

Class TelaBoss.GameKeyAdapter

java.lang.Object
java.awt.event.KeyAdapter
TelaBoss.GameKeyAdapter

All Implemented Interfaces:

KeyListener, EventListener

Enclosing class:

TelaBoss

```
public class TelaBoss.GameKeyAdapter
extends KeyAdapter
```

Classe para o leitor te teclas responsável por controlar o jogador

Version:

1.0

Author:

Arthur dos Santos Rezende

Field Summary

Fields		
Modifier and Type	Field	Description
private final Set<Integer>	activeMovementKeys	Set guardando as teclas de movimento (teclas direcionais) ativas
private final Set<Integer>	activeShootingKeys	Set guardando as teclas de tiro (teclas A,W,S e D) ativas
private Timer	continuousActionTimer	Timer para processar ações do jogador de forma contínua (~60 FPS)
private Direction	ultimaDirecaoMovimento	Última direção de movimento do jogador
private Direction	ultimaDirecaoTiro	Última direção de tiro do jogador
private long	ultimoTiro	Guarda a última vez que o jogador atirou

Constructor Summary

Constructors	
Constructor	Description
GameKeyAdapter()	Construtor da classe GameKeyAdapter

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	dispose()	Para o timer de ação contínua
private boolean	isMovementKey (int code)	Verifica se a tecla apertada é uma tecla direcional
private boolean	isShootingKey (int code)	Verifica se a tecla apertada é A,W,S ou D
void	keyPressed (KeyEvent e)	Chamado quando alguma tecla é pressionada.
void	keyReleased (KeyEvent e)	Chamado quando alguma tecla é solta.
private void	processMovement ()	Processa o movimento do jogador
private void	processShooting ()	Processa os tiros do jogador
private void	updateAnimation ()	Atualiza a imagem atual do jogador baseado no que ele está fazendo no momento
Methods inherited from class java.awt.event.KeyAdapter		
keyTyped		
Methods inherited from class java.lang.Object		
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait		

Field Details
activeMovementKeys
<pre>private final Set<Integer> activeMovementKeys</pre> <p>Set guardando as teclas de movimento (teclas direcionais) ativas</p>
activeShootingKeys
<pre>private final Set<Integer> activeShootingKeys</pre> <p>Set guardando as teclas de tiro (teclas A,W,S e D) ativas</p>
ultimaDirecaoMovimento
<pre>private Direction ultimaDirecaoMovimento</pre> <p>Última direção de movimento do jogador</p>
ultimaDirecaoTiro
<pre>private Direction ultimaDirecaoTiro</pre>

Última direção de tiro do jogador

ultimoTiro

```
private long ultimoTiro
```

Guarda a última vez que o jogador atirou

continuousActionTimer

```
private Timer continuousActionTimer
```

Timer para processar ações do jogador de forma contínua (~60 FPS)

Constructor Details

GameKeyAdapter

```
public GameKeyAdapter()
```

Construtor da classe GameKeyAdapter

Method Details

keyPressed

```
public void keyPressed(KeyEvent e)
```

Chamado quando alguma tecla é pressionada.

Specified by:

[keyPressed](#) in interface [KeyListener](#)

Overrides:

[keyPressed](#) in class [KeyAdapter](#)

Parameters:

e - O evento a ser processado

keyReleased

```
public void keyReleased(KeyEvent e)
```

Chamado quando alguma tecla é solta.

Specified by:

[keyReleased](#) in interface [KeyListener](#)

Overrides:

[keyReleased](#) in class [KeyAdapter](#)

Parameters:

e - O evento a ser processado

isMovementKey

```
private boolean isMovementKey(int code)
```

Verifica se a tecla apertada é uma tecla direcional

Parameters:

code - o código da tela apertada

Returns:

true caso seja uma tecla direcional. Caso contrário, false

isShootingKey

```
private boolean isShootingKey(int code)
```

Verifica se a tecla apertada é A,W,S ou D

Parameters:

code - o código da tela apertada

Returns:

true caso seja A,W,S ou D. Caso contrário, false

processMovement

```
private void processMovement()
```

Processa o movimento do jogador

processShooting

```
private void processShooting()
```

Processa os tiros do jogador

updateAnimation

```
private void updateAnimation()
```

Atualiza a imagem atual do jogador baseado no que ele está fazendo no momento

dispose

```
public void dispose()
```

Para o timer de ação contínua