DOCUMENTATION

MEGAMAN.JAVA

Là lớp xây dựng nên nhân vật Megaman, lớp sẽ chứa các phương thức, thuộc tính của nhân vật

1. Import

1.1 Built-In

1.1.1 AudioClip

Lớp AudioClip là lớp sử dụng để phát âm thanh bao gồm các phương thức như play, loop, stop,...

1.1.2 Graphics2D

Lớp Graphics2D là lớp mở rộng của lớp Graphics, cung cấp khả năng kiểm soát phức tạp hơn đối với hình học, chuyển đổi tọa độ, quản lý màu và bố cục văn bản.

1.1.3 Rectangle

Lớp Rectangle xác định một khu vực trong không gian tọa độ được xác định bởi điểm trên cùng bên trái của hình chữ nhật (x, y) trong không gian tọa độ, chiều rộng và chiều cao của nó

1.2 Sefl-Built

GameWorldState

Animation

CacheDataLoader

Những class trên sẽ được giải thích sau

2. Constructors

2.1 Megaman(float x, float y, GameWorldState gameWorld)

Với các tham số truyền vào là x, y, gameWorld trong đó x, y là toạ độ của nhân vật Megaman, và gameWorld là trạng thái game truyền vào, lớp GameWorldState sẽ được định nghĩa rõ ở file GameWorldState.java. Ngoài ra lớp Megaman còn kế thừa các thuộc tính width, height, mass, blood từ lớp cha là Human.java, đó là các thông tin, chỉ số cơ bản của một nhân vật trong game.



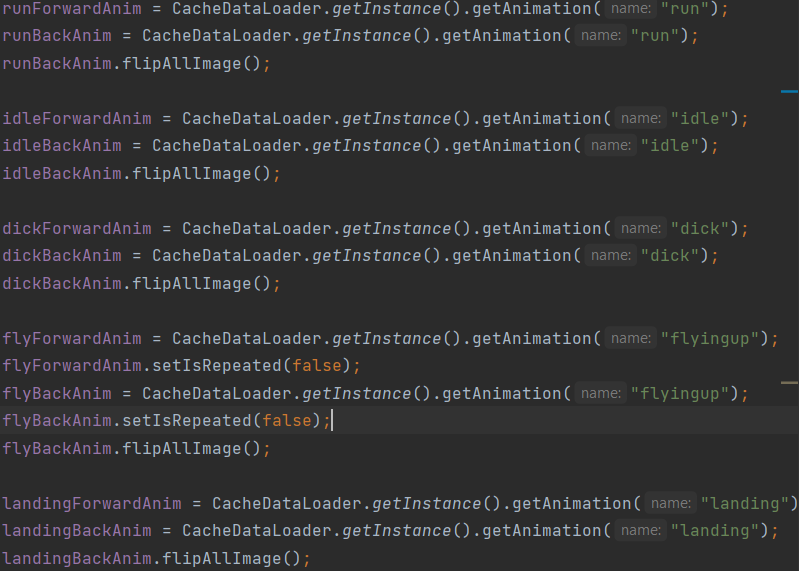
Hai dòng trên là lệnh gán sử dụng phương thức getSound tức lấy các file âm thanh từ folder data

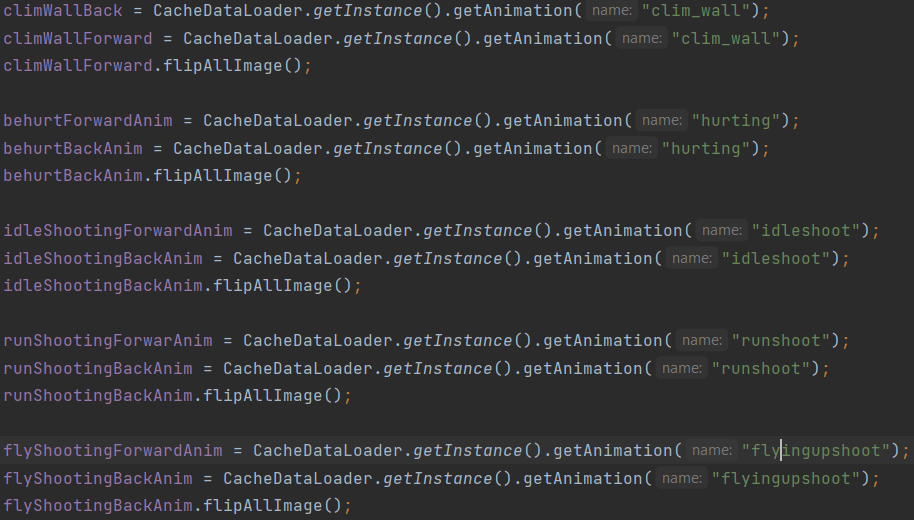


Set loại đối tượng cho nhân vật Megaman, ở đâu là đối tượng ở phe đồng minh, khi đó sẽ bị các đối tượng ở phe địch (ENEMY\_TEAM) gây sát thương



Set thời gian không chịu ảnh hưởng của sát thương sau khi chịu sát thương



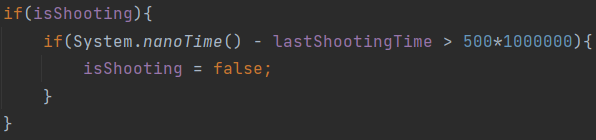


Nạp các Animation vào các biến đã tạo sẵn. Phương thức setIsRepeated() sẽ làm cho các animation lặp lại nếu tham số truyền vào là True. Phương thức filpAllImage() đơn giản là lật các ảnh của animation để tạo nên một animation mới nhưng là phía ngược lại.

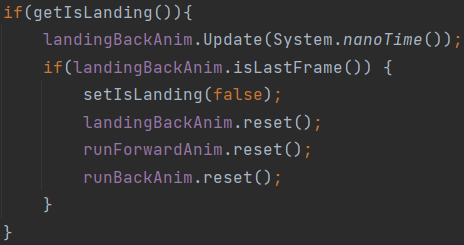
3. Methods

3.1 Update

Được kế thừa từ hàm Update trong Human.java, hàm này có thêm hai câu lệnh điều kiện khác.



Thứ nhất là câu lệnh điều kiện giúp kiểm tra xem thời gian lần bắn lần nhất đến thời điểm hiện tại đã lớn hơn 500.000.000 nano giây chưa, nếu rồi thì mới set trạng thái đang bắn cho nhân vật là false, nếu chưa thì trong khoảng thời gian này nhân vật sẽ không được bắn thêm. Nói cho dễ hiểu thì khoảng cách giữa các lần bắn liên tiếp phải lớn hơn 500.000.000 nano giây.



Thứ hai là là lệnh kiểm tra xem nhân vật đã chạm đất chưa. Ngay khi vào game ta có thể thấy nhân vật rơi tự do từ vị trí giữa màn hình, và khi chạm đất tức là frame cuối của animation tiếp đất ta cần phải reset cho nhân vật về trạng thái đứng yên và có thể chuyển động qua lại

3.2 getBoundForCollisionWithEnemy()

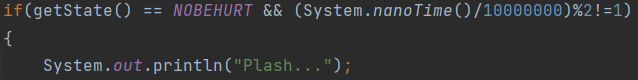


Hàm này chỉ đơn giản là tạo một đối tượng Rectangle là vùng biên xung quanh nhân vật khi nhân vật đứng, chạy, ngồi,… và trả về đối tượng đó

3.3 draw()

Là phương thức để vẽ lên nhân vật dựa vào các trạng thái nhân vật nhận được

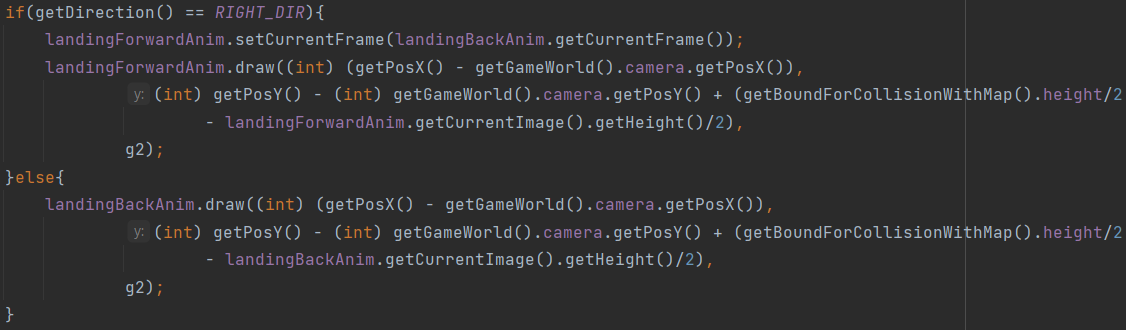
Đối với hai trạng thái ALIVE và NOBEHURT 



In ra console nếu nhân vật đang trong trạng thái bị thương sau khi dính sát thương



Còn nếu nhân vật đang tiếp đất, nhân vật không thể thao tác gì thêm, và ta chỉ update các animation của nhân vật theo hướng di chuyển





Nếu nhân vật đang nhảy, nhân vật vẫn có thể thực hiện bắn và thay đổi hướng. Ta phải update lại animtion nhảy của mỗi hướng khi thao tác chuyển hướng, và set frame nhảy và bắn khi thao tác bắn





Còn nếu nhân vật đang ngồi thì không thể bắn được, chỉ có thể đổi hướng, tương tự như khi tiếp đất



Ngoài các trường hợp trên thì chúng ta còn lại các animation đứng im và chạy, hai hành động này đều có thể kết hợp với bắn

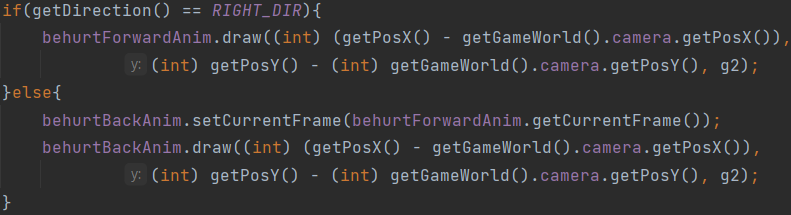


Với chạy thì ta dựa vào SpeedX > 0 hay < 0 để vẽ các animation

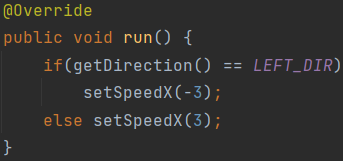


Còn với đứng im thì ta dựa vào hướng qua getDirection()



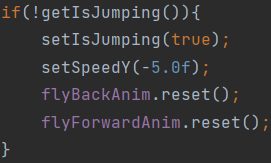
Đối với case BEHURT, chúng ta cũng vẽ animation theo hướng bị thương của nhân vật 

3.4 run()



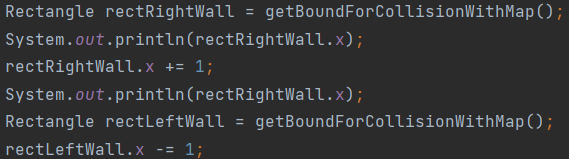
Nếu hướng chạy là bên trái thì set SpeedX = - 3, tức là getSpeedX() < 0, thì nhân vật và camera sẽ di chuyển về bên trái, chúng ta đã làm rõ ở hàm draw() phía trên. Và với SpeedX = 3 thì ngược lại.

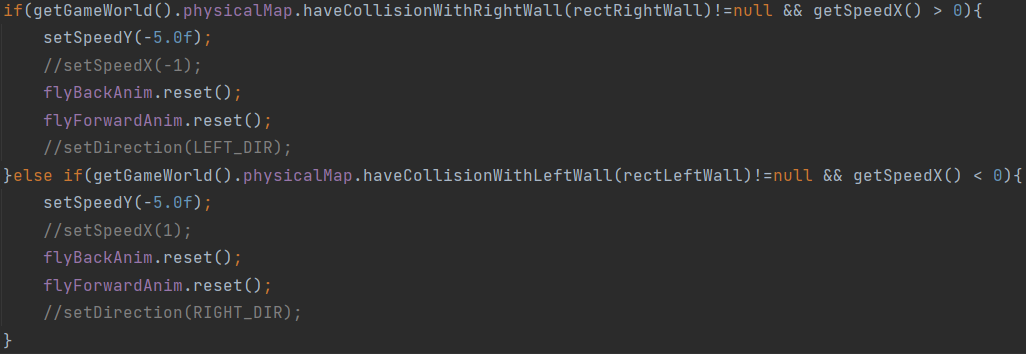
3.5 jump()



Nếu như đang không nhảy tức là khác getIsJumping thì chúng ta mới có thể nhảy (setIsJumping(true)) để tránh việc ta có thể nhảy nhiều lần. Sau đó chúng ta set SpeedY cho nhân vật, -5.0f là do trục y của game bị ngược. Phần reset của của hai animation giúp mỗi lần nhảy xong thì lần nhảy tiếp theo animation nhảy sẽ lại bắt đầu từ frame 0.

Còn một trường hợp khác của nhảy đó là leo trèo, nó chỉ xảy ra khi nhân vật sát với tường cao, ta sẽ tạo hai đối tượng hình chữ nhật bao quanh nhân vật, một cái để xét va chạm với tường bên phải và một cái để xét va chạm với tường bên trái

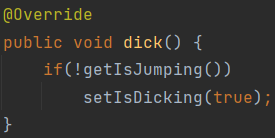




Hàm haveCollisionWithLeftWall() là hàm kiểm tra va chạm với tường bên trái, hàm sẽ trả về 1 đối tượng Rectangle hoặc null, nêu khác null tức là có va chạm với tường bên phải và đồng thời SpeedX > 0 tức là nhân vật vẫn đang di chuyển về phía trước thì lúc này nhân vật mới có thể trèo. Ta set SpeedY cho nhân vật là chỉ số âm và reset lại các animation

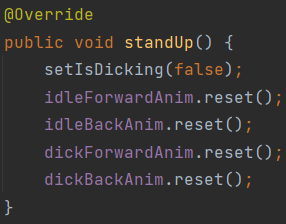
Hàm haveCollisionWithRightWall() cũng tương tự nhưng là đối với tường bên phải và SpeedX phải < 0 tức là đang di chuyển về bên trái

3.6 dick()



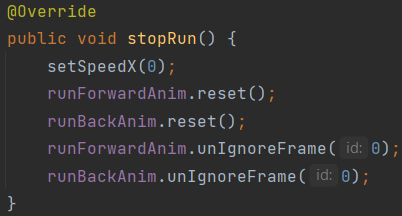
Nếu như đang không nhảy thì chúng ta có thể ngồi

3.7 standUp()



Muốn đứng lên ta phải huỷ trạng thái ngồi, lúc đó phải setIsDicking(false), và phải reset lại các animation về frame 0

3.8 stopRun()



Khi ta nhả các phím di chuyển về hai bên thì ta phải gọi đến hàm stopRun() này, lúc đó SpeedX về lại mốc 0 và reset lại các animation

3.9 attack()



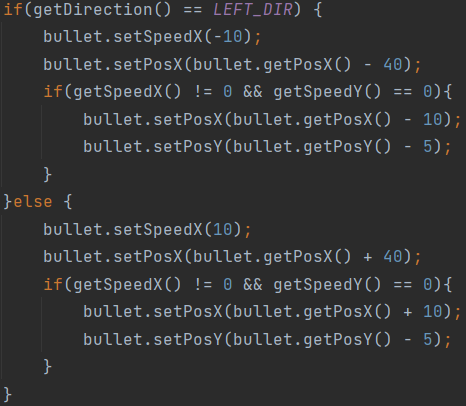
Tấn công là một hành động cần có thời gian chờ và không thể sử dụng khi nhân vật đang ngồi, vì vậy khi cả hai điều này không xảy ra ta mới có thể draw nhân vật.



Âm thanh của đạn bắn được phát



Một biến bullet được tạo dựa trên lớp Bullet và kế thừa lớp BlueFire. Nó có các chỉ số như toạ độ x, y và thế giới chứa đối tượng



Hướng đi của nhân vật chỉ có hai hướng nên hướng đạn cũng vậy, ta sẽ theo hướng của nhân vật mà vẽ. Việc cộng, trừ 10 pixel giúp cho vị trí viên đạn từ tâm nhân vật ra đến bàn tay. Ngoài ra nếu nhân vật đang chạy trên một đường thẳng, ta sẽ phải set toạ độ cho đạn dựa theo animation chạy.



Khi đang nhảy cũng vậy, ta phải trừ PosY đi 20 pixel để trùng với vị trí bàn tay nhân vật

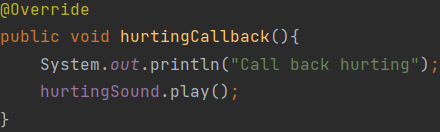


Phía trên chúng ta đã nhắc đến việc setTeamType cho nhân vật, và những thứ thuộc nhân vật cũng sẽ thuộc team đó. Việc setTeamType cho viên đạn sẽ làm cho nhân vật không bị thương và team ENEMY sẽ bị chịu sát thương bởi viên đạn của nhân vật. Sau khi đã cấu hình hết cho viên đạn, viên đạn sẽ được thêm vào lớp GameWorldState chứa list bulletManager để quản lý



Vừa khi viên đạn được bắn ra, biến lastShootingTime sẽ lưu lại thời gian và trạng thái đang bắn set là true

3.10 hurtingCallback()



Hàm sẽ phát âm thanh bị đau khi nhân vật trúng đạn

4. Variables

4.1 Animation runForwardAnim, runBackAnim, runShootingForwarAnim, runShootingBackAnim, idleForwardAnim, idleBackAnim, idleShootingForwardAnim, idleShootingBackAnim, dickForwardAnim, dickBackAnim, flyForwardAnim, flyBackAnim, flyShootingForwardAnim, flyShootingBackAnim, landingForwardAnim, landingBackAnim, climWallForward, climWallBack

Đây là các biến thuộc lớp Animation dùng để biểu diễn cho các animation hành động của nhân vật (bao gồm chạy, đứng im, nhảy, tiếp đất, leo trèo, …)

4.2 long lastShootingTime

Biến trả về giá trị thời gian lần bắn cuối cùng của nhân vật

4.3 boolean isShooting

Biến trả về giá trị đúng sai kiểm tra xem nhân vật có đang bắn đạn không

4.4 AudioClip hurtingSound, shooting1

Hai biến để gán âm thanh (âm thanh khi nhân vật bị thương, và âm thanh khi viên đạn của nhân vật được bắn ra)