

CMPT 381 Assignment 4

Multiple Selection, Grouping, and Cut/Copy/Paste

Due: **Wednesday**, December 15, 11:59pm

NOTE: help from the teaching assistants will not be available after Dec. 8

Overview

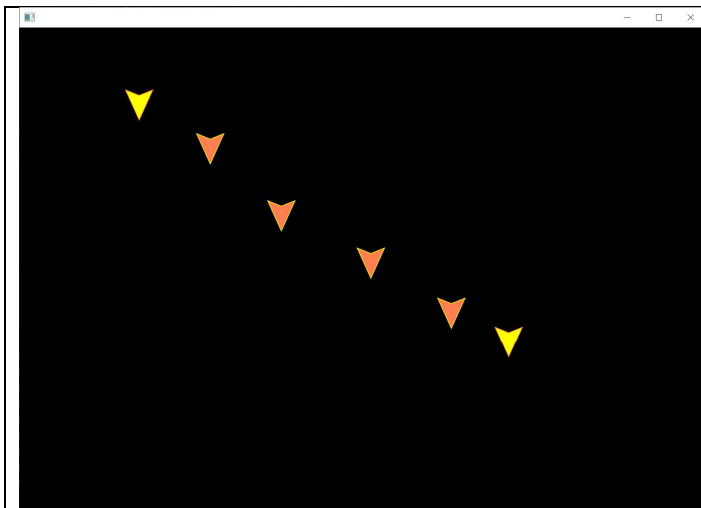
In this assignment you will build a graphical editor in JavaFX that lets the user set up levels for a video game. You will develop only a portion of the level editor, allowing the user to create, position, and group enemy spaceships. The assignment is divided into four parts: part 1 covers selection and multiple selection of the ships; part 2 covers grouping and basic group transforms; part 3 works with an in-app clipboard to cut, copy, and paste ships; and part 4 is a challenge level to support rotation of groups.

Part 1: Multiple Selection (25 marks)

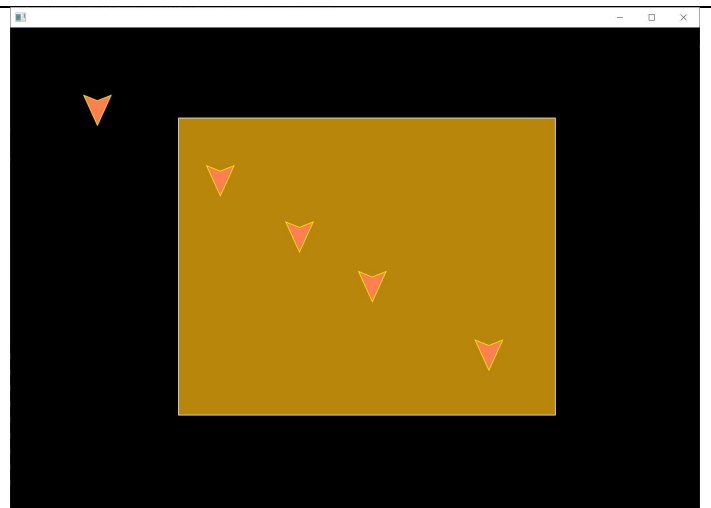
Start with the basic system provided to you (A4Basics.zip), which implements ship creation, selection, and dragging in an MVC architecture. Extend the system to implement multiple selection – either by clicking with the mouse or by dragging a rubberband rectangle.

Interaction Requirements:

- Preserve the single selection mechanisms implemented in the basic system
- Interaction with multiple selection mechanisms should follow the guidelines discussed in lectures
- When the user Control-clicks on a ship, it is added to / removed from the selection set
- When the user presses the left mouse button on the background and drags, a rubberband rectangle is shown; when the user releases the mouse button, the ships within the rectangle become the selection
- When the user holds Control and draws a rubberband rectangle and then releases the mouse button, all ships within the rectangle are added to / removed from the selection set
- Clicking on the background without moving clears the selection
- When multiple ships are selected, dragging any selected ship drags all selected ships by the same amounts



The system after creating six ships, then clicking on the lower-right ship and Control-clicking on the upper-left ship.



The user is about to release the mouse button after creating a rubberband rectangle (four ships will be selected).

Code requirements:

- Add all new components to the appropriate place in the MVC architecture
- Follow the guidelines discussed in lectures for the data structures used to store and manage multiple selections
- Add method `isContained()` to determine if a ship is in the rubberband rectangle
- Since the system has only one view, you do **not** need to use normalized coordinates.

Part 2: Grouping (30 marks)

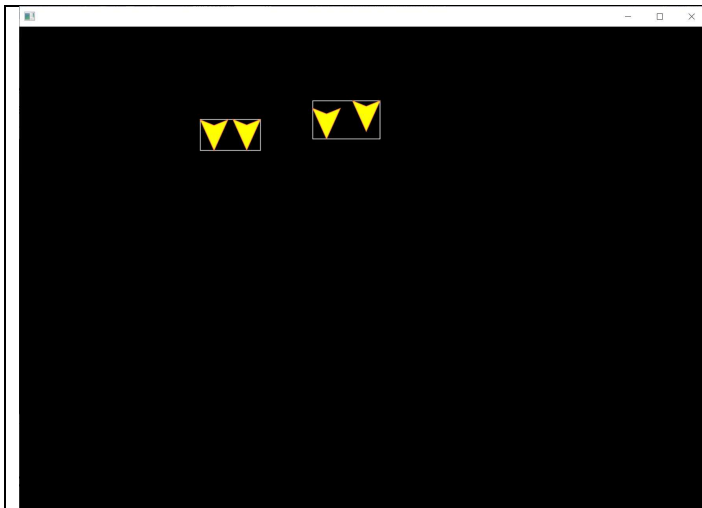
Extend your system to allow sets of ships to be grouped together.

Interaction Requirements:

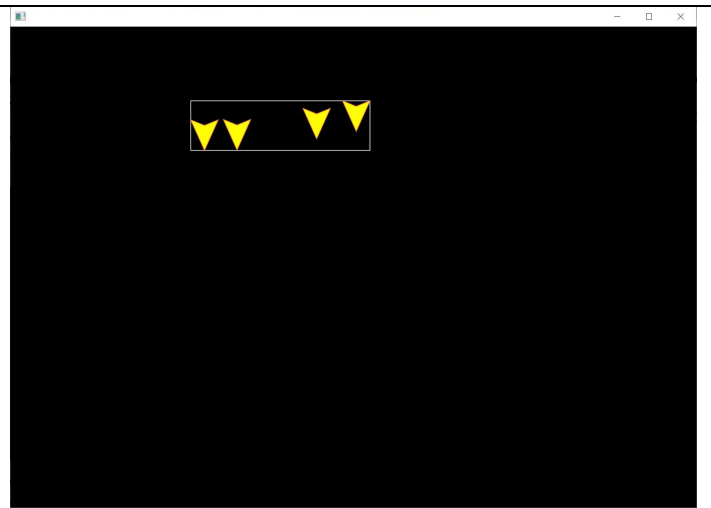
- When multiple ships are selected, the user can group them by pressing the G key
- Grouping causes the selected ships to be added to a group, with the group now selected instead of the individual ships
- A selected group shows the bounding box of the group as a white line
- If the user presses the mouse on any ship in a group and drags, the entire group moves
- Groups can also be grouped (and groups of groups, etc.)
- The user can ungroup a selected group by pressing the U key
- Ungrouping replaces the group with the individual ships (or groups) that were in the group
- When a group is ungrouped, all of the group's children are selected after ungrouping
- If there is more than one item selected, or if the selected item is not a group, ungrouping does nothing

Code Requirements:

- Add interface Groupable as described in lectures to allow both Ship and ShipGroup objects to coexist in the model and selection
- Change your Ship classes to implement Groupable
- Add class ShipGroup to your project; objects of this class will hold groups of ships; ShipGroup should also implement Groupable
- Convert the model's collection of items and the InteractionModel's selection collection to use type Groupable
- For translation, ShipGroup should use the "delegate to children" approach described in lectures



The system contains two groups, each of which has two ships. Both groups have been selected using multiple selection.



After pressing 'G', the two groups have themselves been grouped

Part 3: Clipboards and Cut/Copy/Paste (30 marks)

Interaction Requirements:

- If there is a selection, pressing Ctrl-X puts the selected items on the app's clipboard and removes them from the model
 - The selected items can any combination of ships and/or groups
- Pressing Ctrl-C puts a copy of the selected items on the app's clipboard (and does not remove them from the model)
 - Only one collection of items can be in the clipboard at a time; if the user copies a second selection, the first one is discarded from the clipboard
- Pressing Ctrl-V puts a copy of the item(s) that are in the clipboard into the model
 - The user can paste the items in the clipboard multiple times
 - The pasted items become selected when they are pasted; any existing selection is unselected

Code Requirements:

- Create class ShipClipboard to store items that have been cut or copied. Note: do not use the system clipboard; you must create your own clipboard class. Your clipboard should be managed by the InteractionModel
- Add handlers to your controller to handle keypress events for Ctrl-X, Ctrl-C, and Ctrl-V.
 - Note that class KeyEvent has method “isControlDown()” that you can use to determine whether the Control key is pressed when a keypress event occurs
- Add model methods for cutting, copying, and pasting
- When copying to the clipboard and when pasting, you will need to make a *deep copy* of the objects, because you may be pasting them multiple times, and each paste needs to add different objects to the model

Resources for part 3:

- KeyEvent API: <https://openjfx.io/javadoc/15/javafx.graphics/javafx/scene/input/KeyEvent.html>
- Details on cut/copy/paste operations: Olsen text, chapter 15
- Tutorial on Java copying: <https://dzone.com/articles/java-copy-shallow-vs-deep-in-which-you-will-swim>



Part 4: Rotation with Groups (15 marks)

Extend your code so that the user can rotate ships, multiple-selected ships, or groups.

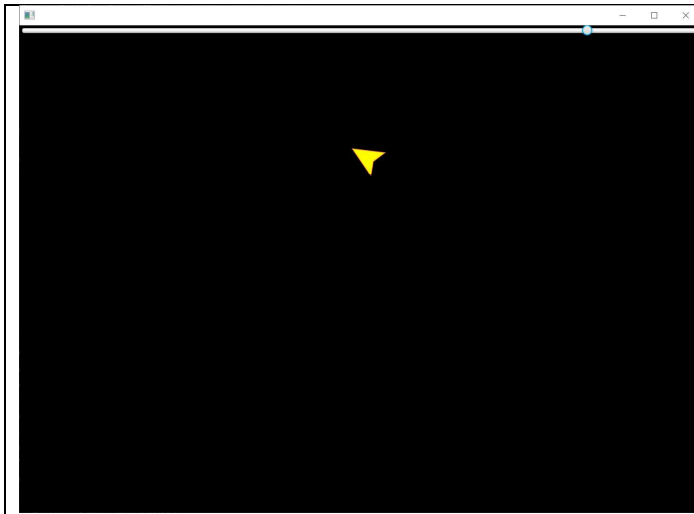
Note: this part is challenging.

Interaction Requirements:

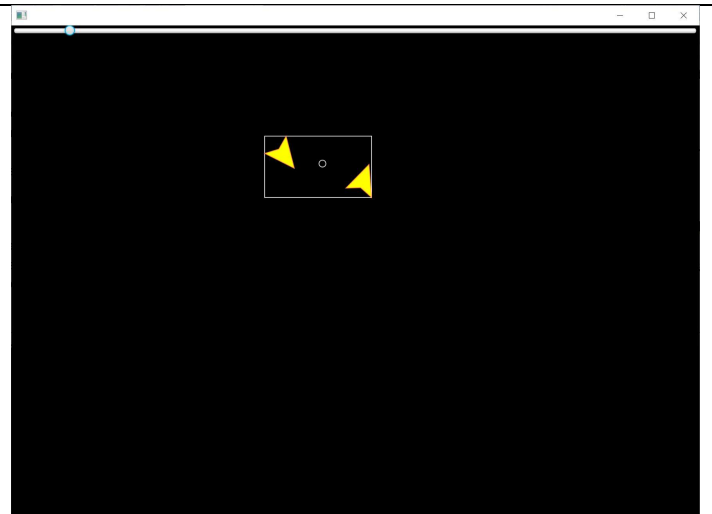
- For any selected ships or groups, the user can drag on a slider at the top of the window to rotate the ships/groups left and right
- Rotations of individual ships should be preserved when ships are grouped together
- Groups rotate around their centre point
- Ships rotate around their 0,0 point (using the home coordinates that are in the Ship class)
- When a group is ungrouped, rotation should not change (i.e., the ships should stay where they are and at the rotation they showed when in the group; the only thing that changes is that they are now ungrouped)

Code Requirements:

- Add a slider widget to the top of the window to allow the user to control rotation (the slider handle should be located in the middle of the slider whenever an object is selected, and should allow 180 degrees of rotation in both directions)
- Rotating multiple items follows the same approach as described in lectures (i.e., a group tells each of its children to rotate)
- Rotate the individual points of each ship's polygon, with the centre of rotation as either the ship's centre (if only a single ship is selected) or the group's centre (if a group is selected)
- Ships already have a centre point of 0,0, but you will need to add code to keep track of the centre point of a group (note that the centre point will need to be updated when the group changes or is ungrouped)



Single ship rotated using the slider at the top of the window; the ship rotates around its own centre



A group of two ships; when rotated, the ships rotate around the group's centre point

What to hand in (each student will hand in an assignment)

- Create a zip file of your JavaFX/IDEA project zip (File → Export → Project to Zip file...).
- Add a readme.txt file to the zip that describes the organization of your handin and provides any comments to the markers about what to look at in your code (as always, systems for 381 should never require the marker to install external libraries). If parts of the system don't work, but you want us to look at the code for these parts, please explain in your readme.txt

Where to hand in

Hand in your zip file to the Assignment 3 link on the course Canvas site.

Evaluation

- *Part 1:*
 - The selection is correctly represented as a collection
 - Control-clicking adds to / removes from the selection set as described above
 - Rubber-band selection correctly selects all items within the rectangle
 - The isContained() method is implemented correctly
 - The selection set is visually clear in the view
- *Part 2:*
 - The code correctly implements the ShipGroup class and Groupable interface
 - Keyboard commands for grouping and ungrouping are correctly implemented
 - The MVC architecture uses the Groupable type to store selection and rubber-band testing
 - The view correctly shows a selected group's bounding box
 - The user can successfully group and ungroup multiple levels deep
- *Part 3:*
 - The ShipClipboard class is correctly implemented and referred to
 - Handlers for Control-X, Control-C, and Control-V are correctly implemented
 - Cutting and pasting correctly remove / add objects from the model
 - Pasting correctly implements deep copying

If parts of your system have only partial implementations (e.g., a feature does not work but has been partially developed), clearly indicate this in your readme.txt file. Code should be appropriately documented and tested (although documentation will not be explicitly marked). No late assignments will be allowed, and no extensions will be given, without medical reasons.