**Drag&Drop**

**DataTransfer**

Drag-and-drop is part of the more general area of data transfer. Data transfer includes drag-and-drop and copy-and-paste operations.

Both types of operations require:

* A **source object** that provides the data.
* A way to temporarily store the transferred data.
* A **target object** that receives the data.

In a copy-and-paste operation, the system ***clipboard*** is used to temporarily store the transferred data; in a drag-and-drop operation, a [***DataObject***](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dataobject)is used to store the data.

**Initiates a drag-and-drop operation**

The drag source initiates a drag-and-drop operation by calling the static [DragDrop.DoDragDrop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dodragdrop) method and passing the transferred data to it.

The [DoDragDrop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dodragdrop) method will automatically wrap the data in a [DataObject](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dataobject) if necessary.

For greater control over the data format, you can wrap the data in a [DataObject](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dataobject) before passing it to the [DoDragDrop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dodragdrop) method.

**Implementing Drag-and-Drop**

To implement basic drag-and-drop, you will complete the following tasks:

1. Identify the element that will be a **drag source**
2. Create an event handler on the drag source that will initiate the drag-and-drop operation. The event is typically the [MouseMove](https://docs.microsoft.com/en-us/dotnet/api/system.windows.uielement.mousemove) event.
3. In the drag source event handler, call the [DoDragDrop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dodragdrop) method to initiate the drag-and-drop operation. In the [DoDragDrop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dodragdrop) call, specify the drag source, the data to be transferred, and the allowed effects.
4. Identify the element that will be a **drop target**.
5. On the drop target, set the [AllowDrop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.uielement.allowdrop) property to true.
6. In the drop target, create a [Drop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.drop) event handler to process the dropped data.
7. In the [Drop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.drop) event handler, extract the data from the [DragEventArgs](https://docs.microsoft.com/en-us/dotnet/api/system.windows.drageventargs) by using the [GetDataPresent](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dataobject.getdatapresent) and [GetData](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dataobject.getdata) methods.
8. In the [Drop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.drop) event handler, use the data to perform the desired drag-and-drop operation.

### **Enabling an Element to be a Drag Source**

An object that is a **drag source** is responsible for:

* Identifying when a drag occurs. ->MouseMove()
* Initiating the drag-and-drop operation. -> DoDragDrop()
* Identifying the data to be transferred. -> DataObject wrap-uje Spomenik
* Specifying the effects that the drag-and-drop operation is allowed to have on the transferred data. -> Copy

The [DoDragDrop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dodragdrop) method takes three parameters:

* **dragSource** – A reference to the dependency object that is the source of the transferred data; this is typically the source of the [MouseMove](https://docs.microsoft.com/en-us/dotnet/api/system.windows.uielement.mousemove) event.
* **data** - An object that contains the transferred data, wrapped in a [DataObject](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dataobject).
* **allowedEffects** - One of the [DragDropEffects](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdropeffects) enumeration values that specifies the permitted effects of the drag-and-drop operation. (move, copy, none)

### **Enabling an Element to be a Drop Target**

An object that is a drop target is responsible for:

* Specifying that it is a valid drop target. ( set [AllowDrop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.uielement.allowdrop) property to true)
* Responding to the drag source when it drags over the target. -> Drop, DropEnter
* Checking that the transferred data is in a format that it can receive.
* Processing the dropped data.

During a drag-and-drop operation, the following sequence of events occurs on the drop target:

1. [DragEnter](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dragenter)
2. [DragOver](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dragover)
3. [DragLeave](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.dragleave) or [Drop](https://docs.microsoft.com/en-us/dotnet/api/system.windows.dragdrop.drop)

- DragEnter is called when the mouse enters your control **while dragging** something.

- DragOver is called while the mouse is **still in that area and still dragging**

**MVVM**

