題目

貪吃蛇 方案B

目標

BN98114

畫面：

setContentView(R.layout.snake\_layout);

顯示畫面

onSaveInstanceState為保存

public void onSaveInstanceState(Bundle outState) {
        //Store the game state
        outState.putBundle(ICICLE\_KEY, mSnakeView.saveState());
    }

暫停時想回到原本畫面

mTileSize為磚塊

mXTileCount為橫軸x

mYTileCount為縱軸y

mXOffset為x偏移位置

mYOffset為y偏移位置

        for (int x = 0; x < mXTileCount; x += 1) {
            for (int y = 0; y < mYTileCount; y += 1) {
                if (mTileGrid[x][y] > 0) {
                    canvas.drawBitmap(mTileArray[mTileGrid[x][y]],
                                mXOffset + x \* mTileSize,
                                mYOffset + y \* mTileSize,
                                mPaint);
                }
            }
        }

畫出外框的磚塊

public void clearTiles() {
        for (int x = 0; x < mXTileCount; x++) {
            for (int y = 0; y < mYTileCount; y++) {
                setTile(0, x, y);
            }
        }
    }

蛇是移動的, clearTiles為清除顯示

控制貪吃蛇方向：

private int mDirection = NORTH;

蛇頭方向預設為向北

private int mNextDirection = NORTH;

下一個蛇頭的方向預設為向北

if (mDirection != SOUTH) {
                mNextDirection = NORTH;
            }
            return (true);
        }

假如蛇頭不面向南,則面向北

if (keyCode == KeyEvent.KEYCODE\_DPAD\_DOWN) {
            if (mDirection != NORTH) {
                mNextDirection = SOUTH;
            }
            return (true);
        }

BN98088

keyCode為鍵盤方向鍵, KEYCODE\_DPAD\_DOWN為當鍵盤按↓,蛇頭不面向北,則面向南

if (keyCode == KeyEvent.KEYCODE\_DPAD\_LEFT) {
            if (mDirection != EAST) {
                mNextDirection = WEST;
            }
            return (true);
        }

KEYCODE\_DPAD\_DOWN為當鍵盤按←,蛇頭不面向東,則面向西

  if (keyCode == KeyEvent.KEYCODE\_DPAD\_RIGHT) {
            if (mDirection != WEST) {
                mNextDirection = EAST;
            }
            return (true);
        }

KEYCODE\_DPAD\_DOWN為當鍵盤按→,蛇頭不面向西,則面向東

private ArrayList<Coordinate> mSnakeTrail = new ArrayList<Coordinate>();

mSnakeTrail.add(new Coordinate(7, 7));
        mSnakeTrail.add(new Coordinate(6, 7));
        mSnakeTrail.add(new Coordinate(5, 7));
        mSnakeTrail.add(new Coordinate(4, 7));
        mSnakeTrail.add(new Coordinate(3, 7));
        mSnakeTrail.add(new Coordinate(2, 7));

mSnakeTrail為蛇身由Coordinate組成

private ArrayList<Coordinate> mAppleList = new ArrayList<Coordinate>();

mAppleList為蘋果存放列表

addRandomApple();
        addRandomApple();

代表隨機添加兩顆蘋果

  public static final int PAUSE = 0;

BN98094

暫停

public static final int READY = 1;

準備

  public static final int RUNNING = 2;

開始

    public static final int LOSE = 3;

結束,死亡

        int applecount = mAppleList.size();
        for (int appleindex = 0; appleindex < applecount; appleindex++) {
            Coordinate c = mAppleList.get(appleindex);
            if (c.equals(newHead)) {
                mAppleList.remove(c);
                addRandomApple();

                mScore++;
                mMoveDelay \*= 0.9;

                growSnake = true;
            }
        }

吃蘋果

int snakelength = mSnakeTrail.size();
        for (int snakeindex = 0; snakeindex < snakelength; snakeindex++) {
            Coordinate c = mSnakeTrail.get(snakeindex);
            if (c.equals(newHead)) {
                setMode(LOSE);
                return;
            }
        }

碰到自己身體,遊戲結束

if ((newHead.x < 1) || (newHead.y < 1) || (newHead.x > mXTileCount - 2)
                || (newHead.y > mYTileCount - 2)) {
            setMode(LOSE);
            return;

        }

碰到牆壁,遊戲結束

參考資料

<http://developer.android.com/resources/samples/Snake/index.html>

分組

組長：張志偉(BN98114)

組員：胡江海(BN98088)、許書翊(BN98094)

工作項目

每個禮拜二下午&線上遠端連線

胡江海→每週要多了解一段程式碼

許書翊→負責找資料、分析大方向

張志偉→分配工作、每次討論做統整

面臨到的問題

張志偉

剩下一個禮拜就要報告了，所以把一些重要的程式碼跟模組開始整理成PPT，讓組員們可以更快了解自己要報告哪些東西好讓他們做準備，但還是對於成是幹嘛在幹嘛還是要找資料做了解。

胡江海

這周開始聽組長開始分析自己分配到的程式碼以及模組下去做分析，不懂的上網找資料，其他一些整理就交給組長。

許書翊

對於開始、暫停、準備這幾的程式碼做了解，只是在於了解之後感覺好像不是說很難，程式碼的樣子好像也式都差不多，只差那一點東西就不一樣了，只是差那一點的程式碼還是要認真去了解，不然還是會看不懂。