Using the objectdraw library with BlueJ

- 1. You must add objectdraw to the list of libraries used by your copy of BlueJ.
 - (a) Launch BlueJ.
 - (b) Select the "Preference" item from the BlueJ menu. Click the "Libraries" tab in the dialog box that appears. The dialog box should then look something like:

User Libraries		
Status	Location	Add
		Delete
System Libraries		
/Applications/BlueJ-130beta	a4/BlueJ-130 beta 4.app/Contents/Resources/Java/blueJcore.jar	(Testing)
/Applications/Bluel-130beta	a4/Bluel-130 beta 4.app/Contents/Resources/Java/Bluejeutol.ja	Testing)
Boot Libraries		5, 1
/Applications/BlueJ-130beta	a4/BlueJ-130 beta 4.app/Contents/Resources/Java/bluejcore.jar	(Boot class library)
/Applications/BlueJ-130beta	a4/BlueJ-130 beta 4.app/Contents/Resources/Java/bluejeditor.ja	ır (Boot class library) 🛛 🚺
/Applications/Bluel=130beta	a4/BlueJ-130 beta 4.app/Contents/Resources/Java/bluejext.jar (I	Boot class library)
At the production of products		

- (c) Click the "Add" button, navigate to find your copy of objectdraw.jar. Click choose once you have located and selected objectdraw.jar.
- (d) Quit BlueJ. The addition of objectdraw.jar will not take effect until you restart BlueJ.
- 2. Do *NOT* run objectdraw programs as applets under BlueJ.

Applets run under BlueJ do not get access to objectdraw.jar even if it has been added as explained above. Therefore, under BlueJ, objectdraw programs should not be run as applets. Instead, when using BlueJ a program should be run by first creating an instance of the Controller class and then invoking the new object's startController method. There are several ways to accomplish this.

(a) Install the "Objectdraw Invoker" BlueJ extension. The extension and instruction on how to install extensions can be found at

http://www.bluej.org/extensions/extensions.html

After this extension has been installed, a "Start Controller" item will appear in the menu displayed when your press the right mouse button (or the mouse button and control key) while pointing at the icon for any class that extends either the WindowController or Controller class. You can then run programs by simply selecting this item as shown below:



By default, programs run in this way will be displayed in a 400 by 400 window. You can alter the window size by entering the desired values in the dialog box displayed when you click on the "Extensions" tab in the window that appears after you select "Preferences" from the "BlueJ" menu as shown below:

ences	BlueJ: Pro		00
ibraries Extensions	Miscellaneous	Editor	
		nvoker	Objectdraw Ir
ndow Height: 500	00	Window Width:	1
			Submitter
		tom	User name: t
		tom@	Email: t
		ls	SMTP Details
Port: 25 Default: 25		12	Server name:
	lame: tom	e and password	🗌 Use name
) TLS	O SSL	connection: 💿	Use secure c
Ok Cancel			
Ok	_	_	_

(b) Programs written as described on our text can be run without installing the "Objectdraw Invoker" extension or modifying their code in any way as follows.

- i. First, either control-click (on a Macintosh) or right click (under Windows) on the icon for your Controller class in the BlueJ window. A menu will appear.
- ii. In the menu, select the "new" item as shown below. It should be the second item in the menu. DO NOT select the "Run Applet" item.



- iii. Click OK in the dialog box that appears. A red icon representing the object you have created will appear near the bottom of the BlueJ project window.
- iv. Next, either control-click or right-click on the red icon representing the object you created and use the menu that appears to navigate to the entry for the startController() method inherited from Controller as shown below.

New Class	Pole	
> ──Þ Compile	inherited from Object inherited from Component inherited from Container inherited from Panel inherited from Applet inherited from JApplet	< <applet>> MagnetGame</applet>
	inherited from Controller	void begin() [redefined in MagnetGame]
	inherited from WindowController	void destroy()
<u>magnetGa</u> MagnetGar	void begin() void onMouseClick(Location point) void onMouseDrag(Location point) void onMousePress(Location point)	AudioClip getAudio(String) Image getImage(String) void init() void placeInFrame(int, int) void resize(int, int)
	Inspect	void start()
	Kemove	void startController(int, int)
nagnetGal: MagnetGame		void stop()

- v. Select the startController() entry in the menu. A window with your program running inside should be displayed shortly. Alternately, you can select the startController(int, int) entry in the menu. In this case, you will be allowed to specify the width and height of the window to be created.
- (c) Navigating through BlueJ's menus to find the startController method can be a bit painful. As an alternative, if you are willing to ask your students to include a little non-standard code (relative to the examples in the text), you can simplify the process of running program as follows.
 - i. Add a constructor to your Controller class that simply invokes startController (either with or without parameters specifying the dimensions of the window for the program). For example, for the MagnetGame class, you would add a constructor of the form

```
public MagnetGame() {
    startController();
}
```

- ii. With this addition, you can run your program by simply constructing an instance of your Controller class as described in items (i) and (ii) under option (a). That is, your program will start to run as soon as you construct an instance of the class by selecting "new MagnetGame()" from the menu that appears when you right-click or control-click on the icon for the program's Controller class.
- 3. If you want to use BlueJ to create and experiment with objectdraw graph-

ical objects (rather than running complete programs), you will first need to create a FrameCanvas object which the graphical objects can live in. To do this:

- (a) Select "Use Library Class..." from the BlueJ "Tools" menu.
- (b) Enter "objectdraw.FrameCanvas" in the "Class:" field of the dialog box that appears and then press return (or enter).
- (c) Select the constructor that expects two ints as parameters and press OK.
- (d) Enter the width and height of the canvas you would like to create as parameters and press OK. A window displaying a blank canvas will appear shortly.

Once this canvas exists, you can use it as a parameter to construct FramedRects, Lines, or any of the other objectdraw graphical objects. To create such objects, you will need to select "Use Library Class..." from the "Tools" menu and enter the qualified name for the class (e.g. objectdraw.FramedRect) as you did to create the Framecanvas.