**Introduction**

**What are Functions as Objects?**

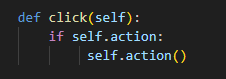
In computing, function objects, also known as "functors" are instances of classes designed to act as functions. This is done by implementing the function call operator method ‘operator()’ in the class. This means that functions are treated like any other object. They can be passed as arguments to other functions, define functions inside other functions and they can be assigned to variables.

In our 2D slider puzzle game, functions as objects were utilized this way:

**Python:**

A black screen with white text

Description automatically generated

****

These are snippets from our code that can be found in the file Button.py. This shows how functions are treated as objects through the action attribute in the Button class. This allows for the execution of different actions on button clicks. This shows the flexibility of using functions as assignable and callable objects in Python.

**Java**

**A computer screen shot of code

Description automatically generated**

In our code, we use a MouseAdapter to show us how java treats functions as objects. This is done by encapsulating the logic for the mousePressed event in an anonymous class. This method connects actions to objects, demonstrating Java’s way of managing events in an object-oriented way.

**What’s Event handling?**

In Programming terms, an event is an action that allows the program to respond to user interaction (UI). A program waits for an event to occur and then responds by executing a certain code.

Every Event needs to have an event handler and an event receiver. An event handler is a call-back routine that takes care of the input received in the program. Once an event is triggered, event handers execute a block of code as a response to it.

Event Receivers are functions in a program that waits for an event to occur. When one does occur, the receiver executes the event handler that’s associated with that event.

In our 2D slider puzzle game, event handling was utilized this way:

**Python**

**A computer screen with colorful text

Description automatically generated**

For our code we used a dictionary that maps pygame events (like pygame.QUIT) to corresponding lambda functions. When an event occurs, the matching action is automatically done, making the game respond to what the player does. In our code, the method that was used allowed us to manage events such as for example, exiting the game or clicking the mouse in a clear and organized way. We did this by directly running the related actions within the event loop.

**Java**

1. A computer screen shot of text

   Description automatically generated

The MouseAdapter snippet handles mouse press events on the game's panel. It determines the click tile, attempts to move it, and then proceeds to update the panel’s display. This method enables advanced mouse interaction handling inside the game.

1. **A computer code with colorful text

   Description automatically generated with medium confidence**

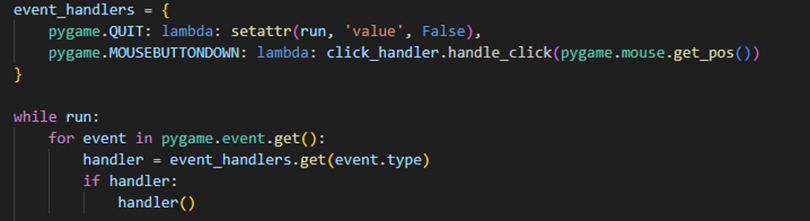
This snippet, found in our code, shows a lambda expression used to add an ActionListener to the newGameButton. Upon being clicked, it triggers the puzzle model to reset and the graphics panel to repaint, showcasing a modern, concise approach to handling button click events in Java.

**What are lambda expressions?**

Lambda expressions are concise, anonymous functions made using the keyword “lambda”. They take any number of arguments but only include one expression. This simplicity, makes them inline and are typically used for short, simple operations. The use of lambda is crucial as it signals Python that what follows is, in fact, an anonymous function.

In our 2D slider puzzle game, lambda expressions were utilized in this way:

**Python**

1. 

As seen in a previous example, Lambda functions are linked to pygame events to handle actions like quitting the game and processing mouse clicks.

1. 

The lambda expression creates an unnamed function that calls self.shuffle\_tiles() when executed. It's assigned to shuffle\_button\_action and can serve as a button click handler to shuffle tiles in a game.

**A computer screen with text

Description automatically generatedJava**

Lambda expression is used for adding an action listener to the newGameButton. It states the behavior without the need to implement a full anonymous class, showing the use of lambda expressions for event handling in Java. This lambda simplifies event handling by directly linking the button click with actions to reset the puzzle model.