# Class Viewer

java.lang.Object
    Viewer

**All Implemented Interfaces:**

PhotoViewer

---

```
public class Viewer
extends java.lang.Object
implements PhotoViewer
```

## Constructor Summary

### Constructors

| Constructor and Description |
| --- |
| **Viewer**() |

## Method Summary

**All Methods**   Instance Methods   Concrete Methods

| Modifier and Type | Method and Description |
| --- | --- |
| void | **clickF**()<br>This event takes a 'front click' user input from the GUI and sends it to the controller to handle. |
| void | **clickL**()<br>This event takes a 'right click' user input from the GUI and sends it to the controller to handle. |
| void | **clickR**()<br>This event takes a 'left click' user input from the GUI and sends it to the controller to handle. |
| void | **initialise**(PhotoController controller)<br>This initializes the controller |
| void | **initImageViews**()<br>Maps String names to FXML Image Views so that it can handle |

requests from the controller faster It is called during initialization.

| void | **initMenuItems**() |
| | Maps String names to FXML Menu Items so that it can handle request to the controller faster It is called during initialization. |
| void | **pickUp**(javafx.event.ActionEvent e) |
| | This event takes a 'pick up' user input from the GUI and sends it to the controller to handle. |
| void | **putDown**(javafx.event.ActionEvent e) |
| | This event takes a 'put down' user input from the GUI and sends it to the controller to handle. |
| void | **setController**(PhotoController controller) |
| | This function initializes the controller |
| void | **setDisable**(java.lang.String menuItem, boolean disable) |
| | This disables the menu item when a change happens |
| void | **setImage**(java.lang.String imageView, javafx.scene.image.Image image) |
| | This sets the image for a particular image view (map or pickable) |
| void | **setImage**(java.lang.String imageView, javafx.scene.image.Image image, int x_coord, int y_coord) |
| | This sets the main image as it is a panorama, only part is set. |
| void | **setText**(java.lang.String text) |
| | This sets the text for the text Box on the GUI with the location |
| void | **setVisible**(java.lang.String imageView, boolean visibility) |
| | This sets a pickable item's visibility or that of the forward arrow |

## Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### Viewer

public Viewer()

## *Method Detail*

### initialise

```
public void initialise(PhotoController controller)
```

**Description copied from interface: `PhotoViewer`**
This initializes the controller

**Specified by:**
initialise in interface PhotoViewer

### initImageViews

```
public void initImageViews()
```

Maps String names to FXML Image Views so that it can handle requests from the controller faster It is called during initialization.

### initMenuItems

```
public void initMenuItems()
```

Maps String names to FXML Menu Items so that it can handle request to the controller faster It is called during initialization.

### setController

```
public void setController(PhotoController controller)
```

**Description copied from interface: `PhotoViewer`**
This function initializes the controller

**Specified by:**
setController in interface PhotoViewer

### setText

```
public void setText(java.lang.String text)
```

**Description copied from interface: `PhotoViewer`**
This sets the text for the text Box on the GUI with the location

**Specified by:**
setText in interface PhotoViewer
**Parameters:**
text - the location in a String

### setImage

```
public void setImage(java.lang.String imageView,
                     javafx.scene.image.Image image)
```

**Description copied from interface:** `PhotoViewer`

This sets the image for a particular image view (map or pickable)

**Specified by:**

setImage in interface PhotoViewer

**Parameters:**

imageView - the image view

image - the image

### setImage

```
public void setImage(java.lang.String imageView,
                     javafx.scene.image.Image image,
                     int x_coord,
                     int y_coord)
```

**Description copied from interface:** `PhotoViewer`

This sets the main image as it is a panorama, only part is set.

**Specified by:**

setImage in interface PhotoViewer

**Parameters:**

imageView - the image view

image - the image

x_coord - the x coordinate from which the image starts

y_coord - the y coordinate from which the image starts

### setVisible

```
public void setVisible(java.lang.String imageView,
                       boolean visibility)
```

**Description copied from interface:** `PhotoViewer`

This sets a pickable item's visibility or that of the forward arrow

**Specified by:**

setVisible in interface PhotoViewer

**Parameters:**

imageView - Which pickable item or arrow

visibility - visible or not visible

---

**setDisable**

---

public void setDisable(java.lang.String menuItem,
                       boolean disable)

**Description copied from interface: `PhotoViewer`**

This disables the menu item when a change happens

**Specified by:**

setDisable in interface PhotoViewer

**Parameters:**

menuItem - the menu item

disable - true or false

---

**pickUp**

---

public void pickUp(javafx.event.ActionEvent e)

This event takes a 'pick up' user input from the GUI and sends it to the controller to handle.

**Parameters:**

e - the action event corresponding to the menu select

---

**putDown**

---

public void putDown(javafx.event.ActionEvent e)

This event takes a 'put down' user input from the GUI and sends it to the controller to handle.

**Parameters:**

e - the action event corresponding to the menu select

---

**clickL**

---

public void clickL()

This event takes a 'right click' user input from the GUI and sends it to the controller to handle.

---

**clickR**

---

public void clickR()

This event takes a 'left click' user input from the GUI and sends it to the controller to handle.

**clickF**

```
public void clickF()
```

This event takes a 'front click' user input from the GUI and sends it to the controller to handle.

**clickF**

```
public void clickF()
```

This event takes a 'front click' user input from the GUI and sends it to the controller to handle.