CSCI-513

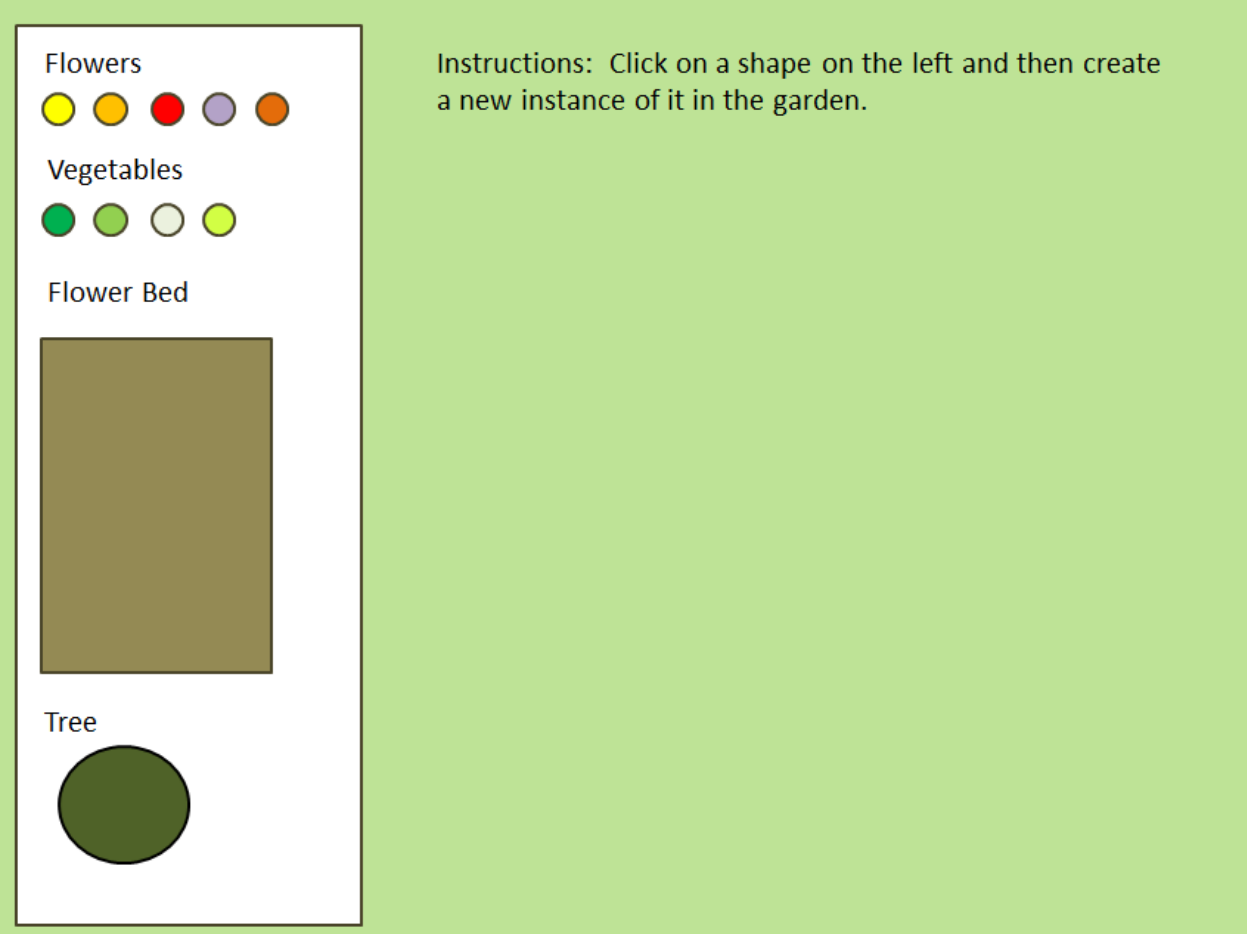
Lab 3: Garden Layout

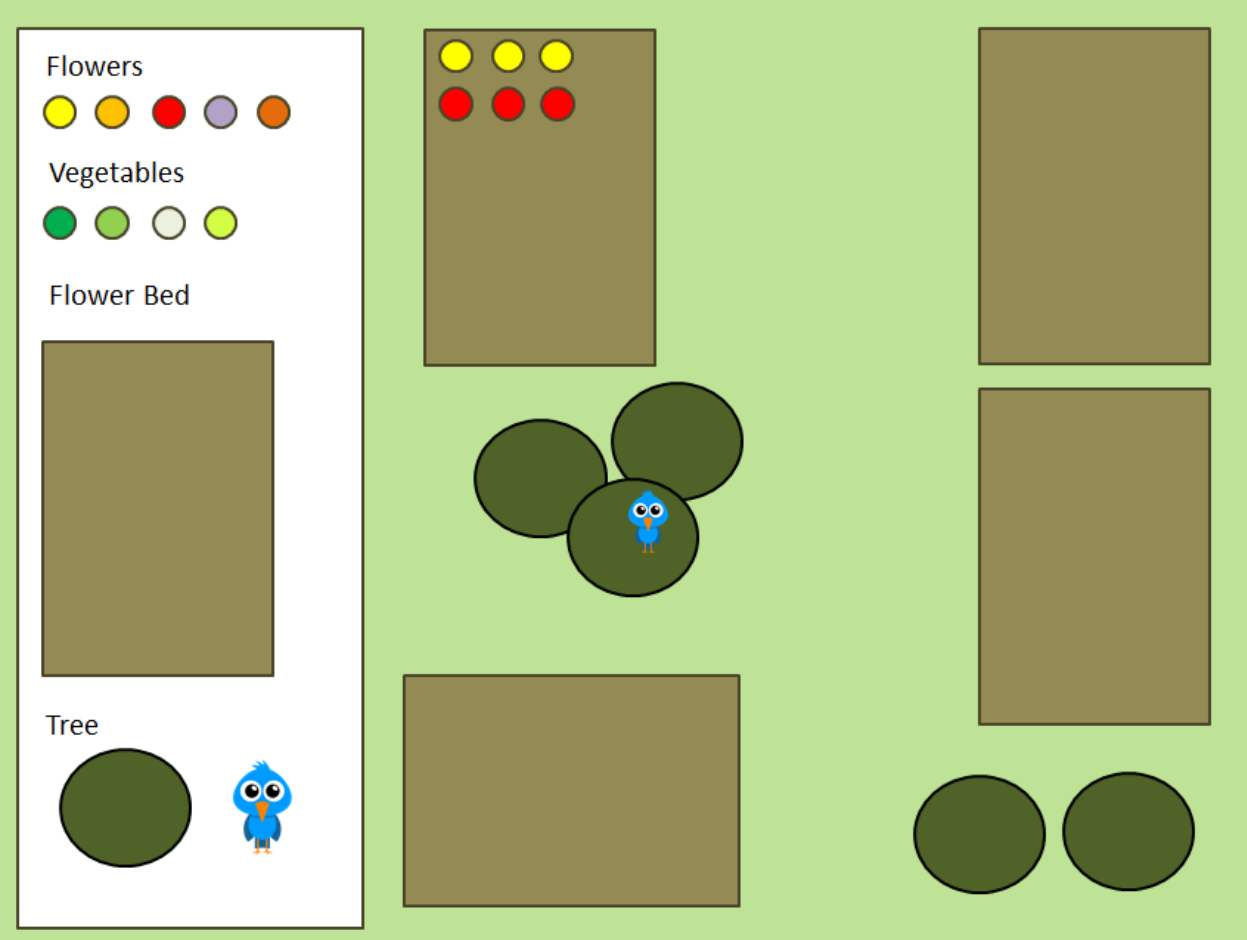
Objective

In this lab we are going to practice using the mouse – and then implement the Composite pattern. We will create a garden planning system that you can use to layout garden plots that include vegetables and flowers. You will be able to move flower beds around by dragging them.

Description:

In this lab we are going to practice using the mouse – and then implement the Composite pattern. We will create a garden planning system that you can use to layout garden plots that include vegetables and flowers. You will be able to move flower beds around by dragging them and all their contents will move with them.





Step # 1:

Create a JavaFX application called “GardenLayout”. Extend Application. Put all the normal things into Start(…).

Step #2:

For this Lab we are going to work with a tool kit of only two items. A small circle representing a flower, and a brown larger rectangle representing a flower bed.

The first thing we will do is to create a composite hierarchy. It needs to have:

1. An interface
2. A composite class (for now that will be the flower bed).
3. A leaf node. (for now we will use that to draw our flower).

Create these three classes and set them up with the code you need for the Composite pattern. We have discussed how to do that in class. It is in the slides and in the worksheet we went through in class.

Step #3:

Create a single red flower. We are going to add code to the class so that we can drag the flower across the screen.

Create a constructor for the flower class that takes three arguments: Position of flower (Point2D) Color of flower (Color.RED) Boolean movable (true/false).
In the constructor – create a circle representing the flower:

```
circle = new Circle();
circle.setCenterX( x coord);
circle.setCenterY( y coord);
circle.setRadius(Try different sizes here);
circle.setFill(choose a color);
circle.setStroke(Color.BLACK);
circle.setStrokeWidth(1);
```

Display the circle onto the garden. You'll most likely do this from within the GardenLayout class.

Step #4:

Add a mouse event handler inside the GardenLayout class, start method – so that you can drag the red circle across the garden.

```
scene.setOnMouseDragged(mouseHandler);
scene.setOnMouseReleased(mouseHandler);
scene.setOnMousePressed (mouseHandler);
```

Now write the mouseHandler (I will go through this line by line in the Lab).

```
EventHandler<MouseEvent> mouseHandler = new EventHandler<MouseEvent>() {

    @Override
    public void handle(MouseEvent mouseEvent) {

        Point2D  clickPoint = new Point2D(mouseEvent.getX(), mouseEvent.getY());
        System.out.println(clickPoint.getX()+ " " + clickPoint.getY());
        String eventName = mouseEvent.getEventType().getName();

        switch(eventName) {

        case("MOUSE_DRAGGED"):

            //try to remove this
            if(lastPosition != null){

                System.out.println("Dragging");
                double delataX = clickPoint.getX()-lastPosition.getX();
                double delataY = clickPoint.getY()-lastPosition.getY();
                flower.move(delataX,delataY);
            }

        break;

        }

        lastPosition = clickPoint;

    }
};
```

Finally, write the move(deltaX, deltaY) method. This also goes into the flower class. I suggest that this is specified in the Composite Interface so that you can perform moves on all items in the hierarchy.

```
@Override
public void move(double dx, double dy) {
        circle.setCenterY(circle.getCenterY()+dy);
        circle.setCenterX(circle.getCenterX()+dx);
}
```

Step #5:

Repeat the whole thing for the flower garden. Remember that the flower garden is a COMPOSITE node.

Step #6:

Add another case to the mouseHandler for “MOUSE_RELEASED”. Check to make sure that you are currently dragging an object (I’ll leave that to you to figure out). If the mouse is released – check to see if the item is IN a composite node. This means that you’ll need a method that looks something like public iComposite isInNode(iComposite gardenEntity).

If you release a flower inside another entity then you should register it with that entity (i.e. it becomes a component node of the composite).

Step #7:

Final step for the Lab. If you drag a composite node – make sure that all its components move with it! You’ll need to modify the move(deltaX, deltaY) method so that it delegates move(deltaX, deltaY) to all its children too.