

How to compile and run my program:

1. `qmake-qt5 -project`
2. `qmake-qt5 QT+=widgets & QT+=gui & QT+=multimedia` (已有 pro 檔就不用執行此段)
3. `make` (Do it if you just modified your program. If not, this step is no need.)
4. Open a new terminal -> `startxwin`
5. Back to the original terminal and execute the following code:
 `$ pulseaudio --start -D` // (For opening the music player)
 `$./project2.exe`

Bonus options

1. 殺球 (碰球前或碰球時需按一次空白鍵，球一碰到皮卡丘，速度就會變得比正常速度快 2 倍)
2. 飛撲 (先按空白鍵，再按方向鍵決定飛撲方向，飛撲時皮卡丘的移動速度會比平常快 2 倍)



Notes:

1. 在每次執行 exe 檔前須先執行 `$ pulseaudio --start -D` 否則會沒有音效
2. 欲殺球，就按空白鍵 (提前按也可以)，殺球的速度($40\sqrt{2}$ pixel)為正常球速($20\sqrt{2}$ pixel)的 2 倍
3. 欲飛撲，先按空白鍵，再按方向鍵，飛撲速度(35 pixel)為正常移動速度(20 pixel)的 1.75 倍
4. 切換模式時，按 `backspace` 鍵 (遊戲畫面右上方會有提示) $v_y = v \cdot \frac{4y}{d}$

Describe your function in detail (how to implement specs and bonus)

詳細程式碼內容與如何實現 spec 都在第 4 頁開始做介紹

Images be used

Background + net	
右邊的神奇寶貝 (臭臭泥)	
臭臭泥 飛撲時	
左邊的神奇寶貝 (呆呆獸)	
寶貝球	
獎盃(印在贏家那邊)	

Codes

Mainwindow.h

```
class MainWindow : public QMainWindow {
Q_OBJECT
public:
    MainWindow(QWidget *parent = 0);    //建立視窗
    float g=0.98;                      //重力加速度 (9.8 會太大，故設定為 0.98)
    int keycode=-1;                    //紀錄所按的鍵
    int keycode_first=-1;              //也是記錄所按的鍵，為了連續動作而存在的
    double ball_x=1500,ball_y=0;       //球的 x,y 位置，最一開始從右邊開始
    double ball_vx=0,ball_vy=50;       //球的 x,y 方向速度，最一開始設為往下掉(y>0)
    double ball_vabs=20*sqrt(2);       //球的絕對速度，一開始設為 20 根號 2，只在殺球時會變大
    double l_x=100,l_y=700;           //左邊皮卡丘的 x,y 位置(y=700 即為地板)
    double r_x=1500,r_y=700;          //右邊皮卡丘的 x,y 位置
    double r_vx=0;                    //只有在右邊皮卡丘飛撲時會用到
    int fly;                          //只有在右邊皮卡丘飛撲時會用到，若值為 1，就代表飛撲中
    double l_vy=0;                    //左邊皮卡丘 y 方向的速度
    double r_vy=0;                    //右邊皮卡丘 y 方向的速度
    double t_ball=0;                  //球在空中飛的時間(自由落體公式的時間)
    double t_l=0,t_r=0;               //皮卡丘往上跳得經過時間(自由落體公式的時間)
    bool doubleAI_mode=1;             //決定遊戲模式！若值=0 就是玩家與電腦對打,反之就是兩電腦對打
    double dis_ball_right=900;        //球與左邊皮卡丘的絕對距離
    double dis_ball_left=900;         //球與右邊皮卡丘的絕對距離
    int score_r=0;                    //紀錄右方皮卡丘的分數
    int score_l=0;                    //紀錄左方皮卡丘的分數
    //////////////////////////////////////
    void paintEvent(QPaintEvent *event); //畫畫的函數
    void draw_bg(QPainter &B);          //畫"背景"的函數
    void draw_ball(QPainter &B);        //畫"球"的函數
    void draw_pikachu(QPainter &B);     //畫"兩神奇寶貝"的函數
    void ball_gnd(QPainter &B);         //處理球落地時的函數，亦有畫畫的功能
    void mode(QPainter &B);            //處理遊戲模式的函數
    //////////////////////////////////////
    QImage ball;                      //載入球的變數
    QImage pikachuL;                  //載入左神奇寶貝的變數
    QImage pikachuR;                  //載入右神奇寶貝的變數
    QImage pikachuRfly;               //載入飛撲時的右神奇寶貝的變數
    QImage bg;                        //載入背景與球網的變數
    QImage score;                     //載入贏家畫面的變數
    QSound *pika=new QSound("./dataset/pika.wav"); //載入音樂的變數
};
```

protected:

```
void keyPressEvent(QKeyEvent *event);    //按按鍵時觸發  
void keyReleaseEvent(QKeyEvent *event); //放開按鍵時觸發  
};
```

main.cpp

```
int main(int argc, char *argv[])  
{  
    QApplication a(argc, argv);  
    MainWindow w;           //宣告秀視窗用的 object  
    w.show();               //秀出畫面  
  
    return a.exec();  
}
```

MainWindow.cpp

MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent):

建立視窗並 load 所有需要用到的圖片，我一開始都把 load 寫在下面主要處理畫面變化的程式碼，這樣的寫法會因一直 load 圖片使畫面變得很慢

```
MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent) {
    setFixedSize(1800,900);                //建立一 1800*900 pixel^2 的畫面
    this->setWindowTitle("Pikachu volleyball"); //建立一標題為 Pikachu volleyball 的視窗
    ball.load("./dataset/pokeball.png");    //載入寶貝球
    pikachuL.load("./dataset/daidai.png");  //載入左邊的神奇寶貝(呆呆獸)
    pikachuRfly.load("./dataset/chochonifly.png"); //載入右邊飛撲的神奇寶貝(臭臭泥)
    pikachuR.load("./dataset/chochoni.png"); //載入右邊的神奇寶貝(臭臭泥)
    bg.load("./dataset/bg.png");           //載入背景 (包含了籃網)
    score.load("./dataset/win.png");       //載入獎盃送給贏家
}
```

void MainWindow::mode(QPainter &B)

專門處理遊戲模式的變化

```
void MainWindow::mode(QPainter &B){
    B.drawText(1300,30,QString("Enter 'backspace' to change mode")); //提示按 backspace 可切換模式
    if(keycode==10)
        doubleAI_mode=!doubleAI_mode; //若按了 backspace 則切換畫面
    if(doubleAI_mode)
        B.drawText(1300,55,QString("Current mode:\n2 AI")); //提示玩家目前為雙電腦模式
    else
        B.drawText(1300,55,QString("Current mode:\n1 AI")); //提示玩家目前與電腦對戰模式
}
```

void MainWindow::keyPressEvent(QKeyEvent *event)

負責處理按下指定按鍵時的函數，按了某些按鍵後，會給 **keycode** 指定的整數值

```
void MainWindow::keyPressEvent(QKeyEvent *event) {
    switch (event->key()) {
        case Qt::Key_Up:
            keycode = 0;
            break;
        case Qt::Key_Down:
            keycode = 1;
            break;
        case Qt::Key_Left:
            keycode = 2;
            break;
        case Qt::Key_Right:
            keycode = 3;
            break;
        case Qt::Key_Backspace:    //切換模式時，按 backspace 鍵
            keycode = 10;
            break;
        case Qt::Key_Space:
            keycode = 11;
            break;
        default:
            keycode = -1;
            break;
    }
    update();
}
```

void MainWindow::keyReleaseEvent(QKeyEvent *event)

```
void MainWindow::keyReleaseEvent(QKeyEvent *event){
    switch (event->key()) {
        case Qt::Key_Space:
            keycode = 11;
            break;
        default:
            keycode = -1;
            break;
    }
    keycode_first=keycode;
```

```
}
```

void MainWindow::draw_pikachu(QPainter &B)

負責處理左右皮卡丘的任何行為

```
void MainWindow::draw_pikachu(QPainter &B){
```

```
//For left pikachu x axis
```

處理左邊皮卡丘x方向的移動

```
if(l_x>ball_x-100)
```

```
l_x-=15;
```

```
else
```

```
l_x+=15;
```

讓左邊皮卡丘持續追蹤球

其中 $l_x > ball_x - 100$ 是為了讓皮卡丘盡量偏球的左邊，這樣球撞到皮卡丘才會往右反彈，遊戲較順利進行(移動速度為15pixel)

```
if(l_x<0){
```

```
l_x=1;
```

若皮卡丘跑到視窗左邊，就強制讓皮卡球回來視窗內

```
}
```

```
if (l_x>650) l_x=650-1;
```

若皮卡丘跑到視窗右邊，就同上

```
//For right pikachu on y axis
```

處理右邊皮卡丘y方向的移動

```
if(r_y>690){
```

```
if(keycode==0){
```

```
r_vy=-25;
```

```
t_r=0;
```

```
}
```

```
else{
```

```
r_vy=0;
```

```
r_y=700;
```

```
}
```

```
}
```

按up鍵就跳起來

並開始計滯空時間

滯空時間可決定y方向

速度

當右邊皮卡丘在地面時的行為

若沒按up鍵

若皮卡丘跑到視窗左邊，就強制讓皮卡球回來視窗內

```
else{
```

```
if(fly==1)
```

若為飛撲

```
t_r+=1;
```

時間調快會比較像飛撲

```
else
```

若為普通跳躍

```
t_r+=0.15;
```

就調慢一點比較像正常跳躍

當右邊皮卡丘在空中時

```
r_vy=r_vy+0.5*g*t_r;
```

y方向的物理公式

```
}
```


處理右邊皮卡丘x方向的行為

```
//For right pikachu on x axis
```

```
if((r_x_before_fly-r_x>350||r_x_before_fly-r_x<-350)&&fly){  
    r_vx=0; 這邊設定飛撲可以飛總共350pixel  
    fly=0;   一飛完350pixel就停止(r_vx=0; fly=0);  
}
```

```
if(r_x>1580||r_x<940){  
    fly=0;  
    r_vx=0;  
}
```

皮卡丘有時候會飛過頭 (飛超過球網或超過視窗右邊緣)
此時也要強制停止飛撲

```
if(doubleAI_mode){  
    if(r_x>930&&r_x<=1600){  
        if(ball_x>0){  
            if(r_x>ball_x+50)  
                r_x-=35;  
            else  
                r_x+=35;  
        }  
    }  
    if(r_x<931){  
        r_x=931;  
    }  
    if(r_x>1600)r_x=1600-1;  
}
```

如同左皮卡丘一樣
只是右邊皮卡丘改為盡量
偏右(r_x>ball_x+50)
且移動到視窗外或球網也
要強制歸位
(移動速度為35pixel)

此專為雙電腦模式下
右邊皮卡丘x的行為設
計

專為真人玩家與電腦
對打的模式下，右邊
皮卡丘的行為的設計

```
else{  
    if(r_x>930&&r_x<=1600){  
        if(keycode_first==11&&r_y>680&&(keycode==2||keycode==3)){  
            r_vy=-10;  
            r_x_before_fly=r_x;  
            keycode_first=-1;  
            fly=1;  
            t_r=0;  
            if(keycode==2)  
                r_vx=-35;  
            if(keycode==3)  
                r_vx+=35;  
        }  
        else{  
            if(keycode==2)  
                r_x-=20;  
            if(keycode==3)  
                r_x+=20;  
        }  
    }  
    if(r_x<931){  
        r_x=932;  
    }  
    if(r_x>1600)r_x=1599;  
}
```

此為飛撲時的行為
一開始讓皮卡丘有往上10pixel的速度
並記錄當前位置，將拿來計算前述所說
飛撲350pixel的事件

按左就往左飛撲
按右就往右飛撲
速度皆設定45pixel
較平常移動速度快7/4倍

BONUS!!

```
l_y+=l_vy;  
r_y+=r_vy;  
r_x+=r_vx;  
if(fly==1)
```

將上面辛苦計算完的各個速度參數都加在原本皮卡丘所在的位置

```
    B.drawImage(QRect(r_x,r_y,200,200),pikachuRfly); 若在飛撲就印飛撲的臭  
else 臭泥  
    B.drawImage(QRect(r_x,r_y,200,200),pikachuR); 若非飛撲就印普通  
    B.drawImage(QRect(l_x,l_y,200,200),pikachuL); 臭臭泥  
    印呆呆獸(左方)
```

void MainWindow::draw_ball(QPainter &B)

負責處理球在空中移動的各種行為

```
void MainWindow::draw_ball(QPainter &B){
    B.drawImage(QRect(ball_x,ball_y,150,150),ball); } 無時無刻印寶貝球

//ball collides with window side
if(ball_x>1800-150){
    ball_x=1650;
    ball_vx=-ball_vx;
}
if(ball_x<0){
    ball_x=0;
    ball_vx=-ball_vx;
}
if(ball_y>900-150){
    ball_vy=-ball_vy;
    ball_y=900-150;
}
if(ball_y<0){
    ball_vy=-ball_vy;
}

//ball collides with net from one side
if(ball_vx<0&&ball_x>920-100&&ball_x<920&&ball_y>=565-150&&ball_y<=900-150){
    ball_vx=-ball_vx;
}
if(ball_vx>0&&ball_x>880-150&&ball_x<880-50&&ball_y>=565-150&&ball_y<=900-150){
    ball_vx=-ball_vx;
}

//ball collides with net from top
if(ball_vy>0&&ball_x<=920&&ball_x>=880-150&&ball_y>=540-150&&ball_y<565-150){
    ball_vy=-ball_vy;
    t_ball=0;
}
if(ball_vy>20&&ball_x<=920&&ball_x>=880-150&&ball_y>=530-150&&ball_y<565-150){
    ball_vy=-ball_vy;
    t_ball=0;
}

if(keycode==11) //if wanna Smash!
    ball_vabs=40*sqrt(2);
else
    ball_vabs=20*sqrt(2);

dis_ball_left=sqrt(pow(ball_x+75-l_x-100,2)+pow(ball_y+75-l_y-100,2));
dis_ball_right=sqrt(pow(ball_x+75-r_x-100,2)+pow(ball_y+75-r_y-100,2));
if(dis_ball_left<150){
    ball_vx=ball_vabs*(ball_x+75-l_x-100)/dis_ball_left; //Normalize
    ball_vy=ball_vabs*(ball_y+75-l_y-100)/dis_ball_left;
    t_ball=0;
    keycode=-1;
    pika->play();
}
if(dis_ball_right<150){
    ball_vx=ball_vabs*(ball_x+75-r_x-100)/dis_ball_right; //Normalize
    ball_vy=ball_vabs*(ball_y+75-r_y-100)/dis_ball_right;
    t_ball=0;
    keycode=-1;
    pika->play();
}

t_ball+=0.03;
ball_vy=ball_vy+g*t_ball;
ball_x+=ball_vx;
ball_y+=ball_vy;
}
```

處理球撞到視窗的行為

球撞到左右兩側視窗
x方向就反彈(速度就相反)

球撞到天花板與地板時
y方向就反彈(速度也是相反)

負責球撞到球網兩邊的行為

撞到右側就x方向就反彈

撞到左側x方向亦反彈

負責球撞到球網上端的行為· 每次撞到上端· 飛行時間歸零

condition 1

球撞到上端, y方向就反彈 (普通情況)

有時y方向速度太快, y方向pixel無法進入上面標的condition 1的條件 (球會掉得太下面) · 於是需將太快的球(ball_vy>20)設不同的條件(ball_y>=535-150)

按住空白鍵時· keycode=11 · 代表想殺球· 若想殺球· 球的絕對速度就快兩倍!

BONUS!!!

計算球與皮卡丘的距離 $d = \sqrt{\Delta x^2 + \Delta y^2}$

左右皮卡丘都做一樣的事
依照皮卡丘與球的x,y的相對位置去
計算出x,y方向分到的絕對速度分量· 公式如下:

$$v_x = v \cdot \frac{\Delta x}{d}$$
$$v_y = v \cdot \frac{\Delta y}{d}$$

且每次撞到球後· 球的滯空時間須歸零· 以符合物理現象
並播放出pika的叫聲

每次更新球的滯空時間都加0.03
並帶入自由落體公式

void MainWindow::ball_gnd(QPainter &C)

負責處理球掉到地板後的計分行為

```
void MainWindow::ball_gnd(QPainter &C){
```

```
if(ball_y>900-150){
```

球碰地時(ball_y>750)

```
if(ball_x<880){
```

```
score_r++;
```

```
ball_x=1500,ball_y=0;
```

```
ball_vx=0,ball_vy=0;
```

```
l_x=100,l_y=700;
```

```
r_x=1500,r_y=700;
```

球落在左側地板則右方得分，且下一局寶貝球由右方落球

(ball_x=1500,ball_y=0)

且左右皮卡丘需歸位

```
}
```

```
else{
```

```
score_l++;
```

```
ball_x=150,ball_y=0;
```

```
ball_vx=0,ball_vy=0;
```

```
l_x=100,l_y=700;
```

```
r_x=1500,r_y=700;
```

球落在右側地板則左方得分，且下一局寶貝球由左方落球

(ball_x=100,ball_y=0)

且兩皮卡丘需歸位

```
}
```

```
t_ball=0; } 重新計算滯空時間
```

```
}
```

```
C.drawText(150,100,QString("%1").arg(score_l));
```

```
C.drawText(1600,100,QString("%1").arg(score_r));
```

在畫面顯示雙方分數!

```
if(score_l>=4&&score_r>=4){
```

```
C.drawText(800,100,QString("DEUCE"));
```

```
if(score_r-score_l>1){
```

```
C.drawText(1600,100,QString("%1").arg(score_r));
```

```
C.drawImage(QRect(1350,100,300,300),score);
```

```
ball_x=-200,ball_y=-200;
```

```
g=0;
```

```
}
```

```
if(score_l-score_r>1){
```

```
C.drawText(150,100,QString("%1").arg(score_l));
```

```
C.drawImage(QRect(150,100,300,300),score);
```

```
ball_x=-200,ball_y=-200;
```

```
g=0;
```

```
}
```

```
}
```

雙方分數都4分以上時
進入deuce

Deuce狀態下若右方比左方多2分，則右方獲勝，並在右方印出獎盃
並把球弄到視窗外且重力加速度變為0
表示比賽結束

Deuce狀態下左方比右方多2分則左方獲勝，其他如上

```
else{
```

```
if(score_l==5){
```

```
C.drawText(150,100,QString("%1").arg(score_l));
```

```
C.drawImage(QRect(150,100,300,300),score);
```

```
ball_x=-200,ball_y=-200;
```

```
g=0;
```

```
}
```

```
if(score_r==5){
```

```
C.drawText(1600,100,QString("%1").arg(score_r));
```

```
C.drawImage(QRect(1350,100,300,300),score);
```

```
ball_x=-200,ball_y=-200;
```

```
g=0;
```

```
}
```

```
}
```

若非Deuce情況，只要其中一方達到五分即獲勝
並在獲勝方上空印出獎盃表示勝利

```
}
```

void MainWindow::paintEvent(QPaintEvent *event)

```
void MainWindow::paintEvent(QPaintEvent *event) {
    QPainter B(this);
    QPainter C(this);
    B.setRenderHint(QPainter::Antialiasing, true);
    //////////////////////////////////////
    QFont ft,ft1;
    ft.setPointSize(20);
    ft1.setPointSize(50);
    C.setFont(ft1);
    B.setFont(ft);
    //////////////////////////////////////
    draw_bg(B);
    draw_ball(B);
    draw_pikachu(B);
    ball_gnd(C);
    mode(B);
    this->update();
}
```

宣告畫畫的物件

設定兩畫畫事件的字體大小

各種畫！！！！

持續更新畫面

void MainWindow::draw_bg(QPainter &B)

負責印出背景與籃網

```
void MainWindow::draw_bg(QPainter &B){
    B.drawImage(0,0,bg);
}
```