Atividade Framework 03

Nome: Nicolas Luís Crusco Rocha de Moraes

Objetivo

Comente os recursos disponíveis no LibGDX (procure no Google ou no Duckdukgo os métodos que foram usados) e as técnicas de programação utilizadas para se obter as ações que foram abordados na segunda parte do jogo de exemplo “Gota”.

Código

set

```
public Vector3 set(float x,
                   float y,
                   float z)
```

Sets the vector to the given components

Parameters:

`x` - The x-component

`y` - The y-component

`z` - The z-component

Returns:

`this vector for chaining`

isTouched

```
boolean isTouched()
```

Returns:

`whether the screen is currently touched.`

getX

```
int getX()
```

Returns:

`The x coordinate of the last touch on touch screen devices and the current mouse position on desktop for the first pointer in screen coordinates. The screen origin is the top left corner.`

getY

```
int getY()
```

Returns:

`The y coordinate of the last touch on touch screen devices and the current mouse position on desktop for the first pointer in screen coordinates. The screen origin is the top left corner.`

isKeyPressed

```
boolean isKeyPressed(int key)
```

Returns whether the key is pressed.

Parameters:

`key - The key code as found in Input.Keys.`

Returns:

`true or false.`

random

```
public static float random(float start,
                           float end)
```

Returns a random number between start (inclusive) and end (exclusive).

add

```
public void add(T value)
```

nanoTime

```
public static long nanoTime()
```

Returns:

The current value of the system timer, in nanoseconds.

iterator

```
public Array.ArrayIterator<T> iterator()
```

Specified by:

iterator in interface java.lang.Iterable<T>

hasNext

```
public boolean hasNext()
```

Specified by:

hasNext in interface java.util.Iterator<T>

next

```
public T next()
```

Specified by:

next in interface java.util.Iterator<T>

getDeltaTime

```
float getDeltaTime()
```

Returns:

the time span between the current frame and the last frame in seconds.

remove

```
public void remove()
```

Specified by:

remove in interface java.util.Iterator<T>

overlaps

```
public boolean overlaps(Rectangle r)
```

Parameters:

r - the other Rectangle

Returns:

whether this rectangle overlaps the other rectangle.

play

```
long play()
```

Plays the sound. If the sound is already playing, it will be played again, concurrently.

Returns:

the id of the sound instance if successful, or -1 on failure.

Conclusão

Continuando a programação da “Atividade ILJ003-Framework02”, agora fazendo a criação de vetores, e configurando a mecanica como o teclado(*isKeyPressed*), ou a questão da colisão ser tocado ou não, e as coordenadas x e y(*isTouched, getX and getY*), além de ser desenvolver um sistema de gereção de gotas automaticas(*random, add, next e hasNext*), alem de contar o tempo(g*etDeltaTime, nanoTime*), além de ter a colisão que apaga as gotas(*remove*), e fica sobreposto ao balde que faz um som de gota(*overlaps, play*).