Creating a Selection Set in AutoLISP? (Step by step)

Description

Understanding and using selection sets in AutoLISP enables designers to manage multiple objects as a unified entity, modify properties of grouped objects, and navigate complex drawings efficiently. In this comprehensive guide, we will delve into the process of creating a selection set in AutoLISP, the powerful possibilities it offers, and address some frequently asked questions that arise in the process.

Creating a Selection Set in AutoLISP

AutoLISP, a dialect of the LISP programming language built specifically for the full version of AutoCAD, has become an indispensable tool for CAD operators worldwide. A common task performed using AutoLISP is creating a selection set. This process allows you to group and handle multiple objects in AutoCAD as a single entity, thus increasing efficiency and productivity.

Selection sets in AutoLISP have a diverse range of applications, including:

- Changing properties of multiple objects simultaneously
- Deleting multiple objects in one command
- Isolating specific elements in a complex drawing

Now, let’s delve into the process of creating a selection set using AutoLISP.

Understanding Selection Sets

Before you begin, it’s essential to understand what a selection set is in the context of AutoLISP. In AutoCAD, a selection set is a group of objects that you can manage as a single entity. With AutoLISP, you can programmatically create selection sets, enabling you to manipulate multiple objects simultaneously with great precision and efficiency.

How to Create a Selection Set in AutoLISP

Creating a selection set in AutoLISP involves the following steps:

1. **Start the AutoLISP function:** To begin creating a selection set, you’ll need to start an AutoLISP function using the `defun` keyword, followed by the function name and any necessary arguments.

```
(defun c:myFunction (/ selectionSet)
```

2. **Create an empty selection set:** After defining your function, you can create an empty selection set using the `ssget` command with the “X” option. This will return a selection set that includes all objects in the drawing.
3. **Manipulate the selection set:** Once you have your selection set, you can manipulate it as needed. For instance, you might change the properties of all objects within the set, or remove objects from the set.
4. **Close the function:** After manipulating the selection set, remember to close the function with the `princ` function to avoid leaving the command line open.

```
(princ)
)
```

Altogether, your function might look something like this:

```
(defun c:myFunction (/ selectionSet)
(setq selectionSet (ssget “X”))
; Manipulate selectionSet here
(princ)
)
```

Note: The `/` symbol is used to declare local variables in AutoLISP. In the above function, `selectionSet` is a local variable.

Enhancing Efficiency with Filtered Selection Sets

In many instances, you’ll want to create a filtered selection set, i.e., a selection set that only includes specific types of objects. For example, you might want a selection set that only includes line objects, or perhaps only objects on a specific layer.

The `ssget` function provides the capability to create such filtered selection sets. It uses a filter list asits second argument, which is a list of codes and values that define the criteria for the selection. For example, the following function creates a selection set of all line objects in the drawing:

```
(defun c:myFunction (/ selectionSet)
(setq selectionSet (ssget “X” ‘((0 . “LINE”))))
; Manipulate selectionSet here
(princ)
)
```

In the filter list `((0 . "LINE"))`, the `0` is a group code that signifies the object type, and “LINE” is the value.

Useful AutoLISP Commands for Working with Selection Sets

AutoLISP provides several commands for working with selection sets. Here are a few:

Command	Description
`ssget`	Creates a selection set.
`ssadd`	Adds an object to a selection set.
`ssdel`	Removes an object from a selection set.
`sslength`	Returns the number of objects in a selection set.
`ssmemb`	Checks if an object is in a selection set.

FAQ on Creating a Selection Set in AutoLISP

1. Can I Create a Selection Set from a User’s Selection in AutoLISP?

Yes, you can create a selection set from a user’s selection in AutoLISP. Instead of using the “X” option with the `ssget` command, which selects all objects in the drawing, you can call `ssget` with no arguments. This prompts the user to make a selection, and the selected objects become the selection set. This approach can be useful when you want to provide the user with flexibility in determining what objects the selection set should include.

The syntax would look like this:

```
(defun c:myFunction (/ selectionSet)
(setq selectionSet (ssget))
; Manipulate selectionSet here
(princ)
)
```

2. Can I Add or Remove Specific Objects from a Selection Set?

Absolutely! The `ssadd` and `ssdel` functions enable you to add and remove specific objects from a selection set, respectively. This can be beneficial when you want to modify an existing selection set based on new requirements or changes in the drawing.

For example, the following code would add a selected entity to an existing selection set:

```
(setq entity (entsel))
(ssadd entity selectionSet)
```

In the code above, `(entsel)` prompts the user to select an entity, and `(ssadd entity selectionSet)` adds the selected entity to the existing selection set.

3. How Can I Check the Size of a Selection Set in AutoLISP?

The `sslength` function in AutoLISP provides the number of objects in a selection set. This information can be useful when you need to verify the number of objects your program will be manipulating, or when you need to debug your program.

Here's a sample code that demonstrates how to use `sslength`:

```
(setq numObjects (sslength selectionSet))
```

This code will set the variable `numObjects` to the number of objects in the selection set `selectionSet`.

4. Can I Iterate Over a Selection Set in AutoLISP?

Yes, iteration over a selection set is not only possible but also a common operation in AutoLISP. By iterating over a selection set, you can perform an operation on each individual object within the set. This is typically done using the `ssname` function in conjunction with a loop.

The following is an example of a function that iterates over a selection set and performs an operation on each entity:

```
(defun c:myFunction (/ selectionSet entity)
(setq selectionSet (ssget “X”))
(setq numObjects (sslength selectionSet))
(repeat numObjects
(setq entity (ssname selectionSet 0))
; Perform operation on entity here
(setq selectionSet (ssdel entity selectionSet))
)
(princ)
)
```

In the code above, `(ssname selectionSet 0)` retrieves the first object in the selection set, which can then be manipulated as needed.

5. What If I Want to Filter a Selection Set Based on Properties Other Than Type?

In addition to type, AutoLISP allows you to filter selection sets based on many different object properties, including layer, color, and line type, to name a few. Each property has an associated group code that you can use in the filter list of the `ssget` function.

For example, if you wanted to select all objects on layer “Layer1”, you would use group code 8, which represents the layer of an object:

```
(setq selectionSet (ssget “X” ‘((8 . “Layer1”))))
```

This would create a selection set of all objects on layer “Layer1”.

6. How Can I Handle Errors When Creating Selection Sets in AutoLISP?

Error handling in AutoLISP is generally done using the `*error*` function. This special function gets called whenever an error occurs during the execution of your program.

You can define your own `*error*` function to customize how your program responds to errors. For example, you might want to display a specific message to the user, or you might want to clean up some data before terminating the program.

Here’s an example of a simple `*error*` function:

```
(defun *error* (msg)
(if (not (null msg))
(princ (strcat “\nError: ” msg))
)
(princ)
)
```

This function will print an error message to the command line whenever an error occurs.

7. Can I Create Selection Sets of Objects in Specific Locations?

Yes, you can create selection sets of objects located within a specific area in your drawing. You can do this by providing the `ssget` function with a window or crossing selection, which allows you to specify two points that define a rectangular area in your drawing. Only objects within this area will be included in the selection set.

Here’s an example of how to create a window selection:

```
(setq selectionSet (ssget “W” ‘(0 0) ‘(10 10)))
```

This will create a selection set of all objects within the window defined by the points (0,0) and (10,10).

Conclusion

Creating a selection set in AutoLISP is a powerful tool in the hands of CAD designers. It streamlines operations by allowing batch processing of objects and facilitates precise control over specific groupings of objects in a drawing. This operation, while simple to implement, can considerably enhance productivity, accuracy, and overall performance in AutoCAD. With the guidelines and tips shared in this guide, you can leverage this feature to your advantage, regardless of the complexity or scale of your project.