

Using Ambient

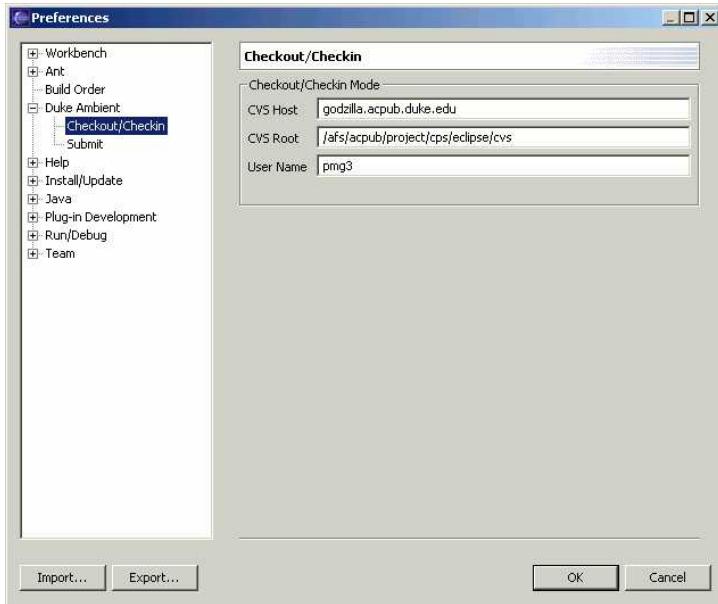
by Duke Curious 2004

preparing the environment

Before you can use the Duke Ambient environment to start working on your projects or labs, you need to make sure that all configuration settings suit your needs. If you do not verify the preference settings, it is probable that Ambient will not function correctly.

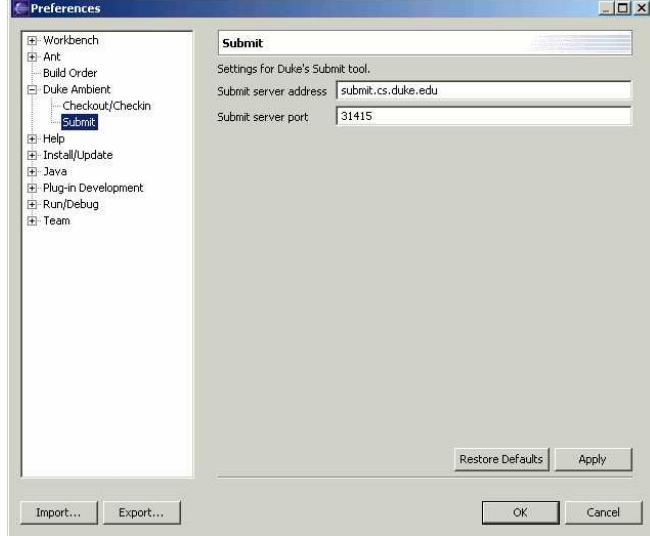
verifying the preference setting

- Open the Ambient preferences by selecting Window > Preferences > Duke Ambient > Checkout/Checkin. You should now see the following window.



- The CVS Host should be [any acpub computer].acpub.duke.edu (ie. teer27.acpub.duke.edu) and the CVS Root should be /afs/acpub/project/cps/[your course designation]/cvs. Lastly, you need to replace [your login] with your NetID.

- Next, change the Submit preferences by selecting Window > Preferences > Duke Ambient > Submit. You should now see the following window.



- The Submit server address should be submit.cs.duke.edu and the Submit server port should be 31415 (these are the default preferences).
- Click OK to save the preference settings. This will close the window. You are now ready to check out your first project.

checking out a project

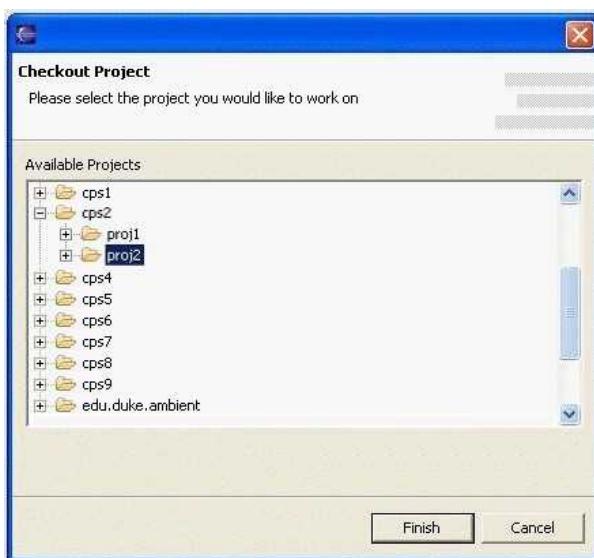
There are two ways to start working on a project. You can either create a completely new project, or you can checkout a project from the CVS server. To see if there is a project for you to check out, please follow these steps.

- Open the Ambient Checkout window by selecting Ambient > Checkout Basic Project or Ambient > Checkout Project. You may need to first change your Checkout preferences to have your login name stored. The first time you do this you will be prompted for a password.

- *NOTE* There is most certainly a difference between Checkout Project and Checkout Basic Project. The difference in Checkout Basic Project is that it will delete all previous project work on both the remote server (including your checkins) and also your local files. It will give you fresh start, but you must be sure that this is what you want because you cannot revert back. Checkout Project will bring you your latest checkin on the remote server, and if you haven't ever checked in a project then it will get the basic (beginning) project.



- Enter your NetID password and click OK.
- A new window should open that will prompt you to select a project that you wish to check out.



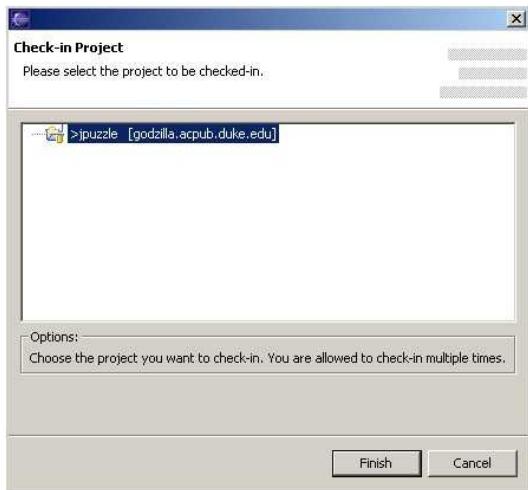
- Find your course's designated folder and navigate to the project you'd like to work on. Make your selection and click Finish. Checking out a new project may take some time depending on the speed of your computer and the network connection.
Please be patient!
- The window should disappear, and you should be returned to the Eclipse workbench, where you should notice a project in the Ambient Package Explorer.

a word of caution

- You will have to be careful if you want to work on the same project from home and at school. Let's say you start working on a project in school and submit it. Then you go home, check out the project again, work on it, and finally submit it again. So far no problem. However, if you want to keep working on the project in school the next day, you have to remember to check out the project again. This will ensure that you always work with the latest copy of your project. If you check out the project again it will ask you to overwrite the old project. Answering yes will allow you to work on the latest version.
- Before you can use the Duke Ambient environment to start working on your projects or labs, you need to make sure that all configuration settings suit your needs. If you do not verify the preference settings, it is probable that Ambient will not function correctly.
- Click OK to save the preference settings. This will close the window. You are now ready to check out your first project.

checking in a project

- You can and should check-in your files multiple times. Even if you are not done with your project, you should always check-in your project before you leave your computer. The process of checking in a project is very similar to checking out a project.
- In order to check in a project select Ambient > Check-in Project (alternatively you may press Ctrl + Alt + S). You will see all projects in your workspace that are ready to be checked in. Select one project and click Finish.



- The process will run in the background. You will be informed if your files have not been checked in successfully.

undo

- With each check-in, your project files will be tagged with the date. This lets you come back later and undo any incorrect changes you later made. Please be very careful with this function as it will remove your latest code and only let you use

the code that you decided to revert to. Use the Undo button on the Ambient toolbar for this functionality.



the ambient perspective

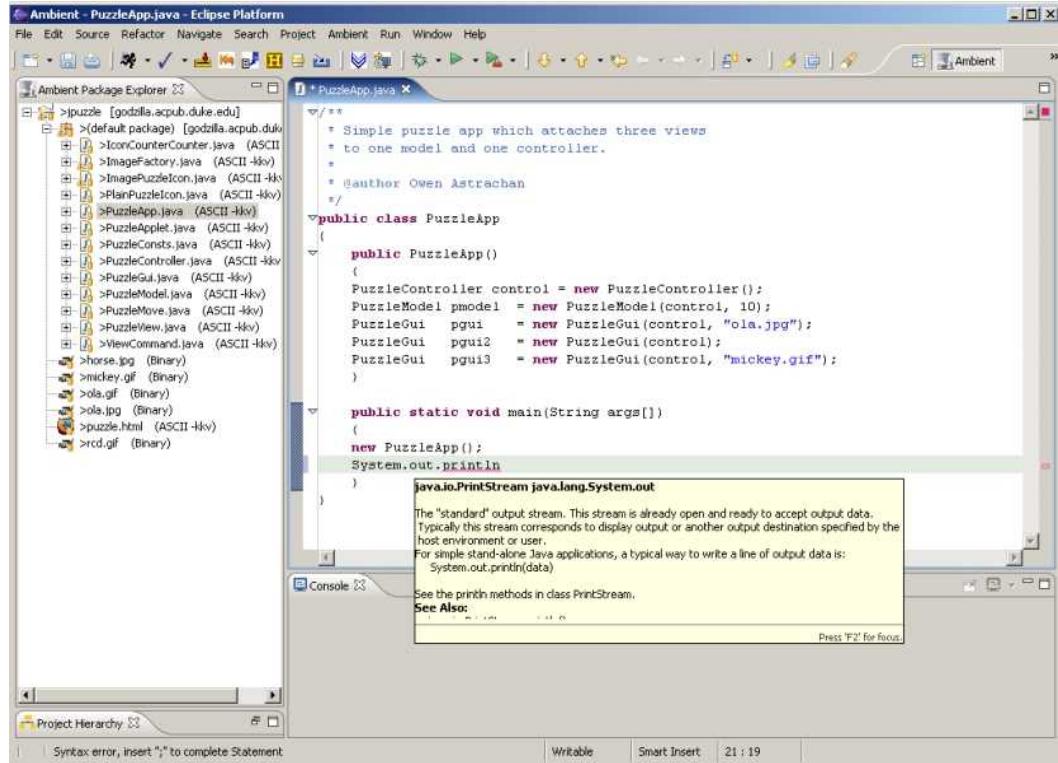
- Now that you have checked out your first project, you should familiarize yourself with the workbench. You should see the Ambient perspective (notice the name "Ambient" in the title bar). If this is not the case, you should switch to this perspective by selecting the Ambient in the perspective list.

elements of the workbench

- menubar at the top of the window
 - it includes the Ambient menu, which should become very handy
- toolbar underneath the menubar
 - it provides shortcuts to functions in the menubar that have been symbolized with icons
- Ambient Package Explorer on the left side
 - displays all projects in your workspace
- Outline view on the bottom left side
 - this will show the full structure of classes, methods, and variables in whichever file is currently selected.
- editor on the right side

console window on the bottom

- will display output and allow for input when you run programs



using the ambient package explorer

- At first, you should only see one object (probably called "project1") with a little plus sign in front of it. Click on the plus sign to expand the object and see the objects it contains. Do this to all children until the view is completely expanded. You should now see that there are some classes listed in the default package. Notice also that you can see an outline of the class. All data members and methods are displayed. In order to open a class or a specific method, simply double-click on its name.

using the java editor

A Java editor containing the file you just selected should have opened up in the editor area on the right side. Please select Ambient Perspective (alternatively you may press Ctrl + Alt + P) again. This should hide some of the menus you do not need. Using the editor is just like using any text editor. However, there are some additional features that should help you write programs.

- Syntax highlighting

By default, the editor analyzes your code and highlights certain elements to make it easier to read. For example, it highlights keywords in purple, comments in light blue or light green, and strings in blue.

- Error detection

When the editor finds an expression that it does not know how to evaluate, it will underline it with a red line. You might not be able to run a program before all errors are fixed. If the editor thinks a line is unnecessary, it will underline it in yellow.

- Quickfix

Usually, when the editor detects an error, it will also attempt to present you with a solution. To do so, the editor will display a yellow light bulb on the left edge of the line in which the error occurred. Clicking on the bulb once will open a menu that will offer you different solutions. Selecting one of these solutions will immediately alter your code (however, you can always undo the changes).

- Formatting your code

Formatting your code is not only good programming practice but will also help you determine the location of errors. Fortunately, Eclipse has a functionality that will allow you to automatically format your code. To do so select Ambient > Format Code (alternatively you may press Ctrl + Shift + F).

- Organizing Imports

Java comes with a wealth of predefined classes that you can use. To do so, you will have to import the classes so your Java compiler knows about them. Ordinarily this involves looking up where these classes are located. This is not the case in Eclipse. If you are using a class that Eclipse does not recognize, simply select Ambient > Organize Imports (alternatively you may press Ctrl + Shift + O). Eclipse will automatically import all required classes, if they exist.

- Using auto-completion features

There are two coding features that should make your life a lot easier.

- Type a keyword or part of a keyword and press Ctrl + Space. Eclipse will show you a list of elements that you may wish to use. For example, if you type "for" and press the two keys simultaneously, it will show you a list of predefined for loops. Selecting one of them will insert the loop into your code.

- When using member methods or data members of classes, we use the dot operator in Java (e.g., MyFirstClass.printHelloWorld()). Oftentimes we know that there is a method we would like to use, but we cannot remember its exact name. If you are not sure, simply type the name of the object you are working with followed by a period and wait for a moment. A list will appear that shows all members that you may use.

running a project with the ambient perspective

- Before submitting a project as final, you should always make sure it runs. For a project to be able to run, at least one of the classes will have to have a main method. Now all you have to do is select the dropdown menu with the icon  and choose Run Application (or whichever type of program you are working on). Depending on the type of the application, you will either see a new window pop up, or you will see some output in the console window (bottom right).

console window

- The console window will show you all output generated by your program. If you double-click on the console window title bar, you will maximize the size of the window. Another double-click will return it to its original size. Moreover, the window will also allow you to input information. Simply click in the window and start typing. When the program is finished, it will display the word terminated in the window title bar. If your program is stuck or you would like to terminate it for another reason, you can right-click in the console window and select terminate.

multiple main methods

- You may have multiple classes with main methods in the same project. Java can only run one main method at a time. If you select a project that contains multiple main methods, you will be prompted to select which one you would like to run.

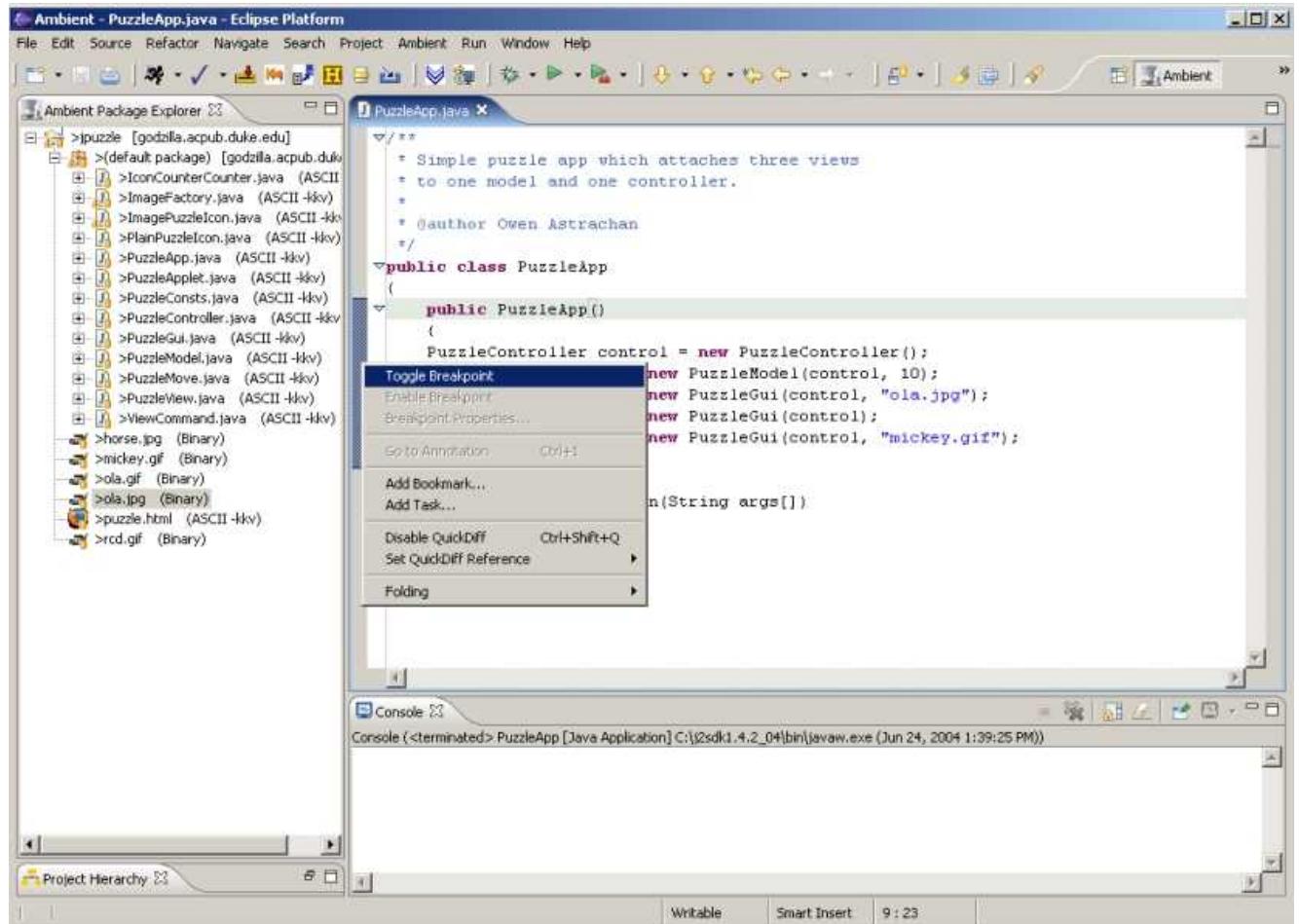
debugging a project

- Oftentimes programs will not function as expected because the programmer possibly forgot to initialize a variable, made a mistake setting conditions such that a loop will never terminate, or did not account for a method call that returns null. Fixing these errors takes very little time, but finding the error in the code can be a very difficult task. The concept of debugging aims at speeding up the locating process of programming flaws.
- Debugging allows the programmer to set break points in the code. When executing the program in debug mode, Eclipse will run the program normally, but it will pause (not stop) execution at the first break point. The programmer can then observe the state of all currently declared variables. Moreover, the programmer can step through the code to see how the next instructions will affect the variables. Execution can be resumed and the program will either terminate normally, or Eclipse will stop at the next break point it detects.

setting break points

- In order to set a break point, simply right click on the gray margin to the left of the line of code in the editor. A small context menu should open, which will allow you to add a break point. To start the debugging process select the dropdown

menu with the icon  and choose Debug Application (or whichever type of program you are working on).



The De bugging perspective

Once you have started the debugging process, Eclipse will switch to the debugging perspective. This perspective may seem complex and confusing at the beginning, but this should not discourage you from using it. Here are the key elements you should be familiar with:

- Debug view

The most interesting part of the debug view is the toolbar (marked in blue).

1. Resume - will result in normal execution of the program until the next break point
2. Suspend - will pause execution and allow you to view state of variables
3. Terminate - will terminate execution of the program
4. Step Into - if you are at a method call (e.g., Math.pow(200)) and you would like to know what the method does you should use Step Into
5. Step Over - if you only care about what a method will return or how it will change your variables use Step Over
6. Step Return - will finish the method you are currently in and return to the point the method was called from

These six buttons will allow you to navigate the program execution while you can observe what values methods return and how they affect your variables.

- Variables view (marked in blue on the right side)

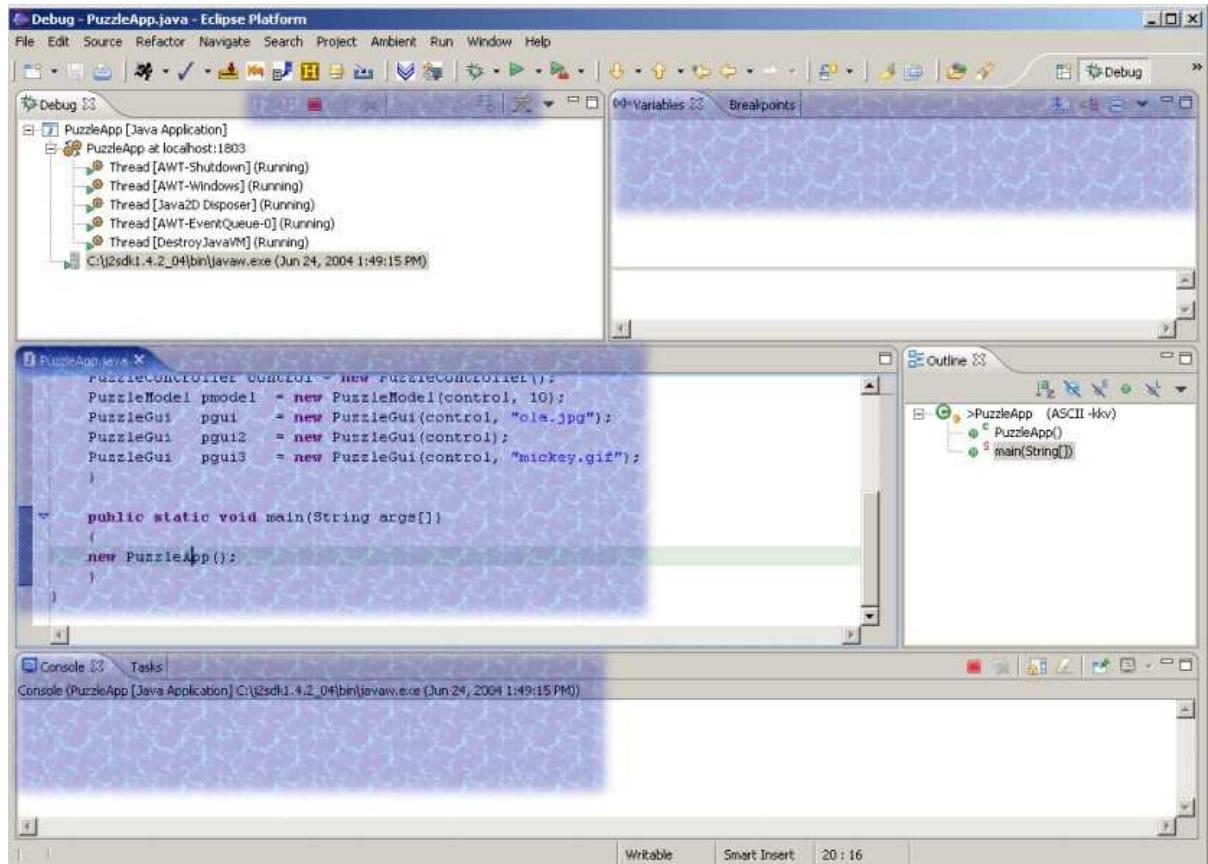
This view simply shows all variables that are currently defined in the context of your program.

- Editor view

You will see the program line the debugger is currently at. If the execution of your program takes you to a different class, Eclipse will open up the corresponding .java file automatically.

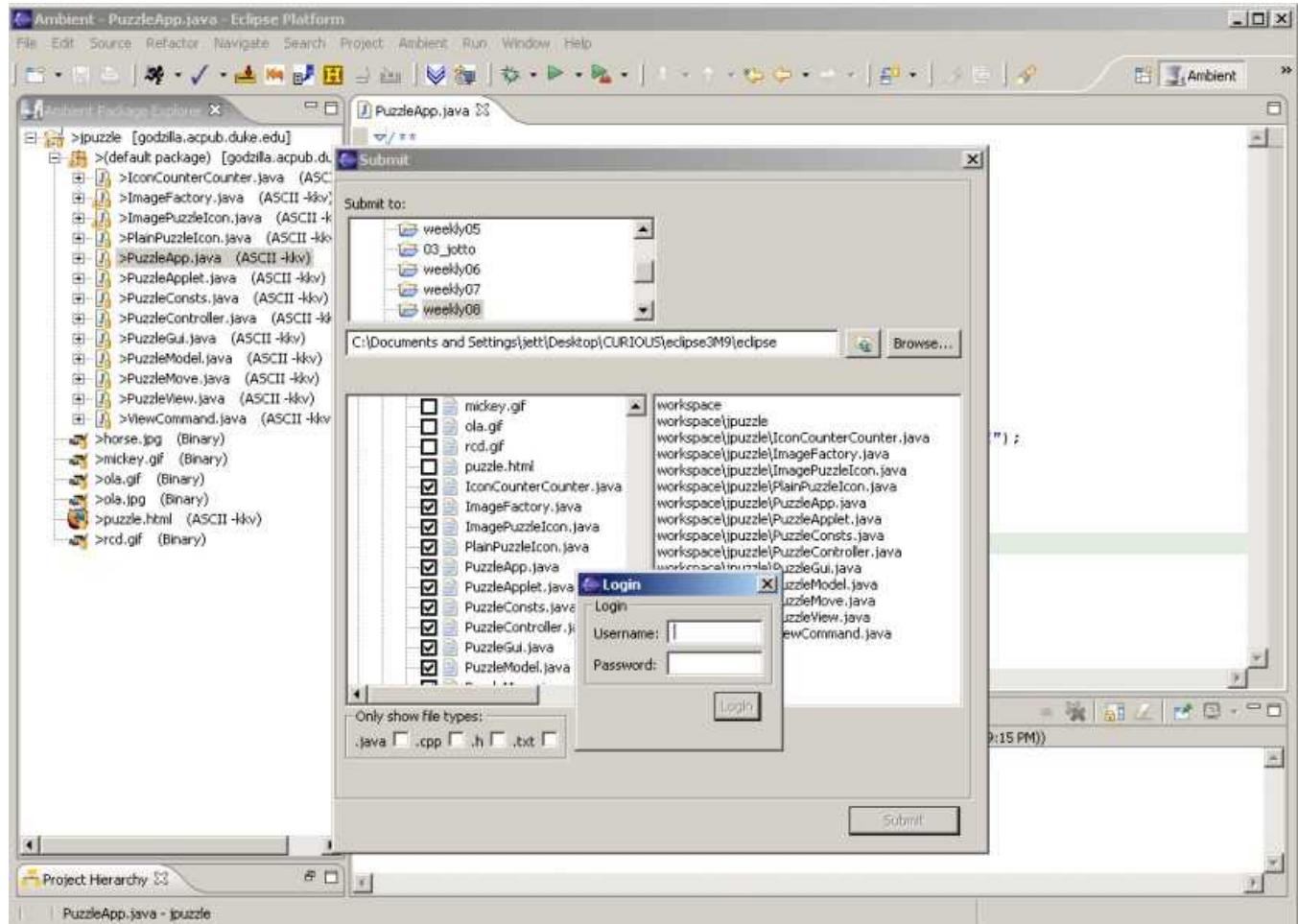
- Console view (bottom blue rectangle)

Much like the console view in the Ambient perspective it will show output of the program that's running and allow for user input.



submitting a project

- Assignment submission through Eclipse not only simplifies the grading process for the professor, TA, and student, but it also lets you to check the history of submissions and allows you to submit a project from your dorm room or any lab computer at any hour of the day.
- Submitting allows the programmer to digitally send in a programming assignment to be graded. In order to submit an assignment stored as a Project folder, find the button Ambient > Submit Project for Grading. After choosing that, you will be faced with a dialog asking which assignment you'd like to submit.



- Navigate through the tree of available courses and assignments in the upper frame of the window and find which one fits your code submission. Next, navigate to your project files in the lower left frame of the window to find your project folder. Select which files you will submit to be graded. These files will be listed in the frame on the right side of the window. Once all files to be submitted have been chosen and are thus listed on the right side, click the Submit button.
- You will be asked to enter your NetID and password before your files are submitted, and after a successful submission, you will always be able to check back on your submission history with the Submit History button.
- Before submitting a project as final, you should always make sure it runs. For a project to be able to run, at least one of the classes will have to have a main method. Now all you have to do is select the dropdown menu with the icon  and choose Run Application (or whichever type of program you are working on). Depending on the type of the application, you will either see a new window pop up, or you will see some output in the console window (bottom right).