HO CHI MINH CITY NATIONAL UNIVERSITY
UNIVERSITY OF SCIENE

Project report

Subject: Object oriented programming

SPACE WAR

PART CODE
CSC10003

Ho Chi Minh city - 2020

TABLE OF CONTENTS

1. OVER VIEW

a. MEMBER

STUDENT ID	FULL NAME	EMAIL
19127525	Nguyễn Thanh Quân	19127525@student.hcmus.edu.vn
19127422	Nguyễn Đức Huy	19127422@student.hcmus.edu.vn

b. INTRODUCE PROJECT

Space war is the game take idea from game Space Shooter, it has 2 round and Player will control the space to kill all enemy to save the world from the apocalypse

We complete all task in proposal

After complete all task in proposal, we add function count time play game, draw galaxy background, update skin of space (buy in shop), update bullet (buy in shop)

Self-research function: multi thread

Function want to bonus score: multi thread

c. JOBS

STUDENT ID	FULL NAME	JOBS
19127525	Nguyễn Thanh Quân	Multi thread Play game Movement of enemy Save game Load game Count score Shop Check name save Update skin of space Update bullet
19127422	Nguyễn Đức Huy	Draw enemy Erase enemy Draw frame Movement of bullet Graphic enemy be shoted Graphic space impact enemy Graphic next round Load game bar Show score Menus Reset data Shot bullet

		Update graphic Graphic boss die Set enemy coordinate Sound effect Cout time play game

d. EVALUATE MEMBER

STUDENT ID	FULL NAME	EVALUATED
19127525	Nguyễn Thanh Quân	100%
19127422	Nguyễn Đức Huy	100%

2. GAME CONTENT

a. FUNCTIONS CONTENT

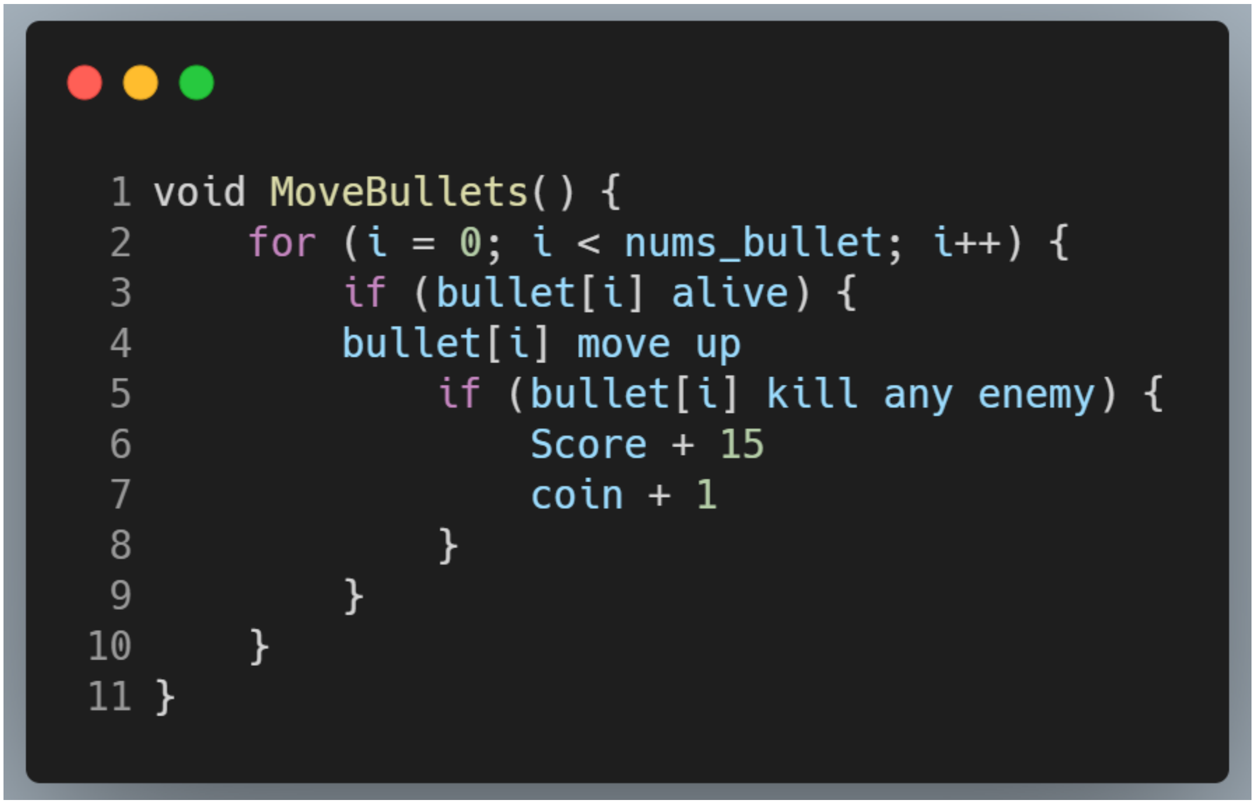
File name	Meaning of functions in files
Header.h	Declare all Class, function and global variable of program.
Coordinate.cpp	**Implement all method of class Coordinate:** Draw(char c): Draw char c at coordinate(x,y) of instance getX(): get x coordinate of instance getY():get y coordinate of instance setCoord(int x,int y): change the coordinate of instance into (int x,int y)
SpaceShip.cpp	**Implement all method of class SpaceShip:** SpaceShip(): Initialize and set the default coordinate of space Draww(): Draww space Erase(): Erase the space Left(): move coordinate of space to the left Right(): move coordinate of space to the right Up(): move coordinate of space to the up Down(): move coordinate of space to the down **Implement function of space:** Shot(int x,int y): let the space shot bullet
Bullet.cpp	**Implement all method of class Bullet:** Bullet(): Initialize and set the default coordinate of bullet instance **Implement function of bullet:** DrawBullet(): Draw all object of Class Bullet
Enemy.cpp	**Implement all method of class Enemy:**

	Enemy(): Initialize and set the default coordinate of enemy instance TouchBoundary(): check whether any enemy touch the boundary to change the movement of enemy **Implement function of enemy:** SetCoordEnemy_R1(Enemy e[]): set up coordinate of all enemy in round 1 SetCoordEnemy_R2(Enemy e[]):set up coordinate of all enemy in round 2 to build up the Boss DrawEnemy(): Draw all enemy alive EraseEnemy(): Erase all enemy alive
Star.cpp	**Implement all method of class Enemy:** Star(): Initialize and set up default coordinate of star instance **Implement function of star:** SetCoordStar(Star star[STAR]): set up coordinate of all star DrawStar(): Draw all star (galaxy backgound) EraseStar(): Erase all star
Graphic.cpp	FixConsoleWindow(): Fixed the console window Announce(): show information of game (time playing, coin, score, life, bullet left and enemy left) GraphicNextRound(): Graphic when you pass round 1 and enter round 2 BossHP(): Draw HP of Boss in round 2 LoadGame(): Draw loading game bar MenuDie(): Show notice and option to play again when you impact or out of bullet MenuWin(): Let you save game when you win all round Menu(): Graphic when you start the game Menu1(): Graphic when you choose “Start game” in Menu() MenuShop(): Show the shop and Item which you can buy MenuScore(): Show all information had saved before (name, total score, time had played) MenuRule(): Show the detail rule of game MenuAbout(): Show the detail information of project draw_bye1(): Draw Bye draw_bye2(): Draw Bye ByeBye(): Graphic when you exit application GraphicBossDie(): Graphic when boss die

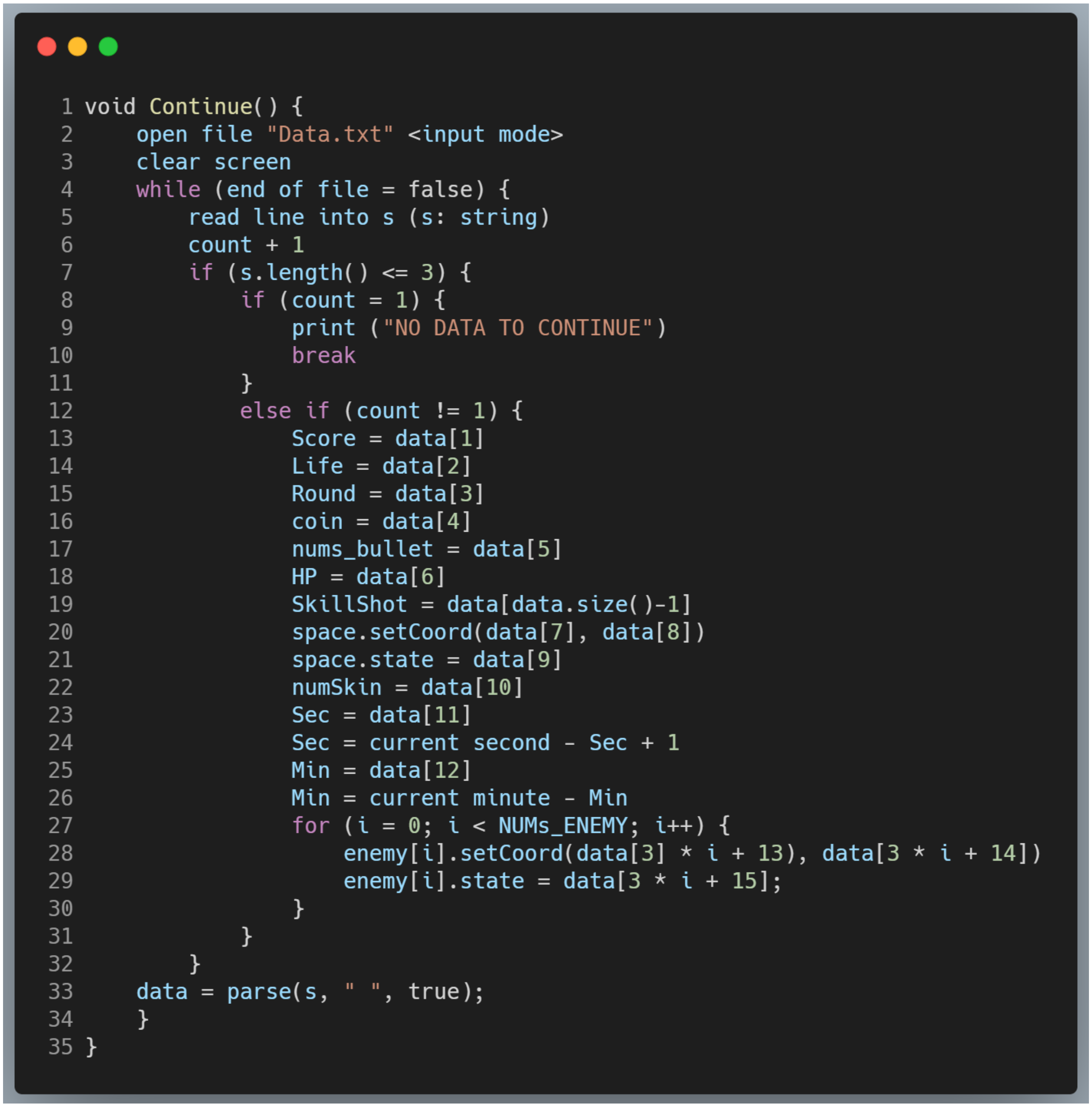
	GraphicEnemyDie(int x,int y): Graphic when you kill enemy GraphicImpact(): Graphic when you impact the enemy DrawFrame(): Draw the frame of game void GraphicEarthSafe(): Graphice earth
Function.cpp	vector<string> parse(string haystack, string seperator = " ", bool RemoveEmptyEntries = false): Split string into vector string ResetData(): Set up default data StartGame(): Repare all fucntion before play GotoXY(): move the point to coordinate (x,y) ExitGame(): Exit application SetColor(): Change the text color TextColor(): Change the text and backgound of text color Kill(): Check whether bullet kill enemy CheckDie(): Check whether the space impact or out of bullet isImpact(): Check whether the space impact enemy PauseGame(HANDLE t): Pause game Pause(HANDLE t): Pause thread PlayGame(): Enter the game and start play SubThread(): Run the thread parrallet game checkRound1(): Check whether you pass the round 1 (kill all enemy of round 1) CheckName(): Check whether the name you save coincide with all name had saved before TotalScore(): Caculate the total score SaveGame(): Save game Continue(): Continue the last game had saved before
Sound.cpp	SoundGame(): Sound while you playing game SoundBossDie(): Sound when boss die SoundLoading(): Sound loading data SoundExplosion():Sound explosion space SoundBackground(): Sound while you in menu SoundGameOver(): Sound when you lose SoundGameWin(): Sound when you win SoundCheering(): Sound cheering SoundByeBye(): Sound say bye bye SoundRocket(): Sound of rocket EndShowGame(): Stop sound game
Move.cpp	**Implement the function of movement of enemy, bullet, star:**

	MoveRight(Enemy enemy[NUMs_ENEMY]): Set all enemy move right MoveLeft(Enemy enemy[NUMs_ENEMY]) : Set all enemy move left MoveDown(Enemy enemy[NUMs_ENEMY]) : Set all enemy move down MoveUp(Enemy enemy[NUMs_ENEMY]) : Set all enemy move up MoveEnemy(): Set up the movement of all enemy (move right, left, up, down) MoveBullets(): Set up the movement of bullet MoveStar(): Set up the movement of star
Time.cpp	modernization(int sec, int min): return time is playing (second) printTime(): print the time is playing startTime(): take the default second and minute when you play game
Main.cpp	Initialize value of global variable: number of life, index of skin, fixed console window

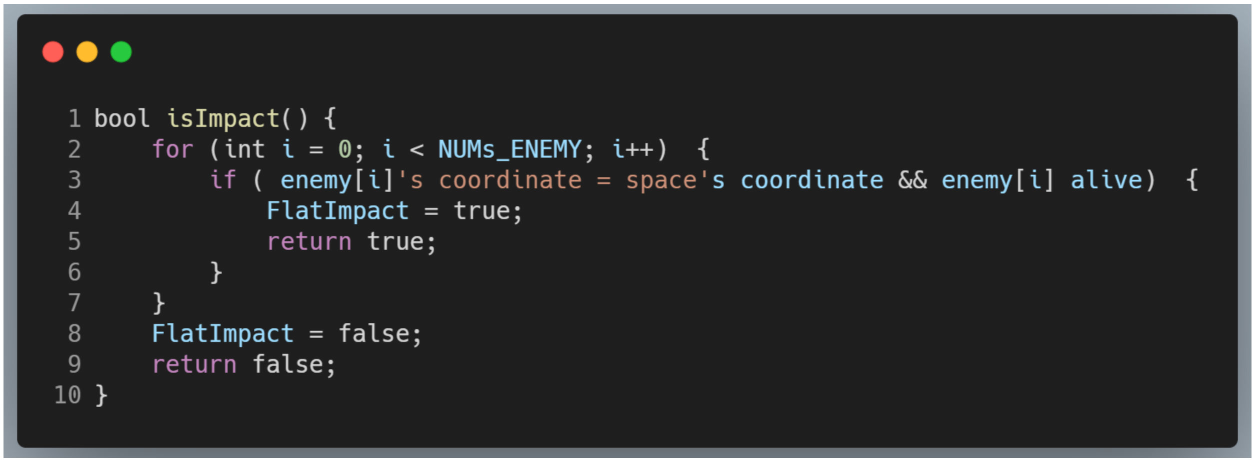
b. MAIN FUNCTION (explain by pseudocode)



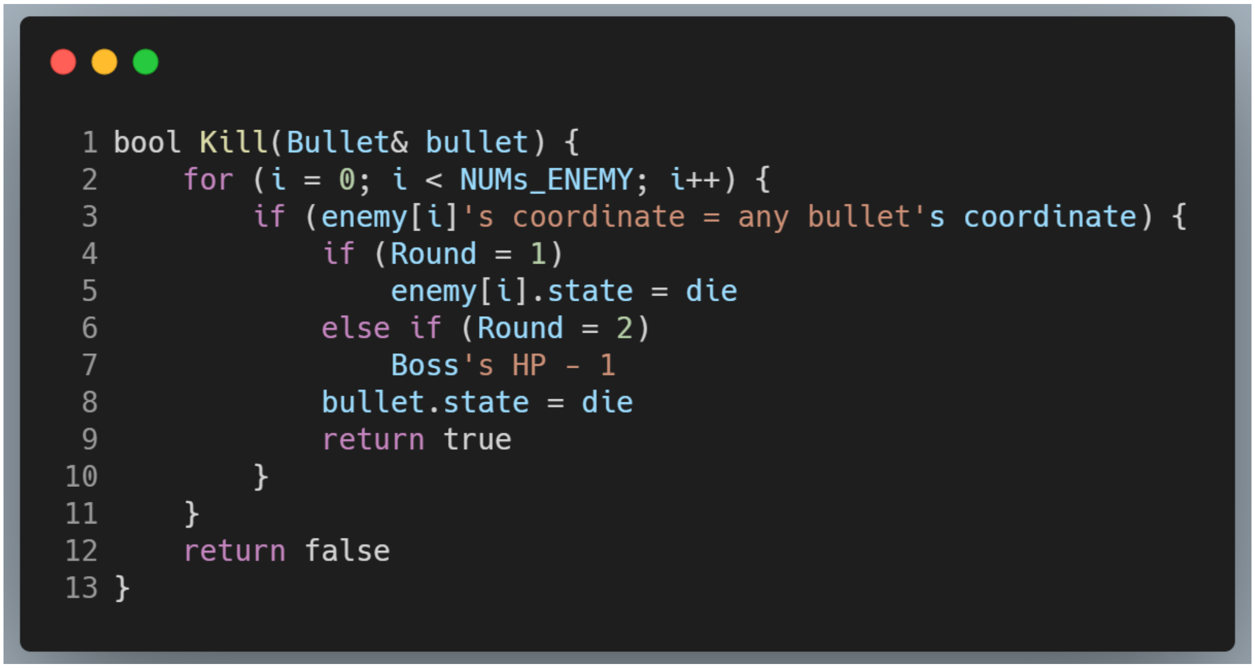
```
void MoveBullets() {
    for (i = 0; i < nums_bullet; i++) {
        if (bullet[i] alive) {
        bullet[i] move up
            if (bullet[i] kill any enemy) {
                Score + 15
                coin + 1
            }
        }
    }
}
```



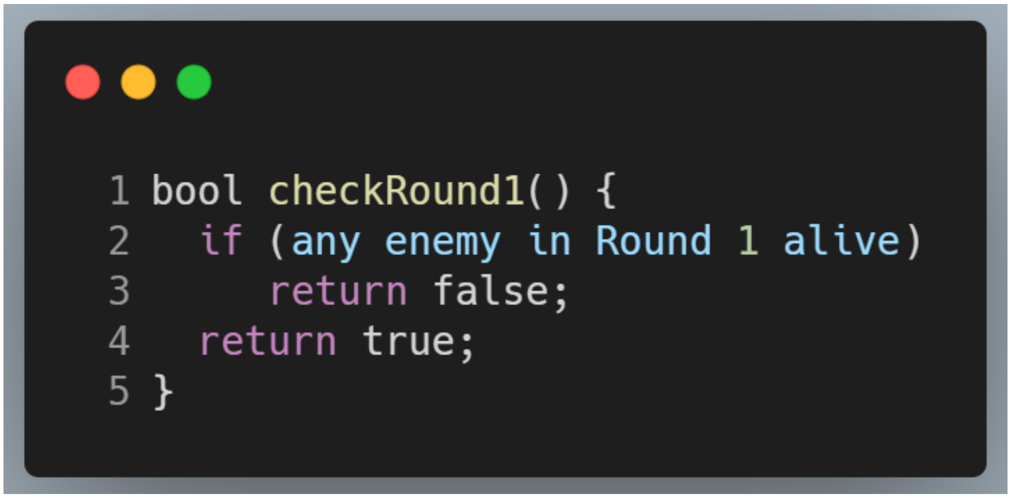
```
void Continue() {
    open file "Data.txt" <input mode>
    clear screen
    while (end of file = false) {
        read line into s (s: string)
        count + 1
        if (s.length() <= 3) {
            if (count = 1) {
                print ("NO DATA TO CONTINUE")
                break
            }
            else if (count != 1) {
                Score = data[1]
                Life = data[2]
                Round = data[3]
                coin = data[4]
                nums_bullet = data[5]
                HP = data[6]
                SkillShot = data[data.size()-1]
                space.setCoord(data[7], data[8])
                space.state = data[9]
                numSkin = data[10]
                Sec = data[11]
                Sec = current second - Sec + 1
                Min = data[12]
                Min = current minute - Min
                for (i = 0; i < NUMs_ENEMY; i++) {
                    enemy[i].setCoord(data[3] * i + 13), data[3 * i + 14])
                    enemy[i].state = data[3 * i + 15];
                }
            }
        }
    data = parse(s, " ", true);
    }
}
```



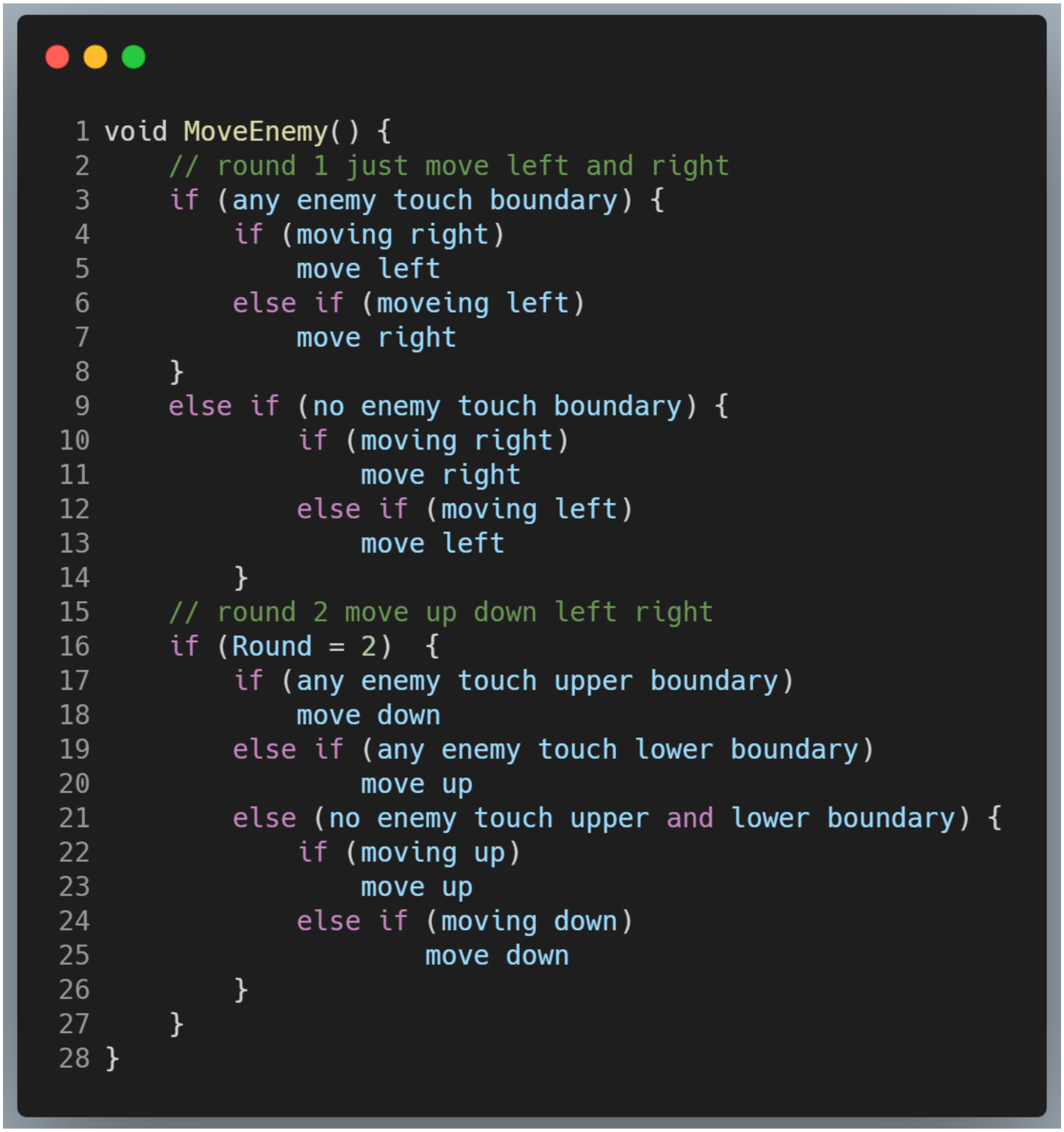
```
bool isImpact() {
    for (int i = 0; i < NUMs_ENEMY; i++)  {
        if ( enemy[i]'s coordinate = space's coordinate && enemy[i] alive)  {
            FlatImpact = true;
            return true;
        }
    }
    FlatImpact = false;
    return false;
}
```



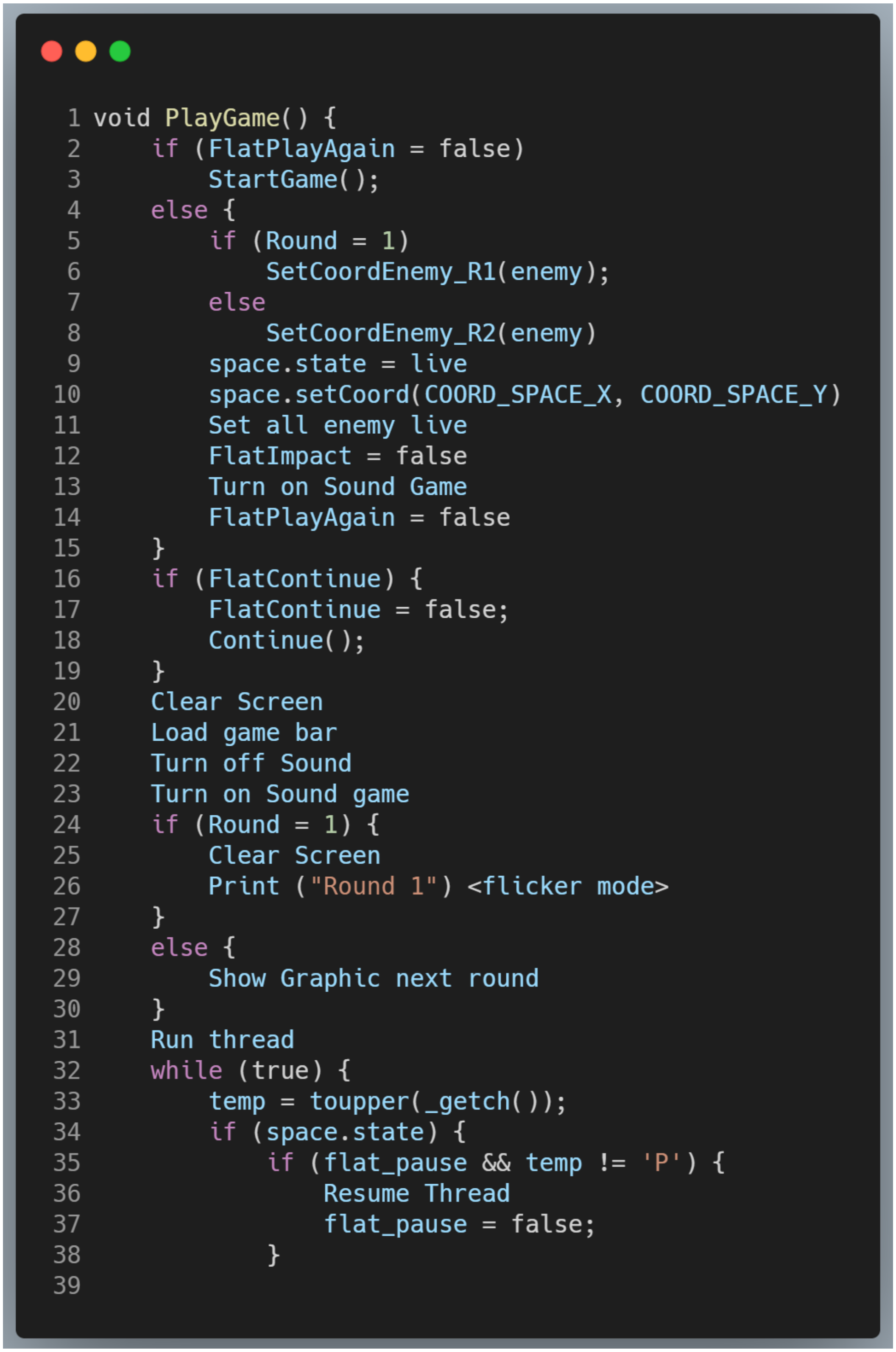
```
bool Kill(Bullet& bullet) {
    for (i = 0; i < NUMs_ENEMY; i++) {
        if (enemy[i]'s coordinate = any bullet's coordinate) {
            if (Round = 1)
                enemy[i].state = die
            else if (Round = 2)
                Boss's HP - 1
            bullet.state = die
            return true
        }
    }
    return false
}
```



```
bool checkRound1() {
  if (any enemy in Round 1 alive)
      return false;
  return true;
}
```

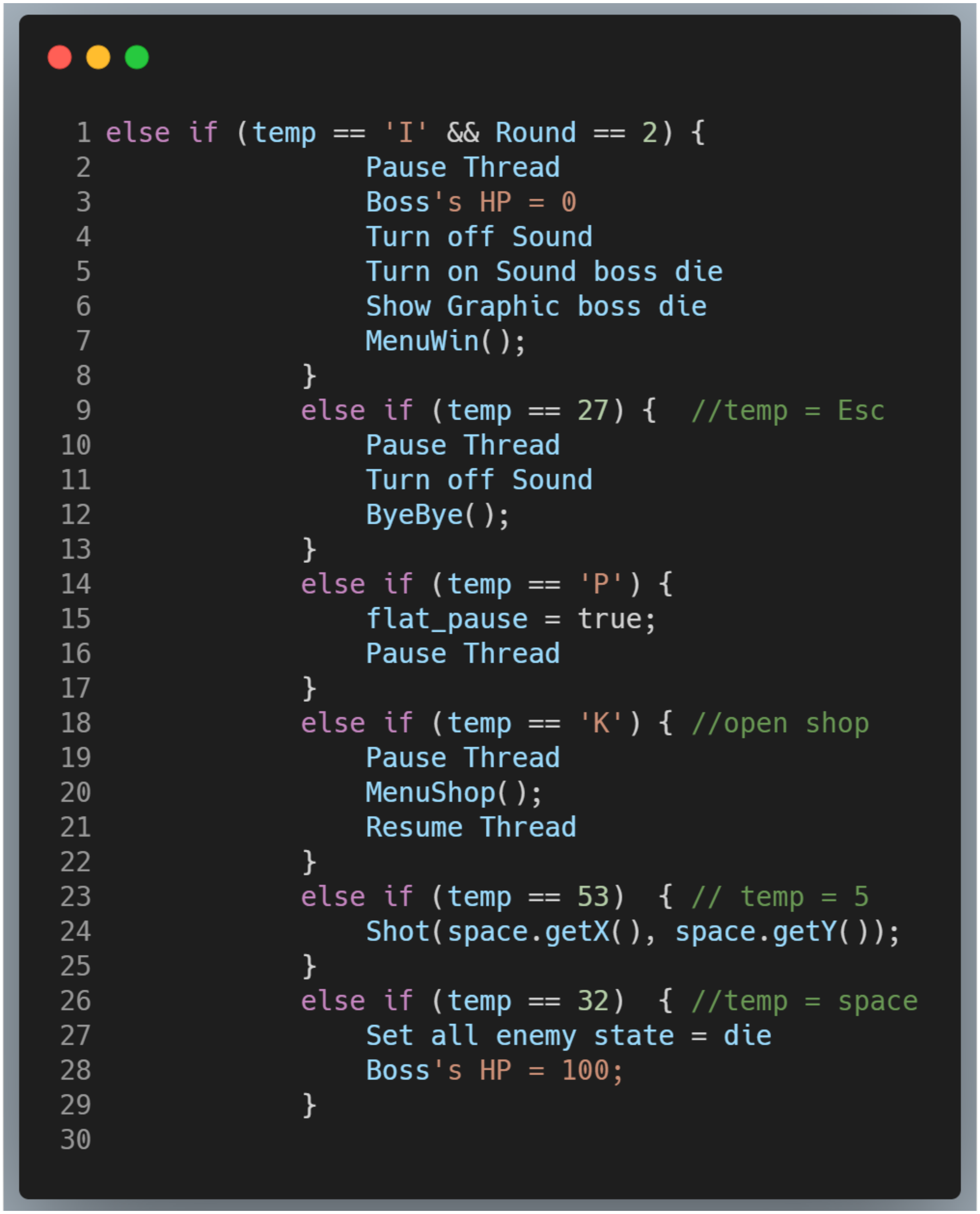


```
void MoveEnemy() {
    // round 1 just move left and right
    if (any enemy touch boundary) {
        if (moving right)
            move left
        else if (moveing left)
            move right
    }
    else if (no enemy touch boundary) {
            if (moving right)
                move right
            else if (moving left)
                move left
        }
    // round 2 move up down left right
    if (Round = 2)  {
        if (any enemy touch upper boundary)
            move down
        else if (any enemy touch lower boundary)
                move up
        else (no enemy touch upper and lower boundary) {
            if (moving up)
                move up
            else if (moving down)
                    move down
        }
    }
}
```



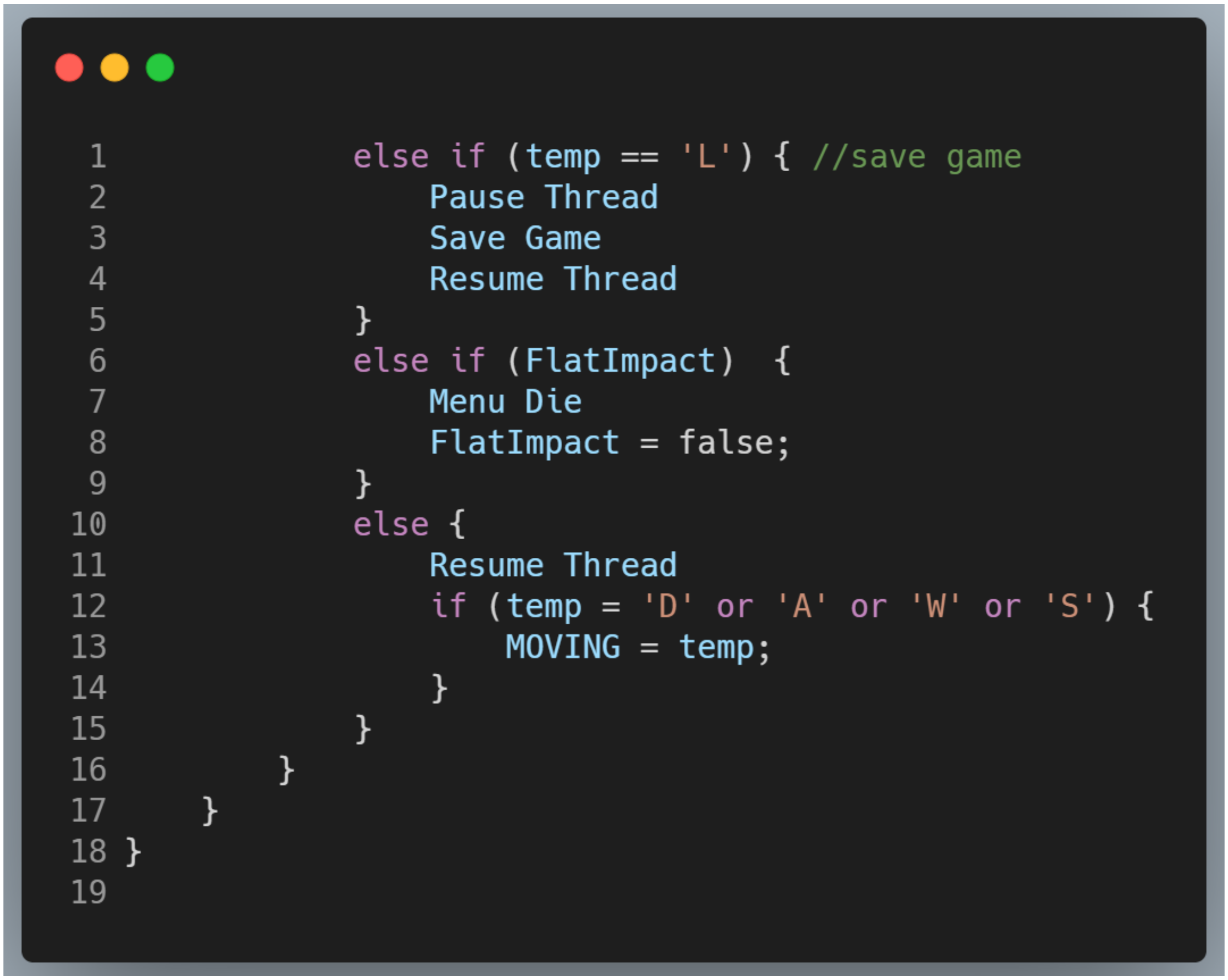
```
void PlayGame() {
    if (FlatPlayAgain = false)
        StartGame();
    else {
        if (Round = 1)
            SetCoordEnemy_R1(enemy);
        else
            SetCoordEnemy_R2(enemy)
        space.state = live
        space.setCoord(COORD_SPACE_X, COORD_SPACE_Y)
        Set all enemy live
        FlatImpact = false
        Turn on Sound Game
        FlatPlayAgain = false
    }
    if (FlatContinue) {
        FlatContinue = false;
        Continue();
    }
    Clear Screen
    Load game bar
    Turn off Sound
    Turn on Sound game
    if (Round = 1) {
        Clear Screen
        Print ("Round 1") <flicker mode>
    }
    else {
        Show Graphic next round
    }
    Run thread
    while (true) {
        temp = toupper(_getch());
        if (space.state) {
            if (flat_pause && temp != 'P') {
                Resume Thread
                flat_pause = false;
            }

```



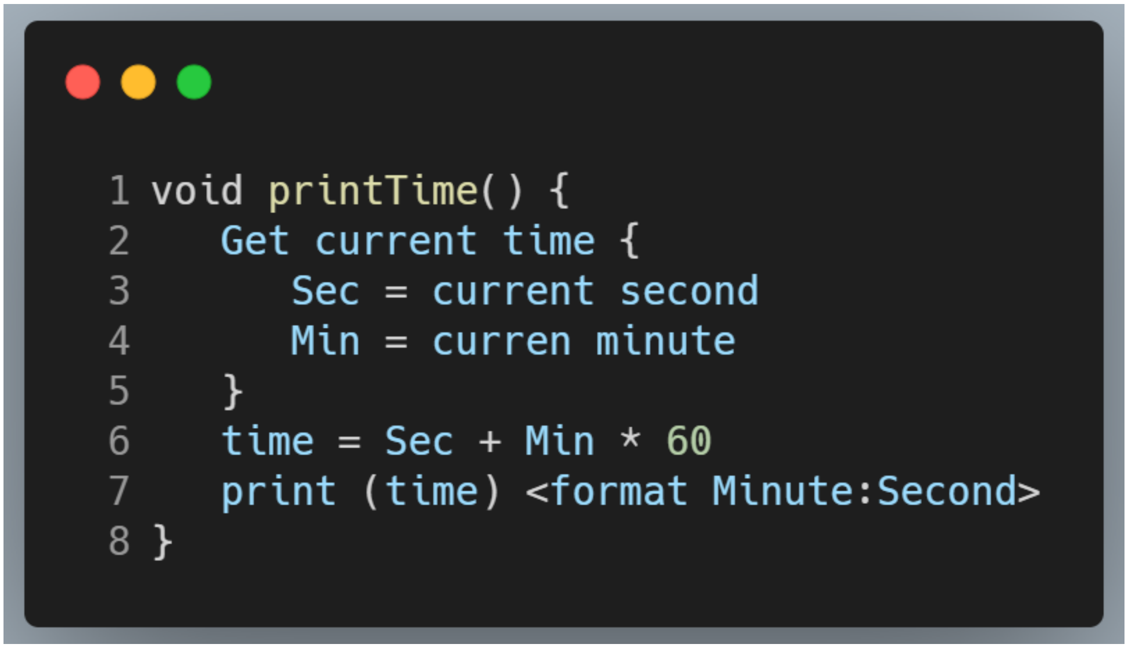
```
else if (temp == 'I' && Round == 2) {
                Pause Thread
                Boss's HP = 0
                Turn off Sound
                Turn on Sound boss die
                Show Graphic boss die
                MenuWin();
            }
            else if (temp == 27) {  //temp = Esc
                Pause Thread
                Turn off Sound
                ByeBye();
            }
            else if (temp == 'P') {
                flat_pause = true;
                Pause Thread
            }
            else if (temp == 'K') { //open shop
                Pause Thread
                MenuShop();
                Resume Thread
            }
            else if (temp == 53)  { // temp = 5
                Shot(space.getX(), space.getY());
            }
            else if (temp == 32)  { //temp = space
                Set all enemy state = die
                Boss's HP = 100;
            }

```

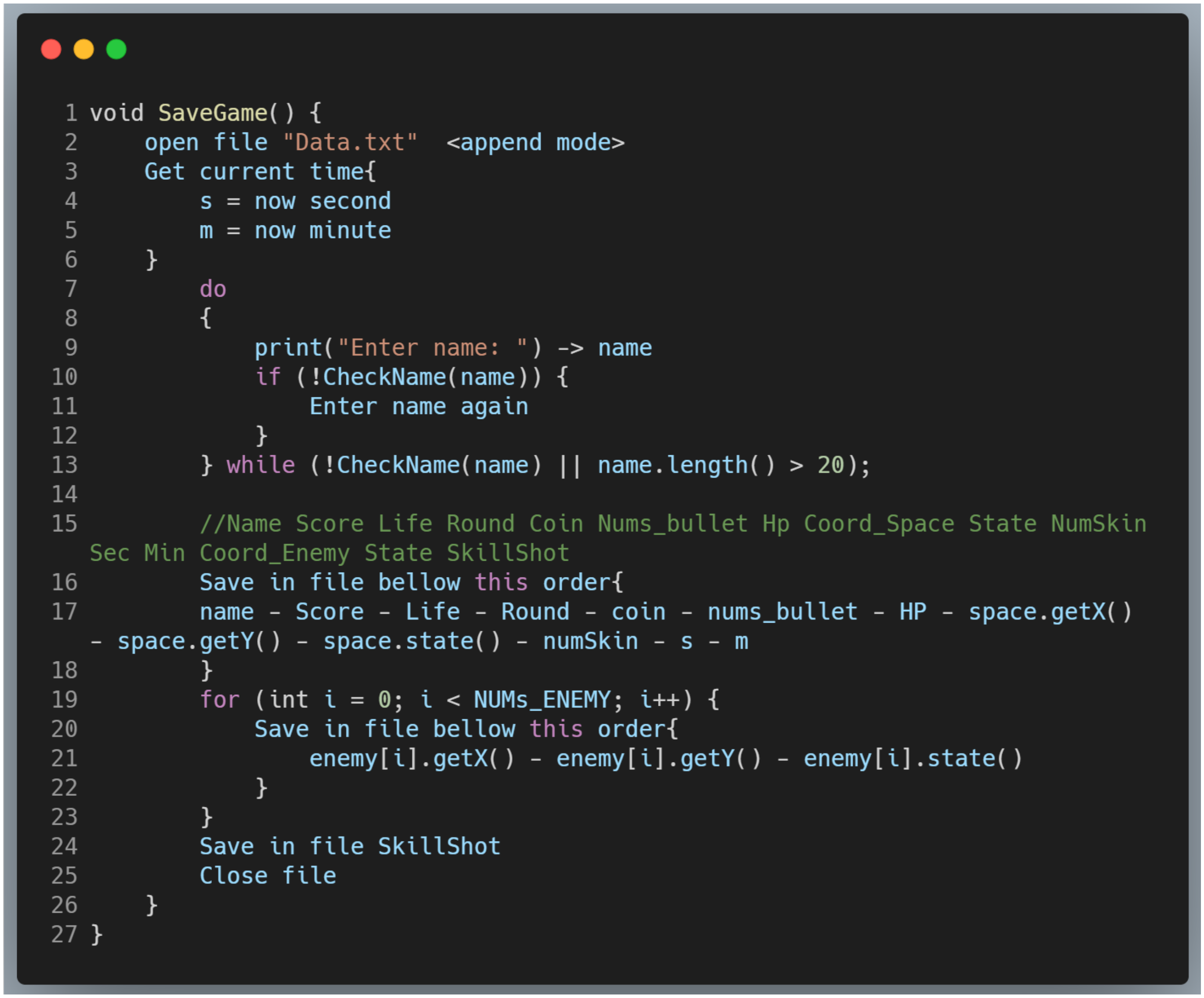


```
            else if (temp == 'L') { //save game
                Pause Thread
                Save Game
                Resume Thread
            }
            else if (FlatImpact)  {
                Menu Die
                FlatImpact = false;
            }
            else {
                Resume Thread
                if (temp = 'D' or 'A' or 'W' or 'S') {
                    MOVING = temp;
                }
            }
        }
    }
}

```

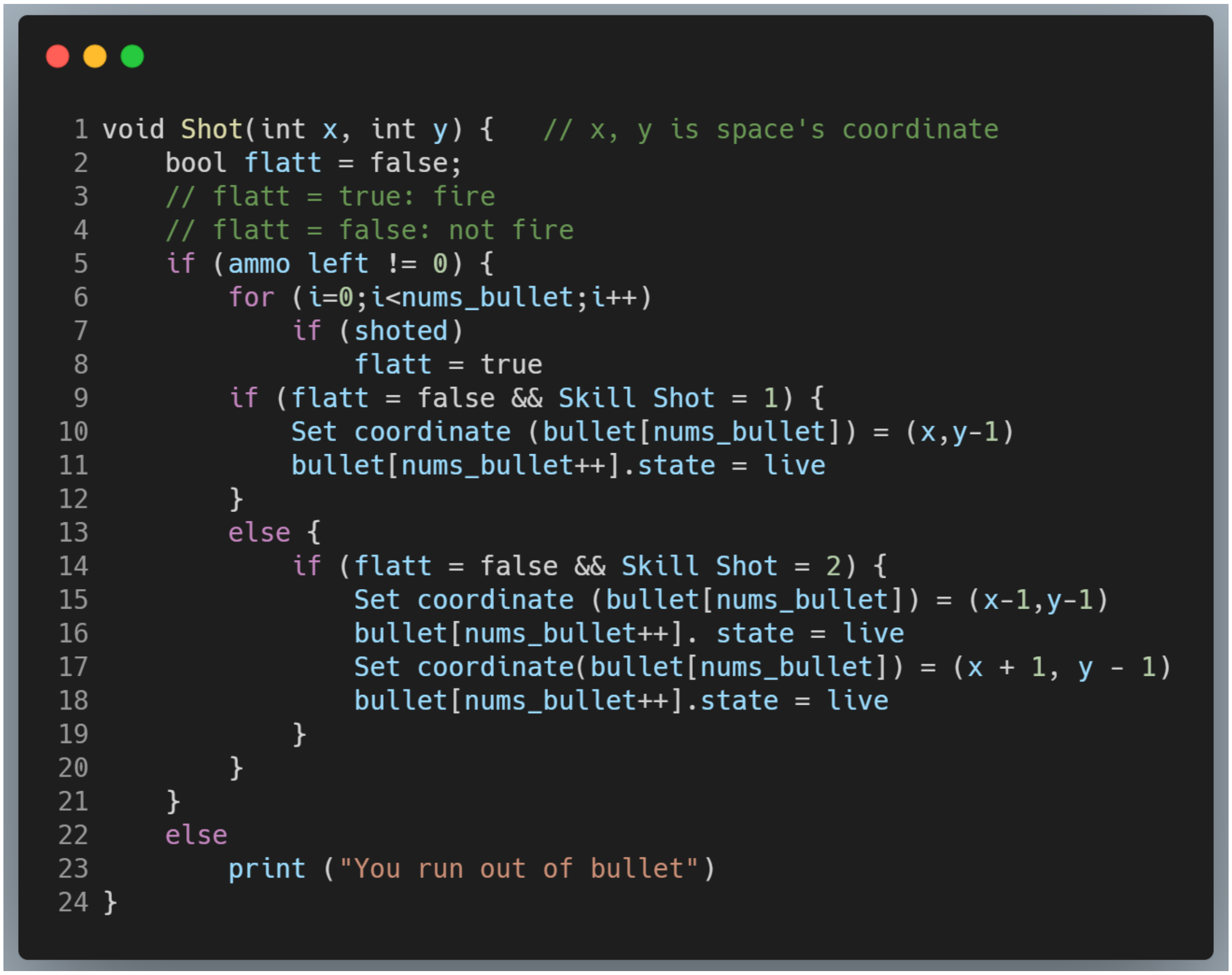


```
void printTime() {
   Get current time {
      Sec = current second
      Min = curren minute
   }
   time = Sec + Min * 60
   print (time) <format Minute:Second>
}
```

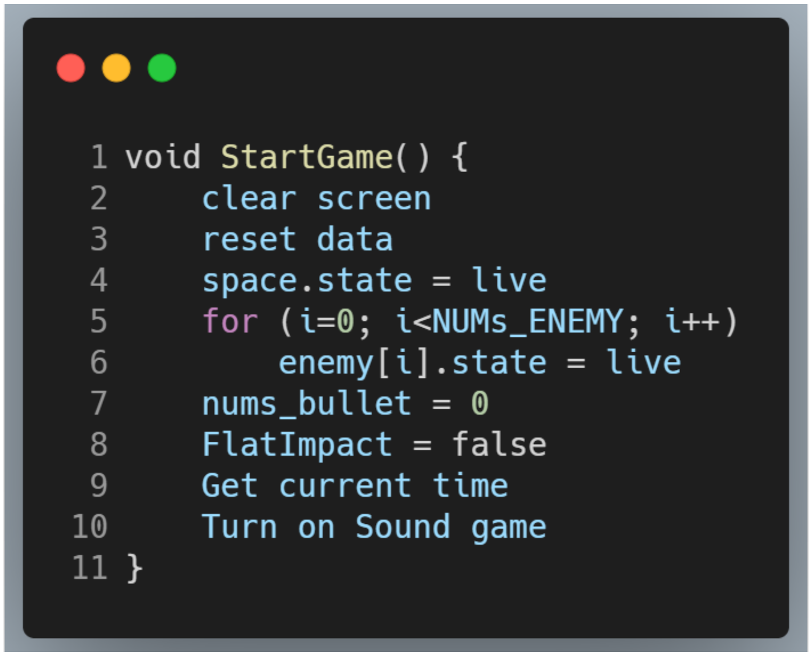


```
void SaveGame() {
    open file "Data.txt"  <append mode>
    Get current time{
        s = now second
        m = now minute
    }
        do
        {
            print("Enter name: ") -> name
            if (!CheckName(name)) {
                Enter name again
            }
        } while (!CheckName(name) || name.length() > 20);

        //Name Score Life Round Coin Nums_bullet Hp Coord_Space State NumSkin Sec Min Coord_Enemy State SkillShot
        Save in file bellow this order{
        name - Score - Life - Round - coin - nums_bullet - HP - space.getX() - space.getY() - space.state() - numSkin - s - m
        }
        for (int i = 0; i < NUMs_ENEMY; i++) {
            Save in file bellow this order{
                enemy[i].getX() - enemy[i].getY() - enemy[i].state()
            }
        }
        Save in file SkillShot
        Close file
    }
}
```



```
void Shot(int x, int y) {   // x, y is space's coordinate
    bool flatt = false;
    // flatt = true: fire
    // flatt = false: not fire
    if (ammo left != 0) {
        for (i=0;i<nums_bullet;i++)
            if (shoted)
                flatt = true
        if (flatt = false && Skill Shot = 1) {
            Set coordinate (bullet[nums_bullet]) = (x,y-1)
            bullet[nums_bullet++].state = live
        }
        else {
            if (flatt = false && Skill Shot = 2) {
                Set coordinate (bullet[nums_bullet]) = (x-1,y-1)
                bullet[nums_bullet++]. state = live
                Set coordinate(bullet[nums_bullet]) = (x + 1, y - 1)
                bullet[nums_bullet++].state = live
            }
        }
    }
    else
        print ("You run out of bullet")
}
```



```
void StartGame() {
    clear screen
    reset data
    space.state = live
    for (i=0; i<NUMs_ENEMY; i++)
        enemy[i].state = live
    nums_bullet = 0
    FlatImpact = false
    Get current time
    Turn on Sound game
}
```

c. GAME

- Start game
  -
  - New Game: Start new game.
    - A: Move left
    - D: Move right
    - S: Move down
    - W: Move up
    - L : Save Game
    - P: Pause Game
    - K: Open shop
    - 5: Fire
    - Esc: Exit Game
    - space: pass the round 1 (hack)
    - I: Kill the boss in round 2 (set boss HP=0) (hack)
  - Continue Game: Continue the last game had saved .
  - Shop: View shop.
  - Menu: Back to the start menu.
- Rule: Description the detail rule.
- About: Description the information of project
- Quit game: Out the game.

3. SUPPORT TOOL AND REFERENCE

a. SUPPORT TOOL

IDE: Microsoft Visual Studio

b. REFERENCE:

https://www.youtube.com/watch?time_continue=22&v=cXuTRyT6QLc&feature=emb_logo

https://opengameart.org/content/library-of-game-sounds

https://stackoverflow.com/questions/29574849/how-to-change-text-color-and-console-color-in-codeblocks

https://tranhanhuy.wordpress.com/2011/07/10/c-ham-mau-dung-cho-vc-6-0-textcolor/

https://www.stdio.vn/article/std-thread-trong-c-cQQFs

4. LINK DEMO (YOUTUBE):

5. LICENSE:

Teacher/ Lecturers can use video/ source code to demo for after course/ another person