DOCUMENTATION

PHYSICALMAP.JAVA

Lớp này được xây dựng để xử lý các vấn đề liên quan đến tương tác vật lý giữa các đối tượng trong game với map

1. Import

1.1 Built-In

1.1.1 Color (java.awt.Color)

Một lớp để đóng gói các màu trong không gian màu sRGB mặc định hoặc các màu trong không gian màu tùy ý được xác định bởi một ColorSpace.

1.1.2 Graphhics2D

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 Selt-Built

1.2.1 GameWorldState

1.2.2 CacheDataLoader

Những class trên sẽ được giải thích tại một file Word khác

2. Constructors

2.1 PhysicalMap

Với 3 tham số truyền vào là x, y, gameWorld trong đó x, y là toạ độ góc trái trên của hình chữ nhật Physicalmap .

3. Methods

Các hàm 3.1 đến 3.4 là các hàm kiểm tra va chạm, tham số được truyền vào là một hình chữ nhật và ta sẽ kiểm tra xem hình chữ nhật được truyền vào đó có va chạm với một hình khác không.

3.1 haveCollisionWithTop(Rectangle rect)



Biến posX1 là index theo cột của vật thể rect trong physicalmap cũng tức là index cột của các điểm 0 hoặc 1 trong file phys\_map.txt. Ở đây posX1 được tính bằng cách lấy giá trị toạ độ x góc trái trên của hình chữ nhật rect và chia cho kích thước của một ô vuông tạo nên physicalmap tức là tileSize. Sau đó ta trừ đi 2 index gọi là sai số,  để khoanh vùng chính xác hơn. Tương tự với posX1 thì posX2 ta chỉ cần cộng thêm với độ rộng của hình chữ nhật rect rồi chia cho tileSize, sau đó cộng thêm 2 index để tăng kích thước vùng kiểm tra.



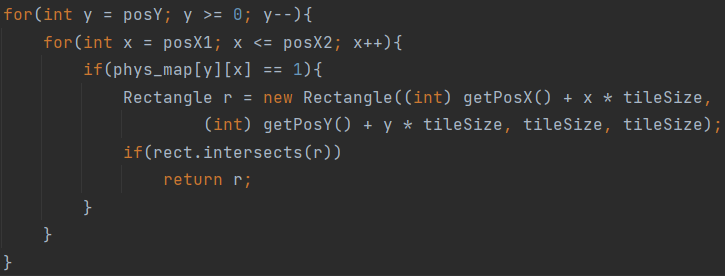


Biến posY là index theo hàng của vật thể rect trong physicalmap cũng tức là index hàng của các điểm 0 hoặc 1 trong file phys\_map.txt. PosY được tính bằng cách lấy toạ độ y góc trái trên của hình chữ nhật rect và chia cho tileSize. Do là chiều cao của game khá ngắn so với chiều dài nên ta không cần mở rộng phạm vi cần kiểm tra đối với posY.





PosX1 < 0 tức là vị trí góc trên bên phải của rect vượt qua biên trái của map, lúc này ta phải set cho posX1 = 0 để rect không bị tràn ra ngoài map. Với vị trí góc trên bên trái của rect cũng vậy, nếu lớn hơn chiều rộng của map thì set nhỏ hơn so với độ rộng của map bằng cách trừ đi 1 index do độ rộng của map lớn hơn 1 đv so với index phần tử cuối của map.



Đây là vòng lặp giúp ta kiểm tra va chạm với vòng for bên ngoài thì lặp theo hàng từ dưới lên trên so với map, vòng lặp trong thì lặp theo cột từ vị trí index posX1 đến posX2. Sau đó ta sẽ kiểm tra xem phys\_map[x][y] tức là tại vị trí hàng y cột x có giá trị bằng 1 hay không , nếu bằng 1 tức là vật cản ta mới kiểm tra. Tiếp đến ta tạo mới một đối tượng Rectangle với toạ độ x, y trong các khoảng đã được xác định sao cho quét hết các ô trong phạm vi cần kiểm tra, khi nào đối tượng r và đối tượng rect được phát hiện va chạm bởi phương thức intersects thuộc lớp Rectangle thì sẽ trả về r, ngược lại thì trả về null. Việc xác định r giúp ta sau này làm việc với va chạm giữa các đối tượng và map dễ dàng hơn.

3.2 haveCollisionWithLand(Rectangle rect)

Tương tự với hàm haveCollisionWithTop(Rectangle rect), nhưng chỉ khác là posY ở đây là toạ độ hàng của góc dưới bên trái cũng chính là toạ độ của cạnh dưới của rect và ta sẽ chạy vòng lặp đối với y tăng dần do trục y ngược và ta đang muốn kiếm tra theo hàng từ trên xuống

3.3 haveCollisionWithRightWall(Rectangle rect)

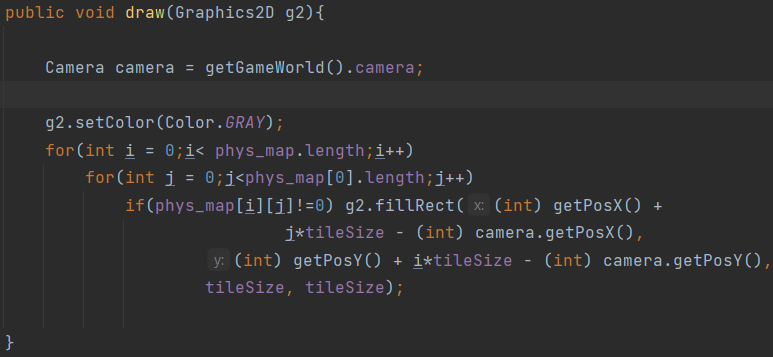
Đối với hai hàm trên thì hàm này và hàm dưới sẽ có chút khác biệt do có hai biến posY1 và posY2 do chúng ta đang xét va chạm với tường bên phải và đương nhiên sẽ có trường hợp va chạm với 1 điểm giữa cạnh trên và dưới

3.4 haveCollisionWithLeftWall(Rectangle rect)

Tương tự với hàm trên, chỉ khác về thứ tự lặp

3.5 draw(Graphics2D g2)

Hàm này sử dụng lớp Camera lấy từ Gameworld, mục đích là lấy toạ độ hiện tại của Camera để khi di chuyển nhân vật, những ô vuông tạo nên map cũng sẽ được thay đổi theo vị trí camera



4. Variables

4.1 phys\_map



Là một mảng hai chiều chứa các phần tử thứ nhât là numberOfRows, phần tử thứ hai là numberOfColumns. Đại diện cho phần tử phys\_map[i][j] được biểu diễn dưới dạng 0 hoặc 1 (0 là đối tượng qua vị trí này sẽ không bị cản, và ngược lại)

4.2 tileSize





Là kích thước của một ô vuông vô hình, nhiều ô vuông có cùng kích cỡ tạo nên map tương tác vật lý