**Introduction**

“DRAGON BALL Z: OUR LAST HOPE” is a 2D game. This game is built using C++ Language (Programing Language).It is created and designed by Samin Shahriar Tokey & Biozid Rahman Niloy.

**Story Line**

After Majin Buu was defeated, the Z fighters thought they had brought peace back to mankind, but they were wrong. Babidi opened the gate of Hell with Dr Gero from where Cell is sending his army of Cell Jrs. Meanwhile Freiza was resurrected and he is training very hard for his revenge on the Z fighters.

The army of Cell jrs were too many and they took out all the Z fighters.

Far away from Earth, Goku was training with Whis. When he came to know about this situation, he used instant transmittion and return back to Earth. Earth is now in chaos, the only Z fighter left is Goku. He is our only Hope for defending the Earth.

**Features**

1. **Endless Gameplay:** The objective of the game is to score as many points as possible.
2. **Collecting Dragon Balls:** Collecting Dragon Ball (all 7 of them) gives the player ability of transforming into a Super Saiyan and toggle between Super Saiyan form and Normal form.

**3) Collecting Senzu Beans:** Collecting Senzu Beans help the player regain his health.

**4) Obstacles:** There are different types and heights of obstacles that makes the player lose health in contact.

**a) Cell Jrs :** They come at the players with speed. The speed increases with distance.

**b) Mountains:** There are 3 shapes of mountains that blocks the way.

**c) Spikes:** Spikes comes randomly on the ground.

**d) Frieza Blasts:** In time of Boss battle, Frieza will use his Energy Ball against the player.

**5) Blasts:** The player can use his powerful blasts to remove obstacles from his way.

**a) Normal blast:** Less speed, more points.

**a) Super blast:** More speed, less points.

**6) Jump:** The player can jump to dodge obstacles.

**7) Boss Battle:** The player will eventually face Frieza who was two forms of his own.

1. **Regular Frieza :** Regular blast speed
2. **Golden Frieza** : Increased Blast speed

**8) High Scores:** There is a High Score page where we can see top 8 high scores.

**9) Cheats:** There are cheats built in the code to make the gameplay easier for some people.

Softwares Used

* Adobe Photoshop CC
* Microsoft Visual Studio 2013
* PhotoScape
* Adobe Lightroom 5.5
* Notepad

**Implementation Details**

**Functions Designated For Goku :**

Function to make Goku run :

void goku\_run() {

if (pause\_state == 0 && !(sp\_sc >= 0 && sp\_sc <= 2) && !(st\_sc >= 0 && st\_sc <= 4) && go == 0)

{

run++;

if (run >= 8)

run = 0;

}

}

Function to make Goku Jump :

void goku\_jump() {

if (pause\_state == 0 && !(sp\_sc >= 0 && sp\_sc <= 2) && !(st\_sc >= 0 && st\_sc <= 4) && go == 0){

if (jumping == 1){

jump\_dif--;

jump\_y += jump\_dif;

if (jump\_dif < 0)

jumping = 2;

if (jump\_y == run\_y) {

jump = 0;

}

else if (jump\_y <= jump\_h2) {

jump = 1;

}

else if (jump\_y < jump\_limit) {

jump = 2;

}

}

else if (jumping == 2) {

jump\_dif--;

jump\_y += jump\_dif;

if (jump\_y >= jump\_h1) {

jump = 0;

}

else if (jump\_y >= jump\_h2) {

jump = 1;

}

else if (jump\_y > run\_y) {

jump = 2;

}

if (jump\_y <= 140)

{

jumping = 0;

jump\_co = 0;

jump\_limit = 450;

jump\_dif = jd;

jump\_y = 140;

}

}

}

}

Function to make Gokus’ Blasts:

void goku\_blast(){

if (pause\_state == 0 && !(sp\_sc >= 0 && sp\_sc <= 2) && !(st\_sc >= 0 && st\_sc <= 4) && go == 0){

if (blast\_button != 0)

{

blast++;

if (blast > 5)

blast\_x += blast\_speed;

if (blast >= 200)

blast = 0;

}

else{

blast = 0;

}

if (blast1\_button != 0)

{

blast1++;

if (blast1 > 5)

blast1\_x += blast\_speed;

if (blast1 >= 200)

blast1 = 0;

}

else{

blast1 = 0;

}

if (blast2\_button != 0)

{

blast2++;

if (blast2 > 5)

blast2\_x += blast\_speed;

if (blast2 >= 200)

blast2 = 0;

}

else{

blast2 = 0;

}

if (blast3\_button != 0)

{

blast3++;

if (blast3 > 5)

blast3\_x += blast\_speed;

if (blast3 >= 200)

blast3 = 0;

}

else{

blast3 = 0;

}

if (blast4\_button != 0)

{

blast4++;

if (blast4 > 5)

blast4\_x += blast\_speed;

if (blast4 >= 200)

blast4 = 0;

}

else{

blast4 = 0;

}

if (blast5\_button != 0)

{

blast5++;

if (blast5 > 5)

blast5\_x += blast\_speed;

if (blast5 >= 200)

blast5 = 0;

}

else{

blast5 = 0;

}

if (blast6\_button != 0)

{

blast6++;

if (blast6 > 5)

blast6\_x += blast\_speed;

if (blast6 >= 200)

blast6 = 0;

}

else{

blast6 = 0;

}

if (blast7\_button != 0)

{

blast7++;

if (blast7 > 5)

blast7\_x += blast\_speed;

if (blast7 >= 200)

blast7 = 0;

}

else{

blast7 = 0;

}

if (blast8\_button != 0)

{

blast8++;

if (blast8 > 5)

blast8\_x += blast\_speed;

if (blast8 >= 200)

blast8 = 0;

}

else{

blast8 = 0;

}

if (blast9\_button != 0)

{

blast9++;

if (blast9 > 5)

blast9\_x += blast\_speed;

if (blast9 >= 200)

blast9 = 0;

}

else{

blast9 = 0;

}

}

}

**Functions Designated For Background :**

Function to make the Ground Move:

void groundMovement() {

ground\_x -= gdx;

ground\_x1 -= gdx;

ground\_x2 -= gdx;

ground\_x3 -= gdx;

ground\_x4 -= gdx;

ground\_x5 -= gdx;

ground\_x6 -= gdx;

ground\_x7 -= gdx;

ground\_x8 -= gdx;

ground\_x9 -= gdx;

ground\_x10 -= gdx;

if (ground\_x < 0) {

ground\_x = 1273;

}

if (ground\_x1 < 0) {

ground\_x1 = 1273;

}

if (ground\_x2 < 0) {

ground\_x2 = 1273;

}

if (ground\_x3 < 0) {

ground\_x3 = 1273;

}

if (ground\_x4 <0) {

ground\_x4 = 1273;

}

if (ground\_x5 < 0) {

ground\_x5 = 1273;

}

if (ground\_x6 < 0) {

ground\_x6 = 1273;

}

if (ground\_x7 < 0) {

ground\_x7 = 1273;

}

if (ground\_x8 < 0) {

ground\_x8 = 1273;

}

if (ground\_x9 < 0) {

ground\_x9 = 1273;

}

if (ground\_x10 < 0) {

ground\_x10 = 1273;

}

Function to make the Background Move:

void background() {

bck\_x -= bck\_speed;

bck\_x1 -= bck\_speed;

bck\_x2 -= bck\_speed;

bck\_x3 -= bck\_speed;

bck\_x4 -= bck\_speed;

bck\_x5 -= bck\_speed;

bck\_x6 -= bck\_speed;

bck\_x7 -= bck\_speed;

bck\_x8 -= bck\_speed;

bck\_x9 -= bck\_speed;

bck\_x10 -= bck\_speed;

if (bck\_x < 0) {

bck\_x = 1403;

}

if (bck\_x1 < 0) {

bck\_x1 = 1403;

}

if (bck\_x2 < 0) {

bck\_x2 = 1403;

}

if (bck\_x3 < 0) {

bck\_x3 = 1403;

}

if (bck\_x4 <0) {

bck\_x4 = 1403;

}

if (bck\_x5 < 0) {

bck\_x5 = 1403;

}

if (bck\_x6 < 0) {

bck\_x6 = 1403;

}

if (bck\_x7 < 0) {

bck\_x7 = 1403;

}

if (bck\_x8 < 0) {

bck\_x8 = 1403;

}

if (bck\_x9 < 0) {

bck\_x9 = 1403;

}

if (bck\_x10 < 0) {

bck\_x10 = 1403;

}

}

**Functions Designated For Cells:**

Function to make Cells Move:

void cell\_move() {

cell\_x -= cell\_speed;

if (cell\_x <= 0) {

cell\_x = 1300;

random\_cell = rand() % 4;

}

else if (cell == 1) {

cell\_x = 1450;

random\_cell = rand() % 4;

}

if (random\_cell == 1) {

cell\_y = cell1\_y;

}

else if (random\_cell == 2) {

cell\_y = cell2\_y;

}

else if (random\_cell == 3) {

cell\_y = cell3\_y;

}

}

**Functions Designated For Frieza:**

Function to make Frieza Move:

void frieza\_move() {

frieza\_x -= 20;

if (frieza\_x <= 1100) {

frieza\_x = 1100;

}

if (f == 0){

frieza\_y += frieza\_speed;

if (frieza\_y >= 700) {

f = 1;

}

}

if (f == 1) {

frieza\_y -= frieza\_speed;

if (frieza\_y <= 160) {

f = 0;

}

}

}

Function to make Friezas’ Blasts:

void frieza\_blast() {

if (gol\_frieza == 0) {

if (frieza\_y < 180) {

ff = 0;

f\_b = f\_blast;

f\_blast\_x = frieza\_x - 20;

f\_blast\_y = frieza\_y + 27;

f\_ap = 2;

}

if (frieza\_y > 180 && frieza\_y < 350) {

f\_ap = 0;

}

if (frieza\_y > 350 && frieza\_y < 430) {

ff1 = 0;

f\_b1 = f\_blast1;

f\_blast1\_x = frieza\_x - 20;

f\_blast1\_y = frieza\_y + 27;

if (frieza\_y < 400) {

f\_ap = 3;

}

else

f\_ap = 1;

}

if (frieza\_y > 430 && frieza\_y < 480) {

ff2 = 0;

f\_b2 = f\_blast2;

f\_blast2\_x = frieza\_x - 20;

f\_blast2\_y = frieza\_y + 27;

f\_ap = 2;

}

if (frieza\_y > 490 && frieza\_y < 560) {

ff3 = 0;

f\_b3 = f\_blast3;

f\_blast3\_x = frieza\_x - 20;

f\_blast3\_y = frieza\_y + 27;

f\_ap = 1;

}

if (frieza\_y > 560 && frieza\_y < 650) {

f\_ap = 0;

if (frieza\_y < 600) {

f\_ap = 3;

}

}

if (frieza\_y > 650) {

ff4 = 0;

f\_b4 = f\_blast4;

f\_blast4\_x = frieza\_x - 20;

f\_blast4\_y = frieza\_y + 27;

f\_ap = 2;

}

if (ff == 0) {

if (f\_b == f\_blast) {

f\_blast\_x -= f\_blast\_speed;

}

}

else if (ff == 1) {

f\_blast\_x = 2300;

}

if (ff1 == 0) {

if (f\_b == f\_blast1) {

f\_blast1\_x -= f\_blast\_speed;

}

}

else if (ff1 == 1) {

f\_blast1\_x = 2300;

}

if (ff2 == 0) {

if (f\_b == f\_blast2) {

f\_blast2\_x -= f\_blast\_speed;

}

}

else if (ff2 == 1) {

f\_blast2\_x = 2300;

}

if (ff3 == 0) {

if (f\_b == f\_blast3) {

f\_blast3\_x -= f\_blast\_speed;

}

}

else if (ff3 == 1) {

f\_blast3\_x = 2300;

}

if (ff4 == 0) {

if (f\_b == f\_blast4) {

f\_blast4\_x -= f\_blast\_speed;

}

}

else if (ff4 == 1) {

f\_blast4\_x = 2300;

}

}

else if (gol\_frieza == 1) {

if (frieza\_y < 180) {

ff = 0;

f\_b = f\_blast;

f\_blast\_x = frieza\_x - 20;

f\_blast\_y = frieza\_y + 27;

f\_ap = 2;

}

if (frieza\_y > 180 && frieza\_y < 350) {

f\_ap = 0;

}

if (frieza\_y > 350 && frieza\_y < 430) {

ff1 = 0;

f\_b1 = f\_blast1;

f\_blast1\_x = frieza\_x - 20;

f\_blast1\_y = frieza\_y + 27;

if (frieza\_y < 400) {

f\_ap = 3;

}

else

f\_ap = 1;

}

if (frieza\_y > 430 && frieza\_y < 480) {

ff2 = 0;

f\_b2 = f\_blast2;

f\_blast2\_x = frieza\_x - 20;

f\_blast2\_y = frieza\_y + 27;

f\_ap = 2;

}

if (frieza\_y > 490 && frieza\_y < 560) {

ff3 = 0;

f\_b3 = f\_blast3;

f\_blast3\_x = frieza\_x - 20;

f\_blast3\_y = frieza\_y + 27;

f\_ap = 1;

}

if (frieza\_y > 560 && frieza\_y < 650) {

f\_ap = 0;

if (frieza\_y < 600) {

f\_ap = 3;

}

}

if (frieza\_y > 650) {

ff4 = 0;

f\_b4 = f\_blast4;

f\_blast4\_x = frieza\_x - 20;

f\_blast4\_y = frieza\_y + 27;

f\_ap = 2;

}

if (ff == 0) {

if (f\_b == f\_blast) {

f\_blast\_x -= f\_blast\_speed;

}

}

else if (ff == 1) {

f\_blast\_x = 2300;

}

if (ff1 == 0) {

if (f\_b == f\_blast1) {

f\_blast1\_x -= f\_blast\_speed;

}

}

else if (ff1 == 1) {

f\_blast1\_x = 2300;

}

if (ff2 == 0) {

if (f\_b == f\_blast2) {

f\_blast2\_x -= f\_blast\_speed;

}

}

else if (ff2 == 1) {

f\_blast2\_x = 2300;

}

if (ff3 == 0) {

if (f\_b == f\_blast3) {

f\_blast3\_x -= f\_blast\_speed;

}

}

else if (ff3 == 1) {

f\_blast3\_x = 2300;

}

if (ff4 == 0) {

if (f\_b == f\_blast4) {

f\_blast4\_x -= f\_blast\_speed;

}

}

else if (ff4 == 1) {

f\_blast4\_x = 2300;

}

}

}

**Functions Designated For Spikes:**

void spike\_move() {

if (random == 1) {

spike\_direction = 1;

}

else if (random == 2) {

spike\_direction = 2;

}

else if (random == 3) {

spike\_direction = 3;

}

else if (random == 4) {

spike\_direction = 4;

}

else if (random == 5) {

spike\_direction = 5;

}

else if (random == 6) {

spike\_direction = 6;

}

if (spike\_direction == 1) {

spike\_x -= spike\_speed;

if (spike\_x <= 0) {

spikehit = 0;

spike\_x = 1300;

random = rand() % 7;

}

}

else if (spike\_direction == 2) {

spike2\_x -= spike\_speed;

if (spike2\_x <= 0) {

spikehit2 = 0;

spike2\_x = 1350;

random = rand() % 7;

}

}

else if (spike\_direction == 3) {

spike3\_x -= spike\_speed;

spike4\_x -= spike\_speed;

if (spike4\_x <= 0) {

spikehit3 = 0;

spikehit4 = 0;

spike3\_x = 1300;

spike4\_x = spike3\_x + spike34\_diff;

random = rand() % 7;

}

}

else if (spike\_direction == 4) {

spike5\_x -= spike\_speed;

spike6\_x -= spike\_speed;

if (spike6\_x <= 0) {

spikehit5 = 0;

spikehit6 = 0;

spike5\_x = 1300;

spike6\_x = spike3\_x + spike56\_diff;

random = rand() % 7;

}

}

else if (spike\_direction == 5) {

spike7\_x -= spike\_speed;

spike8\_x -= spike\_speed;

if (spike8\_x <= 0) {

spikehit7 = 0;

spikehit8 = 0;

spike7\_x = 1300;

spike8\_x = spike7\_x + spike78\_diff;

random = rand() % 7;

}

}

else if (spike\_direction == 6) {

spike9\_x -= spike\_speed;

spike10\_x -= spike\_speed;

spike11\_x -= spike\_speed;

if (spike11\_x <= 0) {

spikehit9 = 0;

spikehit10 = 0;

spikehit11 = 0;

spike9\_x = 1300;

spike10\_x = spike9\_x + spike910\_diff;

spike11\_x = spike10\_x + spike1011\_diff;

random = rand() % 7;

}

}

}

**Functions Designated For Obstacles:**

void obstacle() {

if (randomob == 1) {

obs = 1;

}

if (randomob == 2) {

obs = 2;

}

if (randomob == 3) {

obs = 3;

}

if (obs == 1) {

obs1\_x -= obs\_speed;

if (obs1\_x < 0) {

obs1\_x = 1400;

randomob = rand() % 4;

}

}

if (obs == 2) {

obs2\_x -= obs\_speed;

if (obs2\_x <= 0) {

obs2\_x = 1550;

randomob = rand() % 4;

}

}

if (obs == 3) {

obs3\_x -= obs\_speed;

if (obs3\_x <= 0) {

obs3\_x = 1650;

randomob = rand() % 4;

}

}

}

Button Layout

|  |  |
| --- | --- |
| Properties | Buttons |
| Jump | Space/Mouse Right |
| Fire Blasts | X/Mouse Left |
| Toggle Super Saiyan Mode | S |
| Menu Navigations | Up,Down,Left,Right |
| Menu Selection | Return |
| Pause Screen | P |
| Returning From a Screen | Backspace |
| Skipping Storyline | Space |

Cheats:

Type ‘saiyan’ to go Super Saiyan without collecting dragonballs.

Type ‘hell’ to summon Lord Frieza.

Type ‘ginyu’ to force transform Frieza.

**Facts**

* The Source Code Has 6110 lines
* 706 High Resolution Images Were Used
* The Footage Shown In The Game Are All Images
* The Game is Approximately 2.1GB

Future Works

* Including more boss battle with different enemies.
* Including more characters.
* Adding immersive sound.
* Making the source code efficient.
* Adding efficient algorithms.
* Story Mode and Arcade Mode.