

```

#include <SDL.h>
#include <SDL_image.h>
#include <SDL_mixer.h>
#include <iostream>
#include <stdio.h>
#include <vector>

using namespace std;

//function to load textures

SDL_Texture *LoadTexture(std::string filePath, SDL_Renderer *renderTarget)
{
    SDL_Texture *texture= NULL;
    SDL_Surface *surface = IMG_Load(filePath.c_str());
    if(surface==NULL)
    {
        cout<<"error"<<endl;
    }
    else
    {
        texture=SDL_CreateTextureFromSurface(renderTarget,surface);
        if(texture==NULL)
        {
            cout<<"error"<<endl;
        }
    }
    SDL_FreeSurface(surface);
    return texture;
}

```

```
}
```

```
//main function starts here
```

```
int main(int argc,char *argv[])
```

```
{
```

```
    int n=0;
```

```
    //defining all textures and rectangles which will be used
```

```
    SDL_Window *window = NULL;
```

```
    SDL_Texture *currentImage=NULL;
```

```
    SDL_Texture *initial=NULL;
```

```
    SDL_Texture *entrystickman=NULL;
```

```
    SDL_Texture *exitstickman=NULL;
```

```
    SDL_Texture *effectImage=NULL;
```

```
    SDL_Texture *background=NULL;
```

```
    SDL_Texture *startbutton=NULL;
```

```
    SDL_Texture *adios=NULL;
```

```
    SDL_Renderer *renderTarget=NULL;
```

```
    SDL_Rect playerRect;
```

```
    SDL_Rect exitstickmanRect;
```

```
    SDL_Rect effectRect;
```

```
    SDL_Texture *quitIMG=NULL;
```

```
    SDL_Texture *instructions=NULL;
```

```
    SDL_Texture *finaleffect=NULL;
```

```
    vector<SDL_Rect> finalRect;
```

```
    vector<SDL_Rect> finalPosition;
```

```
    SDL_Rect inst;
```

```
    inst.x=1097;
```

```
inst.y=135;
```

```
inst.w=150;
```

```
inst.h=300;
```

```
int m=-1;
```

```
SDL_Rect quitRect;
```

```
quitRect.x=1097;
```

```
quitRect.y=50;
```

```
quitRect.h=80;
```

```
quitRect.w=169;
```

```
SDL_Rect centreRect;
```

```
centreRect.x=483;
```

```
centreRect.y=300;
```

```
centreRect.w=400;
```

```
centreRect.h=200;
```

```
SDL_Rect backRect;
```

```
backRect.x=0;
```

```
backRect.y=0;
```

```
backRect.w=1366;
```

```
backRect.h=0;
```

```
bool sprites=false;
```

```
bool finish=false;;
```

```
SDL_Rect playerPosition;
```

```
SDL_Rect effectPosition;
```

```
playerPosition.x=600;
```

```

playerPosition.y=-320;

int i=0;

int oldplayerPosition=playerPosition.y;

int frameWidth,frameHeight;

int textureWidth,textureHeight;

int etextureWidth,etextureHeight;

int eframeWidth,eframeHeight;

int ftextureWidth,ftextureHeight;

int fframeWidth,fframeHeight;

float frameTime=0;

int prevTime=0;

int currentTime=0;

float deltaTime=0;

float moveSpeed=200.0f;


//initialize SDL

SDL_Init(SDL_INIT_VIDEO || SDL_INIT_AUDIO);


int imgFlags = IMG_INIT_PNG;

if( !( IMG_Init( imgFlags ) & imgFlags ) )

    printf( "SDL_image could not initialize! SDL_image Error: %s\n", IMG_GetError() );


if (Mix_OpenAudio(44100,MIX_DEFAULT_FORMAT,2,2048)<0)

{

    printf("SDL_MIXER error: %s\n", SDL_GetError());

}

Mix_Chunk *sfire=NULL;


Mix_Chunk *laugh=NULL;

```

```
Mix_Chunk *walk=NULL;
```

```
Mix_Chunk *exitjump=NULL;
```

```
Mix_Chunk *stear=NULL;
```

```
Mix_Chunk *crack=NULL;
```

```
Mix_Chunk *jumpsound=NULL;
```

```
Mix_Music *backmsc=NULL;
```

```
backmsc=Mix_LoadMUS("theme.wav");
```

```
sfire=Mix_LoadWAV("fire.wav");
```

```
walk=Mix_LoadWAV("walk.wav");
```

```
stear=Mix_LoadWAV("tear.wav");
```

```
crack=Mix_LoadWAV("crack.wav");
```

```
laugh=Mix_LoadWAV("laugh.wav");
```

```
exitjump=Mix_LoadWAV("exitjump.wav");
```

```
jumpsound=Mix_LoadWAV("jump.wav");
```

```
if(backmsc==NULL)
```

```
{
```

```
    printf("Couldn't find theme.wav: %s\n", Mix_GetError());
```

```
}
```

```
if(jumpsound==NULL)
```

```
{  
    printf("Couldn't find jumsound.wav: %s\n", Mix_GetError());  
}  
if(sfire==NULL)  
{  
    printf("Couldn't find fire.mp3: %s\n", Mix_GetError());  
}  
if(crack==NULL)  
{  
    printf("Couldn't find crack.wav: %s\n", Mix_GetError());  
}  
if(stear==NULL)  
{  
    printf("Couldn't find tear.wav: %s\n", Mix_GetError());  
}  
if(exitjump==NULL)  
{  
    printf("Couldn't find exitjump.wav: %s\n", Mix_GetError());  
}  
if(walk==NULL)  
{  
    printf("Couldn't find walk.wav: %s\n", Mix_GetError());  
}  
if(laugh==NULL)  
{  
    printf("Couldn't find laugh.wav: %s\n", Mix_GetError());  
}
```

```
    window =  
SDL_CreateWindow("Stickman",SDL_WINDOWPOS_CENTERED,SDL_WINDOWPOS_CENTERED,1366,768,  
SDL_WINDOW_FULLSCREEN_DESKTOP );
```

```
    renderTarget=SDL_CreateRenderer(window,-1, SDL_RENDERER_ACCELERATED |  
SDL_RENDERER_PRESENTVSYNC | SDL_RENDERER_TARGETTEXTURE);
```

```
currentImage = LoadTexture("image.png",renderTarget);
```

```
instructions = LoadTexture("instructions.png",renderTarget);
```

```
if( currentImage == NULL )  
{  
    printf( "Unable to load image! SDL_image Error: %s\n", IMG_GetError() );  
}
```

```
background = LoadTexture("capture.png",renderTarget);
```

```
if( background == NULL )  
{  
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );  
}
```

```
initial = LoadTexture("initial.png",renderTarget);
```

```
if( initial == NULL )  
{  
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );  
}
```

```
}
```

```
entrystickman = LoadTexture("entrystickman.png",renderTarget);
```

```
if( entrystickman == NULL )
```

```
{
```

```
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );
```

```
}
```

```
exitstickman = LoadTexture("exitstickman.png",renderTarget);
```

```
if( exitstickman == NULL )
```

```
{
```

```
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );
```

```
}
```

```
effectImage = LoadTexture("effects.png",renderTarget);
```

```
if( effectImage == NULL )
```

```
{
```

```
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );
```

```
}
```

```
quitIMG= LoadTexture("exit.png",renderTarget);
```

```
if( quitIMG == NULL )
```

```
{
```

```
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );
```

```
}
```



```
finaleffect = LoadTexture("permanent.png",renderTarget);
```

```
if( finaleffect == NULL )  
{  
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );  
}
```

```
startbutton=LoadTexture("start.png",renderTarget);
```

```
if( startbutton == NULL )  
{  
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );  
}
```

```
adios = LoadTexture("adios.png",renderTarget);
```

```
if( adios == NULL )  
{  
    printf( "Unable to load image ! SDL_image Error: %s\n", IMG_GetError() );  
}
```

```
SDL_QueryTexture(currentImage,NULL,NULL,&textureWidth,&textureHeight);
```

```
bool jump=false,fire=false,tear=false;
```

```
frameWidth = textureWidth/5;
```

```
frameHeight = textureHeight/15;
```

```
playerRect.x=playerRect.y=0;
```

```
playerRect.w=frameWidth;
```

```
playerRect.h=frameHeight;
```

```
SDL_QueryTexture(effectImage,NULL,NULL,&etextureWidth,&etextureHeight);
```

```
eframeWidth = etextureWidth/5;
```

```
eframeHeight = etextureHeight/6;
```

```
effectRect.x=0;
```

```
effectRect.y=5 * eframeHeight;
```

```
effectRect.w=eframeWidth;
```

```
effectRect.h=eframeHeight;
```

```
SDL_QueryTexture(finaleffect,NULL,NULL,&ftextureWidth,&ftextureHeight);
```

```
fframeWidth = ftextureWidth/3;
```

```
fframeHeight = ftextureHeight;
```

```
playerPosition.w=frameWidth;
```

```
playerPosition.h=frameHeight;
```

```
effectPosition.w=eframeWidth;
```

```
effectPosition.h=eframeHeight;
```

```
SDL_SetRenderDrawColor(renderTarget,0,0,0,0);
```

```
bool l= true;
```

```
SDL_Event start;
```

```
while(l)
```

```
{
```

```

while(SDL_PollEvent(&start)!=0 )
{
    int x,y;
    SDL_GetMouseState (&x, &y);
    if (start.type == SDL_MOUSEBUTTONDOWN)
    {
        if (start.button.button == SDL_BUTTON_LEFT && x< 883 && x>483 && y>300 && y<500 )
            l = false;
    }
}
SDL_RenderClear(renderTarget);
SDL_RenderCopy(renderTarget,initial,NULL,NULL);
SDL_RenderCopy(renderTarget,startbutton,NULL,&centreRect);
SDL_RenderPresent(renderTarget);
}

```

```

Mix_PlayMusic(backmsc,-1);
Mix_VolumeMusic(MIX_MAX_VOLUME/2);
Mix_PlayChannel( -1, laugh, 0 );

```

```

for (int k=0; backRect.h<768; k+=1)
{
    SDL_RenderClear(renderTarget);
    SDL_RenderCopy(renderTarget,initial,NULL,NULL);
    SDL_RenderCopy(renderTarget,background,&backRect,&backRect);
    SDL_RenderCopy(renderTarget,entrystickman,NULL,&playerPosition);
    SDL_RenderPresent(renderTarget);
    SDL_Delay(50);
    backRect.h+=k;
}

```

```
    playerPosition.y+=k;  
}
```

```
playerPosition.y=430;  
oldplayerPosition=playerPosition.y;
```

```
bool isRunning=true;  
SDL_Event ev;
```

```
while(isRunning)  
{
```

```
    int x,y ;  
    SDL_GetMouseState (&x, &y);
```

```
    prevTime=currentTime;  
    currentTime=SDL_GetTicks();  
    deltaTime= (currentTime - prevTime)/1000.0f;  
    while(SDL_PollEvent(&ev)!=0 )  
    {
```

```
        if(ev.type == SDL_QUIT)  
            isRunning=false;
```

```
        if (ev.type == SDL_MOUSEBUTTONDOWN)  
        {
```

```
            if (ev.button.button == SDL_BUTTON_LEFT && x< 1266 && x>1097 && y>50 && y<638 )  
                isRunning = false;  
        }
```

```

else if(SDL_KEYDOWN)
{
    sprites=true;
    switch(ev.key.keysym.sym)
    {
    case SDLK_t:
        m++;
        tear=true;
        finish=false;
        if(n==0)
        {
            effectPosition.x=playerPosition.x + frameWidth/2 - 15;
            effectPosition.y=playerPosition.y;
            playerRect.y=8*frameHeight;
            finalPosition.push_back (SDL_Rect());
            finalRect.push_back (SDL_Rect());
            finalPosition[m].x=playerPosition.x + frameWidth/2 -15;
            finalPosition[m].y= playerPosition.y;
            finalPosition[m].w=fframeWidth;
            finalPosition[m].h=fframeHeight;
            finalRect[m].x =0;
            finalRect[m].y= 0;
            finalRect[m].w= fframeWidth;
            finalRect[m].h= fframeHeight;
        }
    else
    {

```

```

    effectPosition.x=playerPosition.x - frameWidth/2 + 15;
    effectPosition.y=playerPosition.y;
    playerRect.y=9*frameHeight;
    finalPosition.push_back (SDL_Rect());
    finalRect.push_back (SDL_Rect());
    finalPosition[m].x=playerPosition.x - frameWidth/2 + 15;
    finalPosition[m].y= playerPosition.y;
    finalPosition[m].w=fframeWidth;
    finalPosition[m].h=fframeHeight;
    finalRect[m].x =0;
    finalRect[m].y= 0;
    finalRect[m].w= fframeWidth;
    finalRect[m].h= fframeHeight;
}
effectRect.y=0;
Mix_PlayChannel( -1, stear, 0 );
break;

```

```

case SDLK_j:
    playerPosition.y-=100;
    if(n==0)
    {
        playerRect.y=4*frameHeight;
    }
    else
    {
        playerRect.y=5*frameHeight;
    }
    jump=true;

```

```

    Mix_PlayChannel( -1, jumpsound, 0 );

    break;

case SDLK_RIGHT:

    playerPosition.x+=moveSpeed * deltaTime ;

    playerRect.y=2*frameHeight;

    n=0;

    Mix_PlayChannel( -1, walk, 0 );

    break;

case SDLK_LEFT:

    playerPosition.x-=moveSpeed * deltaTime ;

    playerRect.y=3*frameHeight;

    n=1;

    Mix_PlayChannel( -1, walk, 0 );

    break;

case SDLK_q:

    isRunning=false;

case SDLK_UP:

    playerPosition.y-=moveSpeed * deltaTime ;

    oldplayerPosition=playerPosition.y;

    break;

case SDLK_DOWN:

    playerPosition.y+=moveSpeed * deltaTime ;

    oldplayerPosition=playerPosition.y;

    break;

case SDLK_p:

    m++;

    finish=false;

    if(n==0)

    {

```

```

    effectPosition.x=playerPosition.x + frameWidth/2 - 45;
    effectPosition.y=playerPosition.y - 15;
    playerRect.y=6*frameHeight;
    finalPosition.push_back (SDL_Rect());
    finalRect.push_back (SDL_Rect());
    finalPosition[m].x=playerPosition.x + frameWidth/2 -45;
    finalPosition[m].y= playerPosition.y - 15;
    finalPosition[m].w=fframeWidth;
    finalPosition[m].h=fframeHeight;
    finalRect[m].x =2 * fframeWidth;
    finalRect[m].y= 0;
    finalRect[m].w= fframeWidth;
    finalRect[m].h= fframeHeight;

}
else
{
    effectPosition.x=playerPosition.x - frameWidth/2 + 15;
    effectPosition.y=playerPosition.y ;
    playerRect.y=7*frameHeight;
    finalPosition.push_back (SDL_Rect());
    finalRect.push_back (SDL_Rect());
    finalPosition[m].x=playerPosition.x - frameWidth/2 -15;
    finalPosition[m].y= playerPosition.y ;
    finalPosition[m].w=fframeWidth;
    finalPosition[m].h=fframeHeight;
    finalRect[m].x =2 * fframeWidth;
    finalRect[m].y= 0;
    finalRect[m].w= fframeWidth;

```



```

        finalRect[m].h= fframeHeight;
    }
    effectRect.y=4*eframeHeight;
    Mix_PlayChannel( -1, crack, 0 );
    break;
case SDLK_f:
    fire=true;
    m++;
    finish=false;
    if(n==0)
    {
        effectPosition.x=playerPosition.x + frameWidth/2 + 115;
        effectPosition.y=playerPosition.y - 30;
        playerRect.y=10*frameHeight;
        effectRect.y=2*eframeHeight;
        finalPosition.push_back (SDL_Rect());
        finalRect.push_back (SDL_Rect());
        finalPosition[m].x=playerPosition.x + frameWidth/2 + 115;
        finalPosition[m].y= playerPosition.y - 30;
        finalPosition[m].w=fframeWidth;
        finalPosition[m].h=fframeHeight;
        finalRect[m].x = fframeWidth;
        finalRect[m].y= 0;
        finalRect[m].w= fframeWidth;
        finalRect[m].h= fframeHeight;
    }
    else
    {
        effectPosition.x=playerPosition.x + frameWidth + 20 ;

```

```

    effectPosition.y=playerPosition.y - 30;
    effectPosition.x-=2 * frameWidth;
    playerRect.y=11*frameHeight;
    effectRect.y=1*eframeHeight;
    finalPosition.push_back (SDL_Rect());
    finalRect.push_back (SDL_Rect());
    finalPosition[m].x=playerPosition.x + frameWidth - 590;
    finalPosition[m].y= playerPosition.y - 30;
    finalPosition[m].w=fframeWidth;
    finalPosition[m].h=fframeHeight;
    finalRect[m].x =fframeWidth;
    finalRect[m].y= 0;
    finalRect[m].w= fframeWidth;
    finalRect[m].h= fframeHeight;
}

Mix_PlayChannel( -1, sfire, 0 );

break;

case SDLK_k:

    m++;
    finish=false;
    if(n==0)
    {
        effectPosition.x=playerPosition.x + frameWidth/2 - 30;
        effectPosition.y=playerPosition.y;
        playerRect.y=12*frameHeight;
        finalPosition.push_back (SDL_Rect());
        finalRect.push_back (SDL_Rect());
        finalPosition[m].x=playerPosition.x + frameWidth/2 -30;
        finalPosition[m].y= playerPosition.y;
    }

```

```

        finalPosition[m].w=fframeWidth;
        finalPosition[m].h=fframeHeight;
        finalRect[m].x =2 * fframeWidth;
        finalRect[m].y= 0;
        finalRect[m].w= fframeWidth;
        finalRect[m].h= fframeHeight;
    }
else
{
    effectPosition.x=playerPosition.x - frameWidth/2;
    effectPosition.y=playerPosition.y;
    playerRect.y=13*frameHeight;
    finalPosition.push_back (SDL_Rect());
    finalRect.push_back (SDL_Rect());
    finalPosition[m].x=playerPosition.x - frameWidth/2;
    finalPosition[m].y= playerPosition.y;
    finalPosition[m].w=fframeWidth;
    finalPosition[m].h=fframeHeight;
    finalRect[m].x =2 * fframeWidth;
    finalRect[m].y= 0;
    finalRect[m].w= fframeWidth;
    finalRect[m].h= fframeHeight;
}
effectRect.y=4*eframeHeight;
Mix_PlayChannel( -1, crack, 0 );
break;

```

```
    }  
}  
  
}
```

```
SDL_RenderClear(renderTarget);  
SDL_RenderCopy(renderTarget,background,NULL,NULL);  
SDL_RenderCopy(renderTarget,quitIMG,NULL,&quitRect);  
SDL_RenderCopy(renderTarget,instructions,NULL,&inst);  
if(fire)  
{  
    effectRect.y= 3 *eFrameHeight;  
    if(n==1)  
    {  
        effectPosition.x-=30;  
    }  
  
    SDL_RenderCopy(renderTarget,effectImage,&effectRect ,&effectPosition);  
    if(n==0)  
    {  
        effectRect.y= 2 * eFrameHeight;  
    }  
    else  
    {  
        effectRect.y= eFrameHeight;  
        effectPosition.x+=30;  
    }  
}
```

```

if(finish & (!tear))
{

    for(int j=0; j<m; j++)
    {
        if(finalRect[j].x!=0)
            SDL_RenderCopy(renderTarget,finaleffect,&finalRect[j] ,&finalPosition[j]);
    }
    for(int j=0; j<m; j++)
    {
        if(finalRect[j].x==0)
            SDL_RenderCopy(renderTarget,finaleffect,&finalRect[j] ,&finalPosition[j]);
    }

}

if((!finish) & (!tear))
{
    for(int j=0; j<m-1; j++)
    {
        if(finalRect[j].x!=0)
            SDL_RenderCopy(renderTarget,finaleffect,&finalRect[j] ,&finalPosition[j]);
    }
    for(int j=0; j<m-1; j++)
    {
        if(finalRect[j].x==0)
            SDL_RenderCopy(renderTarget,finaleffect,&finalRect[j] ,&finalPosition[j]);
    }
}

```

```
}
```

```
if(finish & (tear))
```

```
{
```

```
    for(int j=0; j<m; j++)
```

```
    {
```

```
        if(finalRect[j].x!=0)
```

```
            SDL_RenderCopy(renderTarget,finaleffect,&finalRect[j] ,&finalPosition[j]);
```

```
    }
```

```
    for(int j=0; j<m; j++)
```

```
    {
```

```
        if(finalRect[j].x==0)
```

```
            SDL_RenderCopy(renderTarget,finaleffect,&finalRect[j] ,&finalPosition[j]);
```

```
    }
```

```
}
```

```
if(!finish) & (tear))
```

```
{
```

```
    for(int j=0; j<m-1; j++)
```

```
    {
```

```
        if(finalRect[j].x!=0)
```

```
            SDL_RenderCopy(renderTarget,finaleffect,&finalRect[j] ,&finalPosition[j]);
```

```
    }
```

```
    for(int j=0; j<m-1; j++)
```

```
    {
```

```
        if(finalRect[j].x==0)
```

```
            SDL_RenderCopy(renderTarget,finaleffect,&finalRect[j] ,&finalPosition[j]);
```

```
    }
```

```
}
```

```
    if(!tear)
```

```
{
```

```
    SDL_RenderCopy(renderTarget,effectImage,&effectRect,&effectPosition);
```

```
}
```

```
if(tear)
```

```
{
```

```
    SDL_RenderCopy(renderTarget,effectImage,&effectRect,&effectPosition);
```

```
}
```

```
SDL_RenderCopy(renderTarget,currentImage,&playerRect,&playerPosition);
```

```
SDL_RenderPresent(renderTarget);
```

```
SDL_Delay(200);
```

```
frameTime+=deltaTime;
```

```
if(frameTime >= 0.2f && sprites)
```

```
{
```

```
    i++;
```

```
    frameTime=0;
```

```
    effectRect.x+=eframeWidth;
```

```
    playerRect.x+=frameWidth;
```

```
    if(playerRect.x >=5*frameWidth)
```

```
    {
```

```
        playerRect.x=0;
```

```
    }
```

```
    if(effectRect.x >=5*eframeWidth)
        effectRect.x=0;

}
```

```
if(jump && playerPosition.y<430)
{

    playerPosition.y= oldplayerPosition;
    jump=false;
}
```

```
if(playerPosition.x>1200)
{
    playerPosition.x=-100;

}
```



```
if(playerPosition.y>550)
{
    playerPosition.y=-100;

}
```

```
if(playerPosition.y<-100)
{
    playerPosition.y=550;

}
```

```
if(playerPosition.x<-100)
{
    playerPosition.x=1200;
}
```

```
if(n==1 && i>=5)
{
    i=0;
    playerRect.y=frameHeight;
    effectRect.y=5* eframeHeight;
    sprites=false;
    fire=false;
    finish=true;
    tear=false;
}
```

```
if(n==0 && i>=5)
{
```

```

    i=0;
    playerRect.y=0;
    effectRect.y=5* eframeHeight;
    sprites=false;
    fire=false;
    finish=true;
    tear=false;
}

}

Mix_HaltMusic();

exitstickmanRect.x=playerPosition.x;
exitstickmanRect.y=playerPosition.y;
exitstickmanRect.w=playerPosition.w;
exitstickmanRect.h=playerPosition.h+200;
Mix_PlayChannel( -1, exitjump, 0 );
for (int k=0; backRect.h>-500; k+=1)
{
    SDL_RenderClear(renderTarget);
    SDL_RenderCopy(renderTarget,initial,NULL,NULL);
    SDL_RenderCopy(renderTarget,adios,NULL,&centreRect);
    SDL_RenderCopy(renderTarget,background,&backRect,&backRect);
    SDL_RenderCopy(renderTarget,exitstickman,NULL,&exitstickmanRect);
    SDL_RenderPresent(renderTarget);
    SDL_Delay(50);
    //play(laugh);
    backRect.h-=k;
    exitstickmanRect.y-=k;

```

```
}
```

```
SDL_Delay(1000);
```

```
SDL_DestroyWindow(window);
```

```
SDL_DestroyTexture(currentImage);
```

```
SDL_DestroyTexture(background);
```

```
SDL_DestroyRenderer(renderTarget);
```

```
SDL_DestroyTexture(quitIMG);
```

```
SDL_DestroyTexture(effectImage);
```

```
SDL_DestroyTexture(exitstickman);
```

```
SDL_DestroyTexture(entristickman);
```

```
SDL_DestroyTexture(initial);
```

```
background=NULL;
```

```
exitstickman=NULL;
```

```
entristickman=NULL;
```

```
initial=NULL;
```

```
effectImage=NULL;
```

```
quitIMG=NULL;
```

```
window=NULL;
```

```
currentImage=NULL;
```

```
renderTarget=NULL;
```

```
startbutton=NULL;
```

```
adios=NULL;
```

```
Mix_FreeChunk(walk );
```

```
walk=NULL;
```

```
Mix_FreeChunk( sfire );
```

```
sfire = NULL;
```

```
Mix_FreeChunk( jumpsound );  
jumpsound = NULL;  
Mix_FreeChunk( exitjump );  
exitjump = NULL;  
Mix_FreeChunk( laugh );  
laugh = NULL;  
Mix_FreeChunk( stear);  
stear = NULL;  
Mix_FreeChunk( crack );  
crack = NULL;  
Mix_FreeMusic( backmsc);  
backmsc = NULL;  
Mix_Quit();  
  
return 0;  
}
```