Con chim điên phần 4. Thêm trạng thái game và reset lại

# Thêm trạng thái game

Ta sẽ thêm các trạng thái game là RUNNING, GAMEOVER, READY.

1. Mở file GameWorld.java và thêm enum có 3 trạng thái sau đó thêm biến currentState như sau:

**private** GameState currentState;

**public** **enum** GameState {

*READY*, *RUNNING*, *GAMEOVER*

}

1. Trong hàm tạo của GameWorld khởi tạo biến như sau

currentState = GameState.*READY*;

1. Tìm tới hàm update và sửa tên lại thành updateRunning
2. Xây lại hàm update và hàm updateReady như sau:

**public** **void** update(**float** delta) {

**switch** (currentState) {

**case** *READY*:

updateReady(delta);

**break**;

**case** *RUNNING*:

**default**:

updateRunning(delta);

**break**;

}

}

**private** **void** updateReady(**float** delta) {

// chua co gi het

}

1. Trong hàm updateRunning (sửa tên từ hàm update ở trên) ta thêm 1 lệnh vào trong if cuối, lệnh này giúp khi con chim bị va chạm sẽ sửa trạng thái thành lên dĩa

currentState = GameState.*GAMEOVER*;

Sau khi sử xong thì đoạn if nó sẽ như sau:

**if** (Intersector.*overlaps*(bird.getBoundingCircle(), ground)) {

scroller.stop();

bird.die();

bird.decelerate();

currentState = GameState.*GAMEOVER*;

}

1. Xây tiếp một biến toàn cục

**public** **int** midPointY;

1. Khởi tạo nó trong hàm tạo

**this**.midPointY = midPointY;

1. Xây tiếp 2 hàm (còn lỗi). Hàm restart sẽ giúp gọi lần lược để restart lại bird, scroller

**public** **void** restart() {

currentState = GameState.*READY*;

score = 0;

bird.onRestart(midPointY - 5);

scroller.onRestart();

currentState = GameState.*READY*;

}

**public** **boolean** isGameOver() {

**return** currentState == GameState.*GAMEOVER*;

}

# Restart lại chim

1. Mở file Bird.java và thêm hàm để restart lại chim

**public** **void** onRestart(**int** y) {

rotation = 0;

position.y = y;

velocity.x = 0;

velocity.y = 0;

acceleration.x = 0;

acceleration.y = 460;

isAlive = **true**;

}

# Restart lại Scrollhandler, cỏ và đường ống

1. Mở file ScrollHandler.java và thêm hàm restart (còn lỗi)

**public** **void** onRestart() {

frontGrass.onRestart(0, *SCROLL\_SPEED*);

backGrass.onRestart(frontGrass.getTailX(), *SCROLL\_SPEED*);

pipe1.onRestart(210, *SCROLL\_SPEED*);

pipe2.onRestart(pipe1.getTailX() + *PIPE\_GAP*, *SCROLL\_SPEED*);

pipe3.onRestart(pipe2.getTailX() + *PIPE\_GAP*, *SCROLL\_SPEED*);

}

1. Mở file Grass.java thêm hàm

**public** **void** onRestart(**float** x, **float** scrollSpeed) {

position.x = x;

velocity.x = scrollSpeed;

}

1. Mở file Pipe.java thêm hàm

**public** **void** onRestart(**float** x, **float** scrollSpeed) {

velocity.x = scrollSpeed;

reset(x);

}

1. Save all và thấy là đã đầy đủ các hàm restart rồi nên không còn báo lỗi.

# Điều khiển InputHandler để khởi động lại game khi die

1. Mở file InputHandler.java lên. Class này cũng cần truy xuất đến GameWorld nên ta truyền GameWorld vào cho nó, và từ GameWorld có thể truy ra chim nên không cần truyền chim nữa, khai báo thêm biến thuộc lớp GameWorld và sửa hàm tạo lại như sau

**private** Bird myBird;

GameWorld myWorld;

**public** InputHandler(GameWorld myWorld)

{

**this**.myWorld=myWorld;

**this**.myBird=myWorld.getBird();

}

1. Tìm hàm touchdown và sửa lại như sau

@Override

**public** **boolean** touchDown(**int** screenX, **int** screenY, **int** pointer, **int** button) {

**if** (myWorld.isReady()) {

myWorld.start();

}

myBird.onClick();

**if** (myWorld.isGameOver()) {

// Reset all variables, go to GameState.READ

myWorld.restart();

}

**return** **true**;

}

1. Báo lỗi chổ .isReady() và .start() thì mở lại file GameWorld.java và thêm 2 hàm

**public** **boolean** isReady() {

**return** currentState == GameState.*READY*;

}

**public** **void** start() {

currentState = GameState.*RUNNING*;

}

# Sửa GameScreen

1. Do đổi hàm tạo của InputHandler ta cần phải đổi GameScreen . Mở file GameScreen.java lên thay dòng

Gdx.*input*.setInputProcessor(**new** InputHandler(world.getBird()));

Thành

Gdx.*input*.setInputProcessor(**new** InputHandler(world));

# Đổi GameRender để hiển thị thông tin

1. Mở file GameRender.java tìm đến hàm render bỏ khối

// chuyen integer thanh String

String score = myWorld.getScore() + "";

// ve shadow

AssetLoader.*shadow*.draw(batcher, "" + myWorld.getScore(), (136 / 2)

- (3 \* score.length()), 12);

// ve text

AssetLoader.*font*.draw(batcher, "" + myWorld.getScore(), (136 / 2)

- (3 \* score.length() - 1), 11);

Và thay thế bằng khối mới như sau:

**if** (myWorld.isReady()) {

// Draw shadow first

AssetLoader.*shadow*.draw(batcher, "Touch me", (136 / 2)

- (42), 76);

// Draw text

AssetLoader.*font*.draw(batcher, "Touch me", (136 / 2)

- (42 - 1), 75);

} **else** {

**if** (myWorld.isGameOver()) {

AssetLoader.*shadow*.draw(batcher, "Game Over", 25, 56);

AssetLoader.*font*.draw(batcher, "Game Over", 24, 55);

AssetLoader.*shadow*.draw(batcher, "Try again?", 23, 76);

AssetLoader.*font*.draw(batcher, "Try again?", 24, 75);

}

// chuyen integer thanh String

String score = myWorld.getScore() + "";

// ve shadow

AssetLoader.*shadow*.draw(batcher, "" + myWorld.getScore(), (136 / 2)

- (3 \* score.length()), 12);

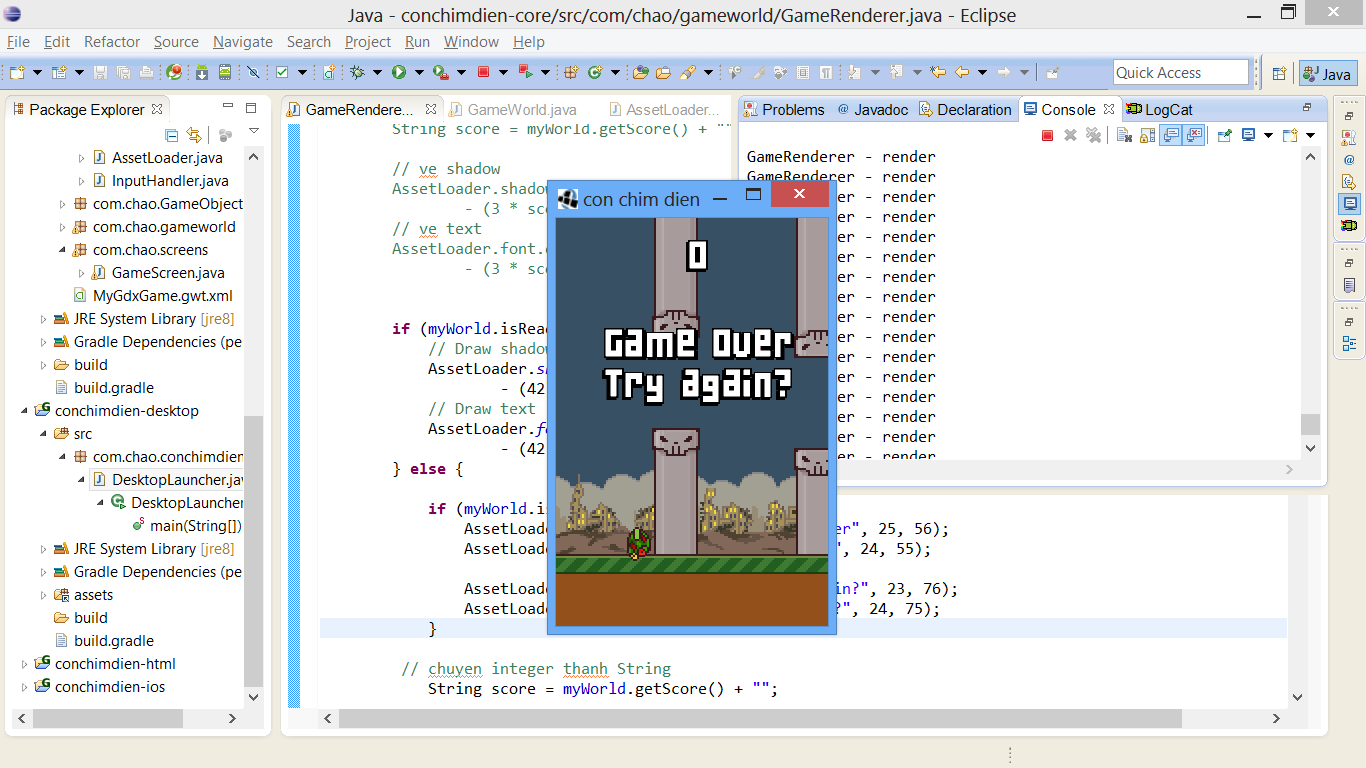
// ve text

AssetLoader.*font*.draw(batcher, "" + myWorld.getScore(), (136 / 2)

- (3 \* score.length() - 1), 11);

}

1. Chạy chương trình và thấy thông báo khi bắt đầu và khi die



# Lưu điểm số cao nhất

Dùng Preferences để lưu lại điểm số cao nhất.

1. Mở file AccessLoader.java khai báo biến toàn cục

**public** **static** Preferences *prefs*;

1. Nhúng thư viện

**import** com.badlogic.gdx.Preferences;

1. Tìm đến hàm load thêm như sau:

// tao hoac truy xuat bien preferences

*prefs* = Gdx.*app*.getPreferences("ZombieBird");

// hightScore mac dinh la 0

**if** (!*prefs*.contains("highScore")) {

*prefs*.putInteger("highScore", 0);

}

1. Thêm 2 hàm để lấy và gán

// gán highScore trong prefs

**public** **static** **void** setHighScore(**int** val) {

*prefs*.putInteger("highScore", val);

*prefs*.flush();

}

// lay high score

**public** **static** **int** getHighScore() {

**return** *prefs*.getInteger("highScore");

}

1. Quay lại GameWorld.java thêm HIGHSCORE và enum

**public** **enum** GameState {

*READY*, *RUNNING*, *GAMEOVER*, *HIGHSCORE*

}

1. Trong hàm updateRunning tìm đến hàm if xét va chạm và thêm vào một if con để lưu HightScore, toàn bộ lệnh if sau khi sửa như sau:

**if** (Intersector.*overlaps*(bird.getBoundingCircle(), ground)) {

scroller.stop();

bird.die();

bird.decelerate();

currentState = GameState.*GAMEOVER*;

**if** (score > AssetLoader.*getHighScore*()) {

AssetLoader.*setHighScore*(score);

currentState = GameState.*HIGHSCORE*;

}

}

1. Thêm một hàm để xét xem đây có phải là highscore hay không

**public** **boolean** isHighScore() {

**return** currentState == GameState.*HIGHSCORE*;

}

1. Mở file GameRender.java, trong hàm render bỏ đoạn

**if** (myWorld.isGameOver()) {

AssetLoader.*shadow*.draw(batcher, "Game Over", 25, 56);

AssetLoader.*font*.draw(batcher, "Game Over", 24, 55);

AssetLoader.*shadow*.draw(batcher, "Try again?", 23, 76);

AssetLoader.*font*.draw(batcher, "Try again?", 24, 75);

}

Và thay bằng đoạn

**if** (myWorld.isGameOver() || myWorld.isHighScore()) {

**if** (myWorld.isGameOver()) {

AssetLoader.*shadow*.draw(batcher, "Game Over", 25, 56);

AssetLoader.*font*.draw(batcher, "Game Over", 24, 55);

AssetLoader.*shadow*.draw(batcher, "High Score:", 23, 106);

AssetLoader.*font*.draw(batcher, "High Score:", 22, 105);

String highScore = AssetLoader.*getHighScore*() + "";

// Draw shadow first

AssetLoader.*shadow*.draw(batcher, highScore, (136 / 2)

- (3 \* highScore.length()), 128);

// Draw text

AssetLoader.*font*.draw(batcher, highScore, (136 / 2)

- (3 \* highScore.length() - 1), 127);

} **else** {

AssetLoader.*shadow*.draw(batcher, "High Score!", 19, 56);

AssetLoader.*font*.draw(batcher, "High Score!", 18, 55);

}

AssetLoader.*shadow*.draw(batcher, "Try again?", 23, 76);

AssetLoader.*font*.draw(batcher, "Try again?", 24, 75);

// Convert integer into String

String score = myWorld.getScore() + "";

// Draw shadow first

AssetLoader.*shadow*.draw(batcher, score,

(136 / 2) - (3 \* score.length()), 12);

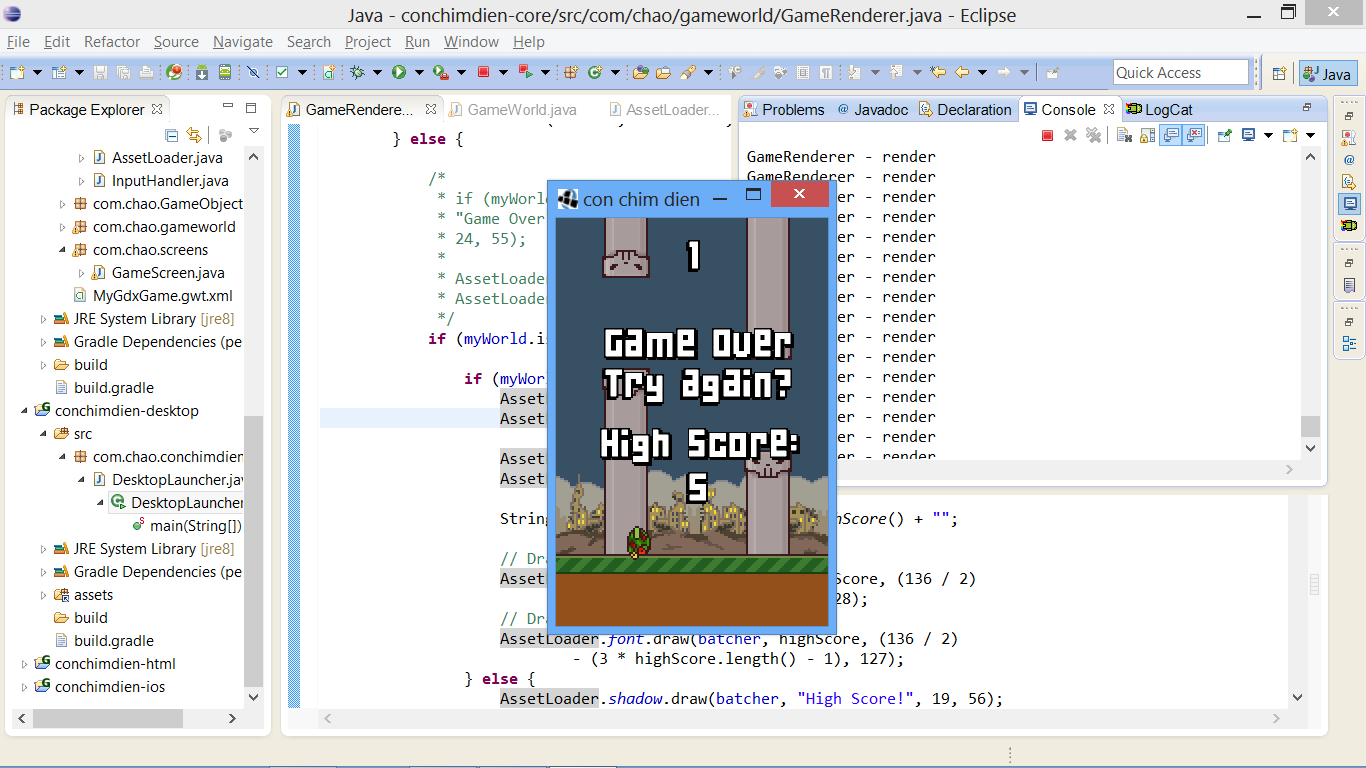
// Draw text

AssetLoader.*font*.draw(batcher, score,

(136 / 2) - (3 \* score.length() - 1), 11);

}

1. Chạy chương trình và xem kết quả



1. Sửa lại một chút, mở file GameWorld.java sửa lại hàm update như sau:

**public** **void** update(**float** delta) {

**switch** (currentState) {

**case** *READY*:

updateReady(delta);

**break**;

**case** *RUNNING*:

updateRunning(delta);

**break**;

**default**:

**break**;

}

}

1. Cuối cùng mở InputHandler.java lên và trong hàm onTouchDown thêm phần || myWorld.isHightScore() vào thêm cho phần if myworld.isGameOver()

**if** (myWorld.isGameOver() || myWorld.isHighScore() ) {

# Chạy trên Android

1. Chọn project android, mở file Manifest.xml lên sửa activity

android:screenOrientation=*"portrait"*

1. Click phải project android chọn Run as -> Android Application để chạy project (chú ý: nên chạy trên máy thật vì máy ảo coi chừng không hỗ trợ OpenGL)

