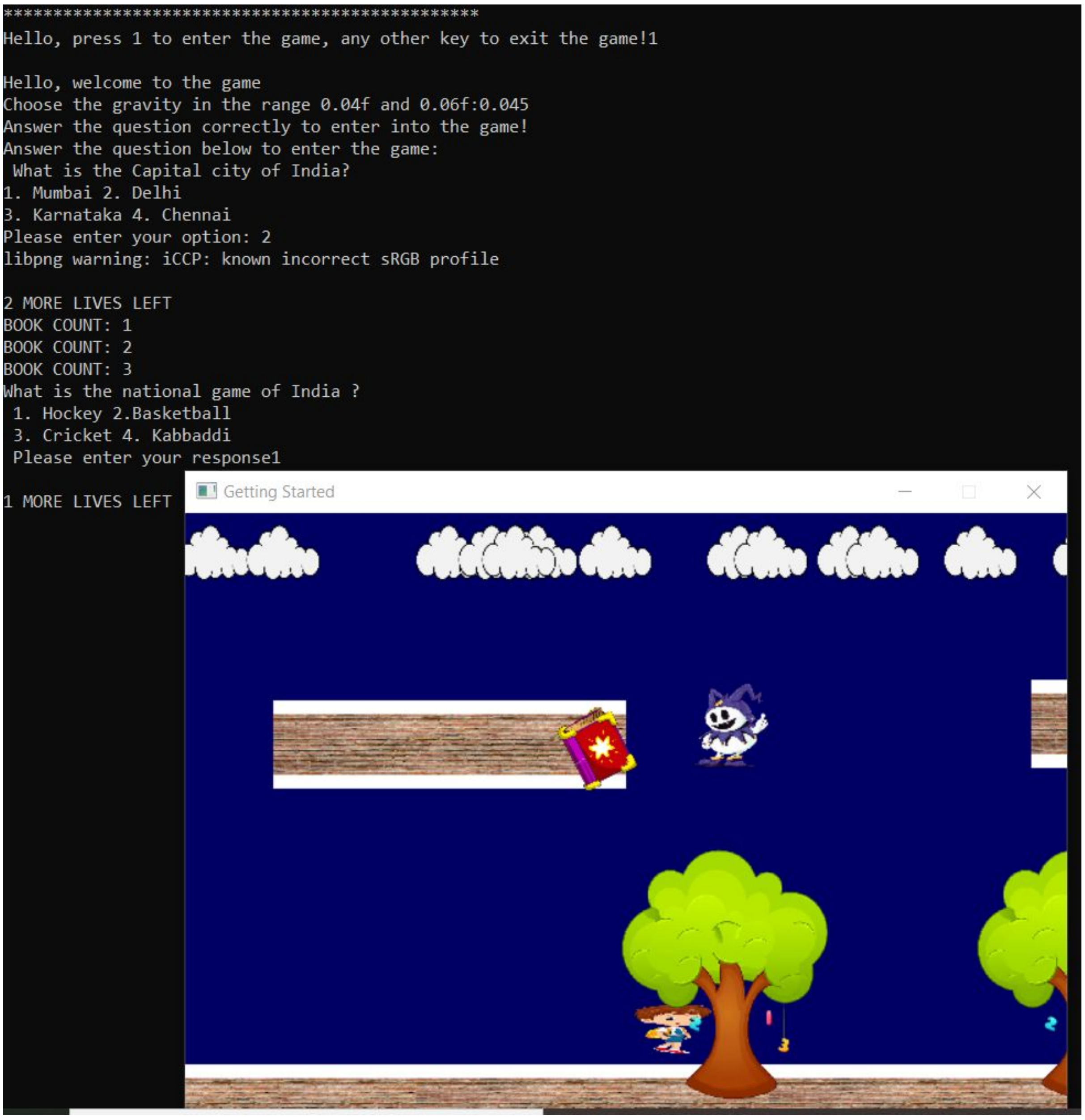
5.Choosing the correct option for prerequisite question



```
************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.053
Answer the question correctly to enter into the game!
Answer the question below to enter the game:
 What is the Capital city of India?
1. Mumbai 2. Delhi
3. Karnataka 4. Chennai
Please enter your option: 2
```

6. Choosing correct answer for a question when the atom-man collects three books



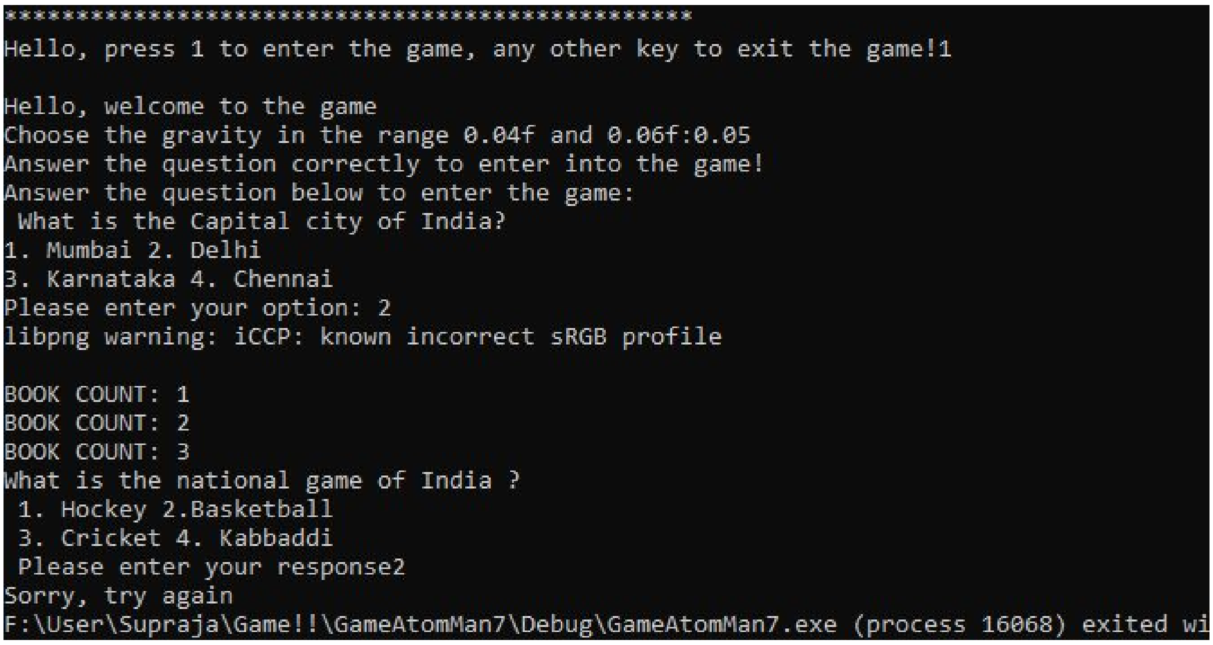
```
************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.045
Answer the question correctly to enter into the game!
Answer the question below to enter the game:
 What is the Capital city of India?
1. Mumbai 2. Delhi
3. Karnataka 4. Chennai
Please enter your option: 2
libpng warning: iCCP: known incorrect sRGB profile

2 MORE LIVES LEFT
BOOK COUNT: 1
BOOK COUNT: 2
BOOK COUNT: 3
What is the national game of India ?
 1. Hockey 2.Basketball
 3. Cricket 4. Kabbaddi
 Please enter your response1

1 MORE LIVES LEFT
```

7. Choosing the incorrect option for the question for the first question

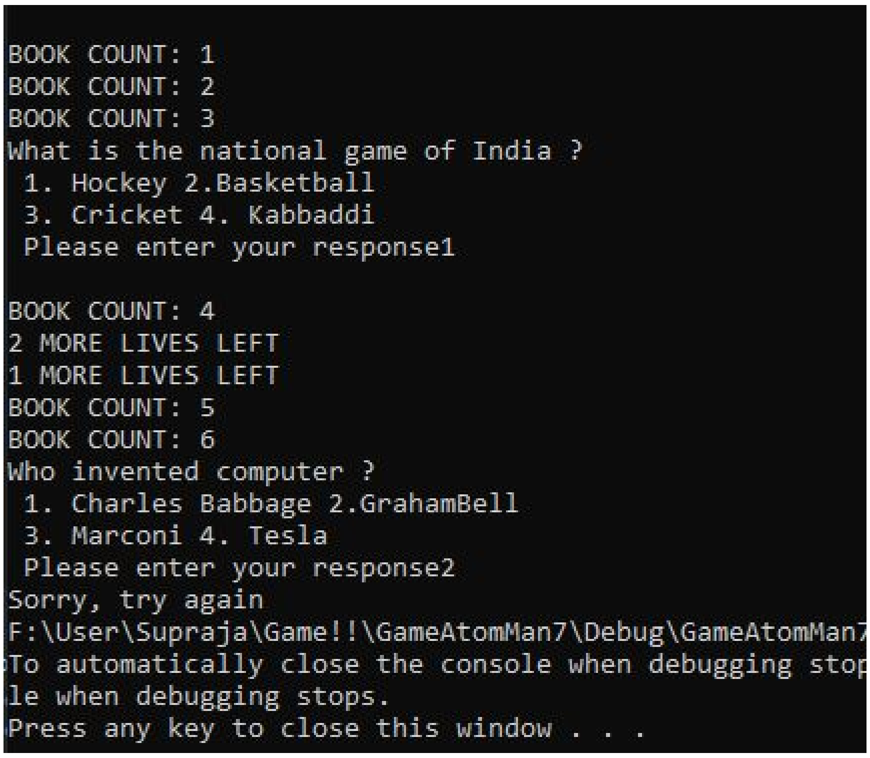


```
************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.05
Answer the question correctly to enter into the game!
Answer the question below to enter the game:
 What is the Capital city of India?
1. Mumbai 2. Delhi
3. Karnataka 4. Chennai
Please enter your option: 2
libpng warning: iCCP: known incorrect sRGB profile

BOOK COUNT: 1
BOOK COUNT: 2
BOOK COUNT: 3
What is the national game of India ?
 1. Hockey 2.Basketball
 3. Cricket 4. Kabbaddi
 Please enter your response2
Sorry, try again
F:\User\Supraja\Game!!\GameAtomMan7\Debug\GameAtomMan7.exe (process 16068) exited wi
```

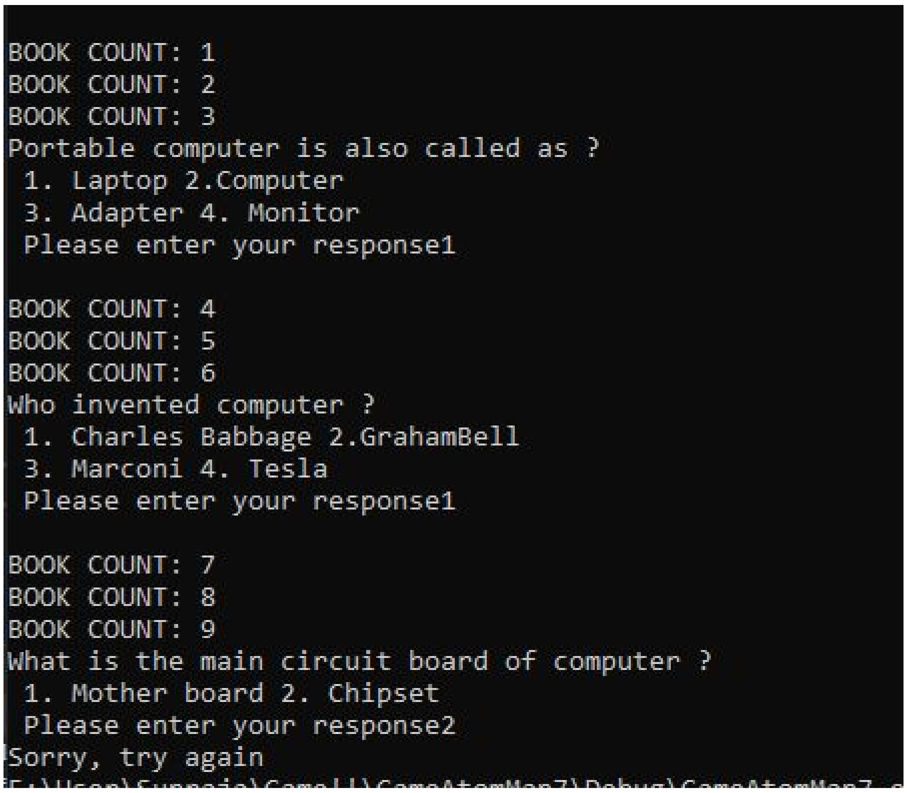
8.Demonstrating that user has lost the game if he/she answers the second quiz question incorrectly.



```
BOOK COUNT: 1
BOOK COUNT: 2
BOOK COUNT: 3
What is the national game of India ?
 1. Hockey 2.Basketball
 3. Cricket 4. Kabbaddi
 Please enter your response1

BOOK COUNT: 4
2 MORE LIVES LEFT
1 MORE LIVES LEFT
BOOK COUNT: 5
BOOK COUNT: 6
Who invented computer ?
 1. Charles Babbage 2.GrahamBell
 3. Marconi 4. Tesla
 Please enter your response2
Sorry, try again
F:\User\Supraja\Game!!\GameAtomMan7\Debug\GameAtomMan7
To automatically close the console when debugging stop
le when debugging stops.
Press any key to close this window . . .
```

9 .Demonstrating that user has lost the game if he/she answers the third quiz question

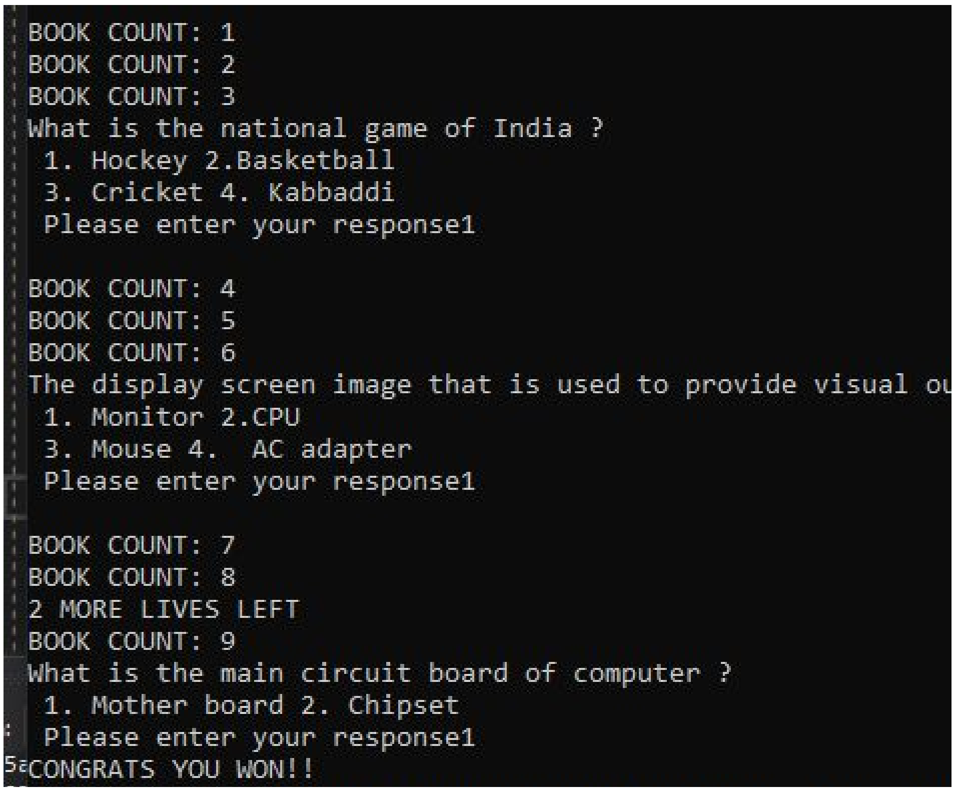


```
BOOK COUNT: 1
BOOK COUNT: 2
BOOK COUNT: 3
Portable computer is also called as ?
 1. Laptop 2.Computer
 3. Adapter 4. Monitor
 Please enter your response1

BOOK COUNT: 4
BOOK COUNT: 5
BOOK COUNT: 6
Who invented computer ?
 1. Charles Babbage 2.GrahamBell
 3. Marconi 4. Tesla
 Please enter your response1

BOOK COUNT: 7
BOOK COUNT: 8
BOOK COUNT: 9
What is the main circuit board of computer ?
 1. Mother board 2. Chipset
 Please enter your response2
Sorry, try again
```

10.Demonstrating that user can win the game by answering all quiz questions correctly.

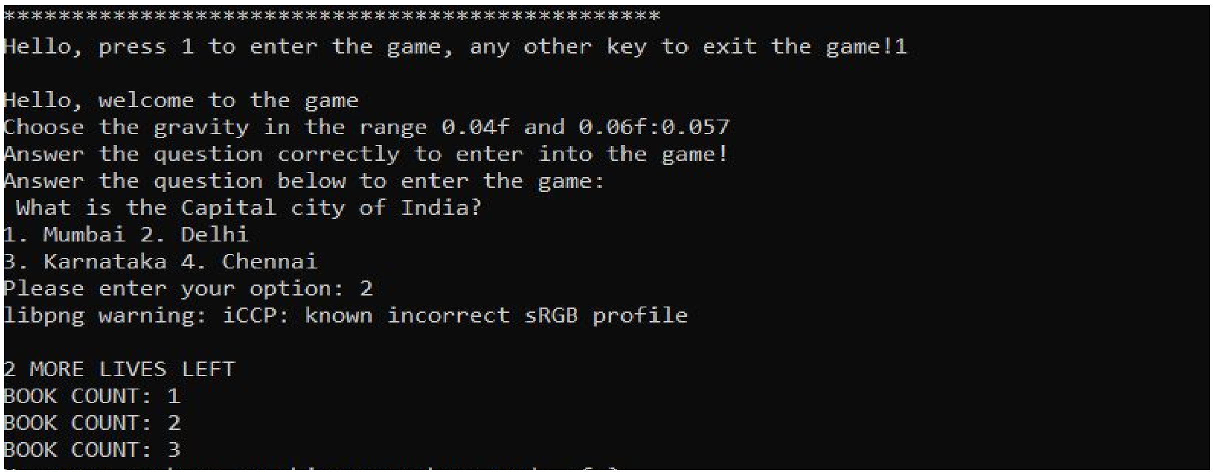


```
BOOK COUNT: 1
BOOK COUNT: 2
BOOK COUNT: 3
What is the national game of India ?
 1. Hockey 2.Basketball
 3. Cricket 4. Kabbaddi
 Please enter your response1

BOOK COUNT: 4
BOOK COUNT: 5
BOOK COUNT: 6
The display screen image that is used to provide visual ou
 1. Monitor 2.CPU
 3. Mouse 4.  AC adapter
 Please enter your response1

BOOK COUNT: 7
BOOK COUNT: 8
2 MORE LIVES LEFT
BOOK COUNT: 9
What is the main circuit board of computer ?
 1. Mother board 2. Chipset
 Please enter your response1
CONGRATS YOU WON!!
```

11. Demonstrating the number of books collected by the user

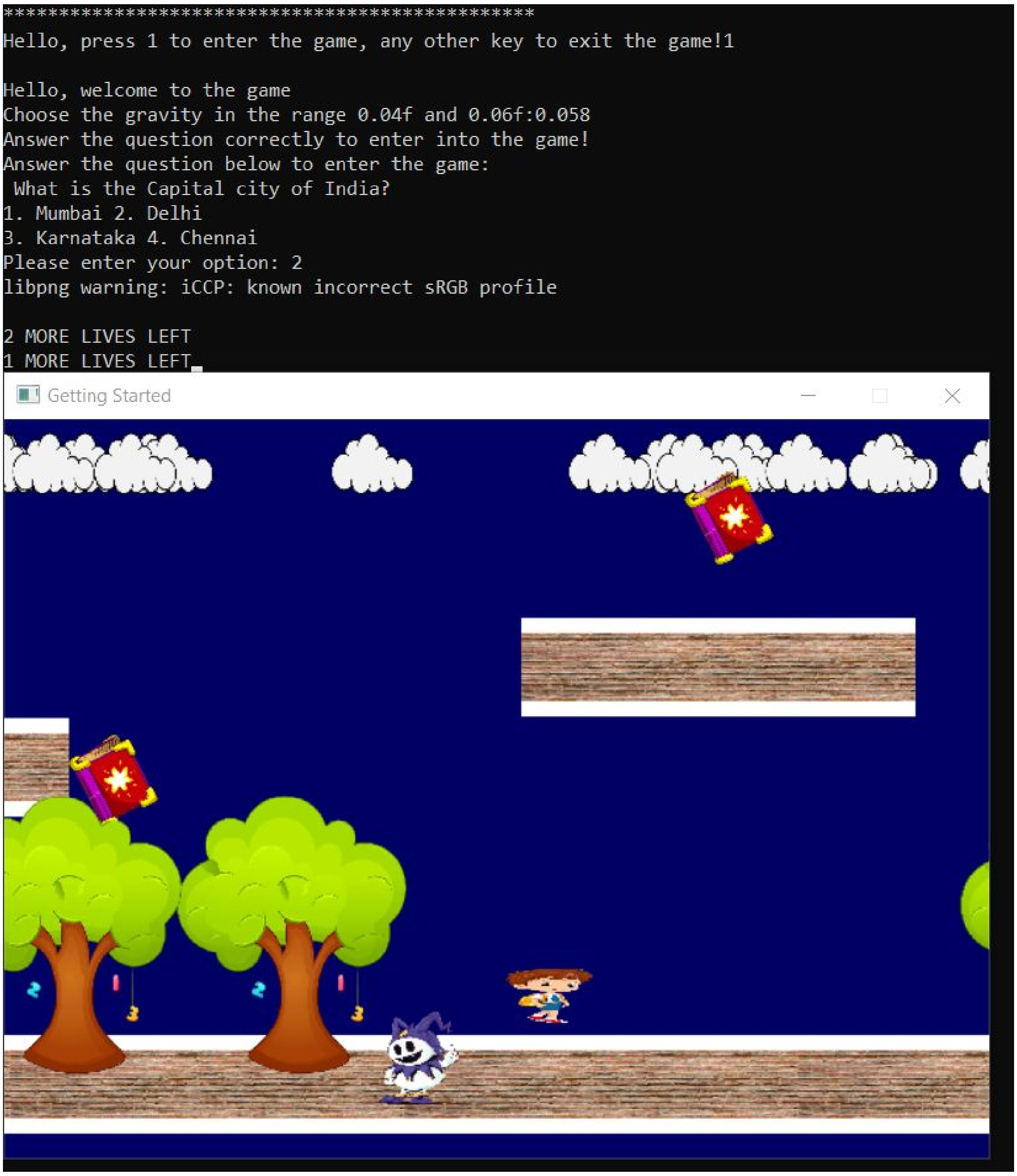


```
************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.057
Answer the question correctly to enter into the game!
Answer the question below to enter the game:
 What is the Capital city of India?
1. Mumbai 2. Delhi
3. Karnataka 4. Chennai
Please enter your option: 2
libpng warning: iCCP: known incorrect sRGB profile

2 MORE LIVES LEFT
BOOK COUNT: 1
BOOK COUNT: 2
BOOK COUNT: 3
```

12. Demonstrating the number of lives left for the user

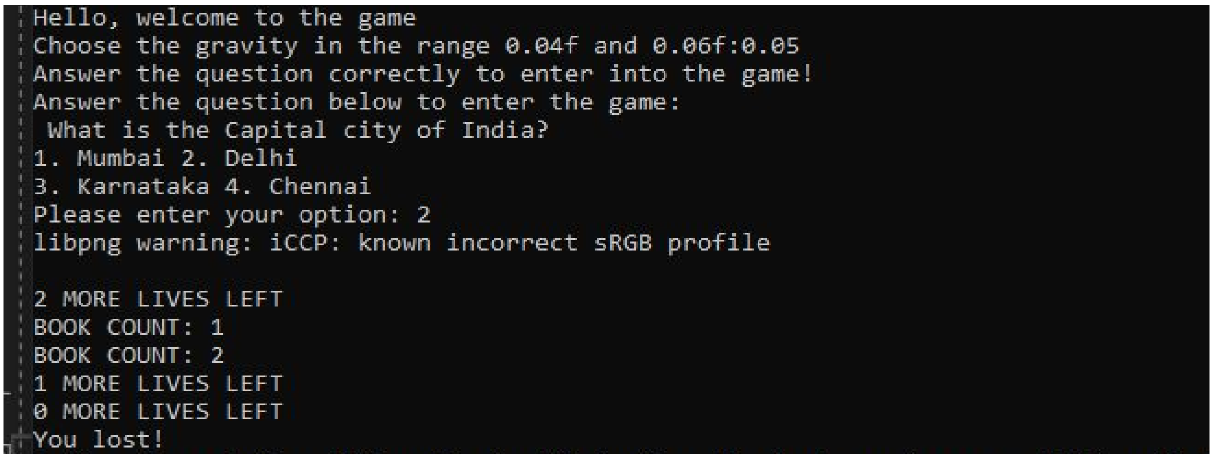


```
************************************************
Hello, press 1 to enter the game, any other key to exit the game!1

Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.058
Answer the question correctly to enter into the game!
Answer the question below to enter the game:
 What is the Capital city of India?
1. Mumbai 2. Delhi
3. Karnataka 4. Chennai
Please enter your option: 2
libpng warning: iCCP: known incorrect sRGB profile

2 MORE LIVES LEFT
1 MORE LIVES LEFT_
```

13. Demonstrating that user has lost the game when he loses all his three lives.



```
Hello, welcome to the game
Choose the gravity in the range 0.04f and 0.06f:0.05
Answer the question correctly to enter into the game!
Answer the question below to enter the game:
 What is the Capital city of India?
1. Mumbai 2. Delhi
3. Karnataka 4. Chennai
Please enter your option: 2
libpng warning: iCCP: known incorrect sRGB profile

2 MORE LIVES LEFT
BOOK COUNT: 1
BOOK COUNT: 2
1 MORE LIVES LEFT
0 MORE LIVES LEFT
You lost!
```

ADDITIONAL LEARNINGS

Besides the C language and its default libraries, we have learnt about several libraries like SDL.h, SDL_image, SDL_ttf.h.

SDL.h

SDL is Simple DirectMedia Layer.It is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D.It can be used to make animations and video games.
It basically provides set of APIs to interact with various devices like graphics hardware, mouse, audio, keyboard etc
It is written in C programming language and works for C++, c#, python as well.
SDL uses OpenGL as a hardware renderer for content that wants hardware rendering on some platforms. If you have such platforms, then OpenGL is the underlying API over which SDL is an abstraction.

SDL_image

SDL_image is an image file loading library.
It loads images as SDL surfaces and textures, and supports the following formats: BMP, GIF, JPEG, LBM, PCX, PNG, PNM, SVG, TGA, TIFF, WEBP, XCF, XPM, XV.
As of SDL_image 1.2.5, JPEG, PNG, TIFF, and WEBP image loading libraries are dynamically loaded, so if you don't need to load those formats, you don't need to include those shared libraries. libpng depends on libz, and libtiff depends on both libz and libjpeg.

SDL_ttf.h

This is a sample library which allows you to use TrueType fonts in your SDL applications. It comes with an example program "showfont" which displays an example string for a given TrueType font file.
For this library to work in the application, we have to download any available .ttf file available on the internet and can use the font to display the text on the window.

ERROR HANDLING IN VISUAL STUDIO CODE 2019

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it.
Software genre: Source-code editor
Developer: Microsoft Corporation

Visual Studio does not support commands like scanf. Instead we need to mention the buffer size as well like: scanf_s, scanf_s_l, wscanf_ s, _wscanf_s.

Similarly, sprintf() does not work as well. More secure versions of it have to be used like sprintf_s, _sprintf_s_l, swprintf_s, _swprintf_s_l.2

DISCUSSION AND FUTURE-WORK

This application can be further improvised by implementing GUI as a part of this project. GUI can be implemented by using gtk library. The quiz questions or choosing the value of the gravity by using GUI and user will be given a choice to select a value by clicking on the buttons

This can be even more modified by placing a leaderboard and by introducing background music while the user makes the atom-man jump and when the atom collides with a demon.

And we also use a display box in the window itself for assuring the user that how many more lives are left.

The project can be furthermore made easy if it were implemented in any of the high-level languages like Java or Python, as GUI and external libraries would be quite easier to handle in them.

Regarding the logic of the application, further modifications can be made like, the user can be given a choice to select among the demons whom he wants to fight with, the user can also be given a choice to select the level of difficulty of the questions. The demons which can move can also be implemented in the application. It makes the game more complicated than it already is. Quiz can also be modified by providing hints for the questions, if the user is unable to answer them.

REFERENCES

https://www.youtube.com/watch?v=tmGBhM8AEj8&t=470s
https://www.youtube.com/watch?v=AC0UFee-oVo
https://stackoverflow.com/questions/63315447/why-is-scanf-s-not-wo
rking-properly-in-visual-studio
https://www.cprogramming.com/tutorial/sdl/setup.html
https://stackoverflow.com/questions/31348266/sprint-navigation-does
nt-show-in-visual-studio-team-services
https://visualstudio.microsoft.com/vs/
https://visualstudio.microsoft.com/
https://wiki.libsdl.org/Tutorials
https://all-free-download.com/font/
https://www.fontspace.com/category/ttf
https://www.libsdl.org/projects/SDL_image/docs/index.html