

Java AP - Introduction To Classes

Assignment #23

Copyright Notice

Copyright © 2013 DigiPen (USA) Corp. and its owners. All rights reserved.

No parts of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language without the express written permission of DigiPen (USA) Corp., 9931 Willows Road NE, Redmond, WA 98052

Trademarks

DigiPen® is a registered trademark of DigiPen (USA) Corp.

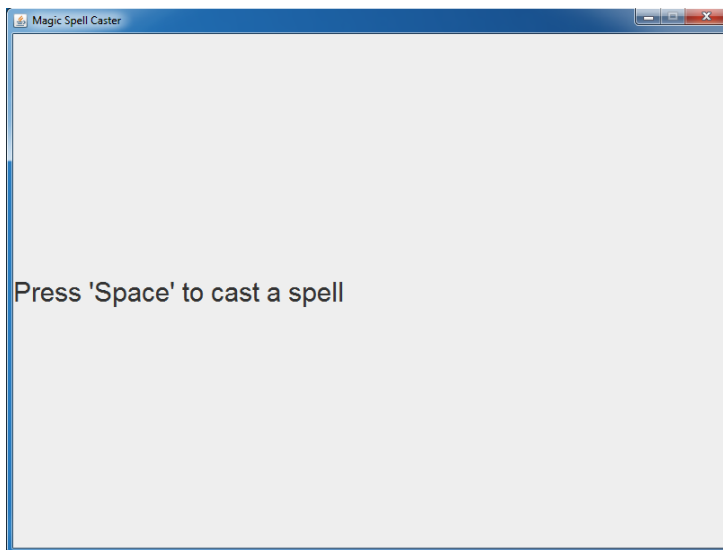
All other product names mentioned in this booklet are trademarks or registered trademarks of their respective companies and are hereby acknowledged.

This is an assignment to help you practice with classes. There are three classes for you to write to complete the assignment.

In this assignment, you will implement a very simple game called "Magic Spell Caster". You have to write all the classes and methods needed. In this document I will guide you by specifying the methods that you need to implement with a small description.

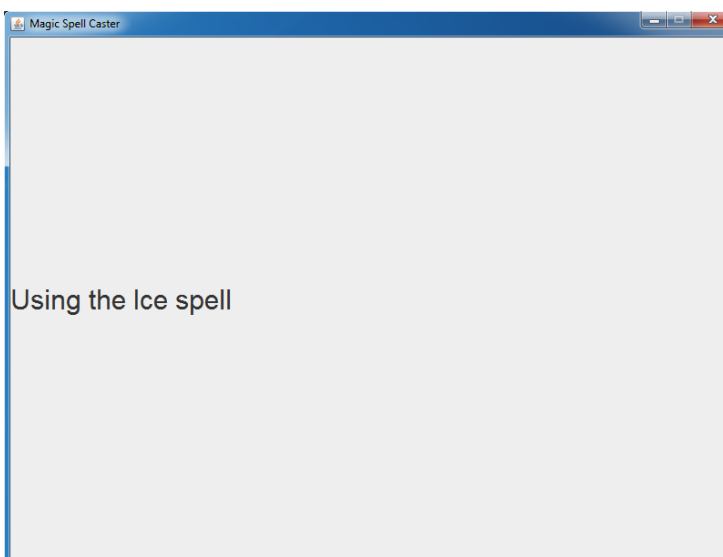
Magic Spell Caster Details:

The game starts with the following screen showing the player that pressing the space key will cast a spell.

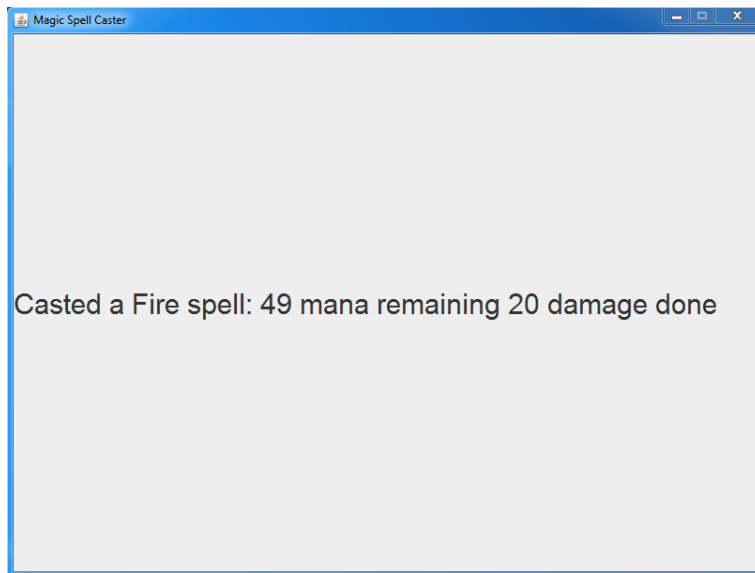


The player has 3 spells in total: "Fire", "Ice" and "Thunder". You use the "N" key to switch to the next spell.

Example: Pressing "N" will switch to the "Ice" spell and you will see the following:

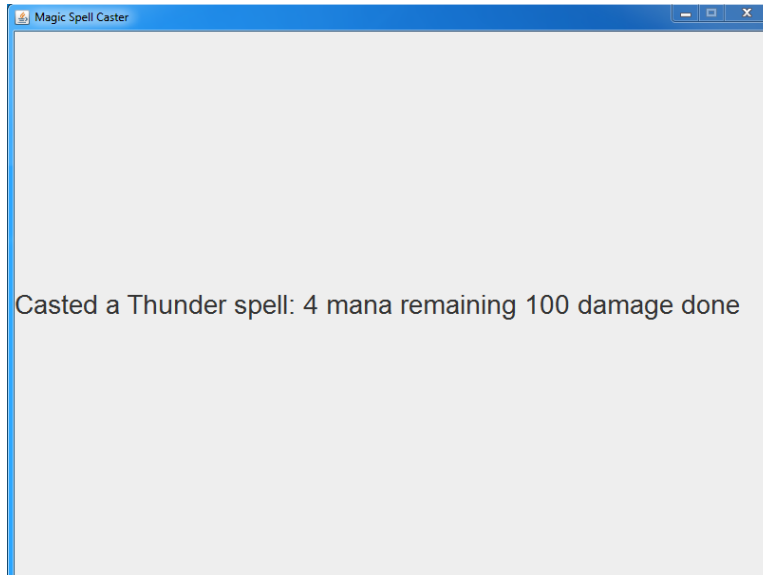


Pressing the "Space" key in other hand will cast a spell from the current spell you are using.
Eg: If you are using the fire spell and you press "Space" the following will happen:

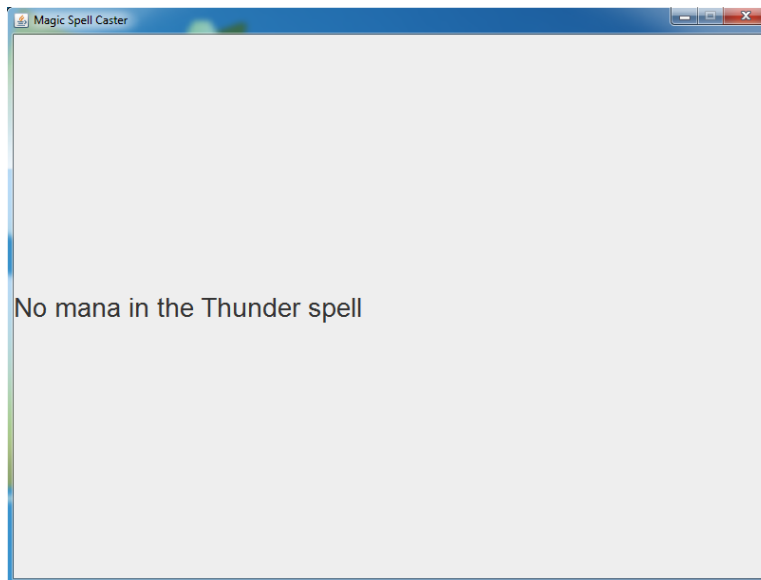


The text above is showing us which spell we are casting, how much mana we still have and how much damage that spell does.

Another example but casting with the "Thunder" spell this time:



If a spell doesn't have any more mana the following message should be displayed when the user tries to cast the spell:



Code Details:

In this assignment, I will guide you with the architecture since it is your first time using classes.

Class Name	Attribute Name	Accessibility	Description
<i>Hero</i>	<i>currentSpellIndex</i>	<i>private</i>	<i>Integer</i> This integer is used as an index that keeps track which spell we are currently using.
<i>Hero</i>	<i>spells</i>	<i>private</i>	<i>Array</i> This array will contain all the spells objects (for us it will be 3, "Fire", "Ice" and "Thunder").
<i>Hero</i>	<i>Constructor</i>	<i>public</i>	<i>Method</i> The constructor won't take any parameters. Its job will be, other than it creates an instance of the Hero Class, to initialize the currentSpellIndex and the spells array by putting the 3 spells in it.

<i>Hero</i>	<i>SwitchToNextSpell</i>	<i>public</i>	<p><i>Method</i></p> <p>This method won't take any parameters but will return a String. Its job is to switch to the next spell by positioning the currentSpellIndex at the right index. The returned string will contain the message that will inform the user about the new spell he/she is using.</p>
<i>Hero</i>	<i>CastSpell</i>	<i>public</i>	<p><i>Method</i></p> <p>This function won't take any parameters but will return a String. Its job is to cast one spell from the currently used spell and update the weapon's mana variable accordingly. The returned string will contain the message that will inform the user about the spell's stats (Check the provided solution to get the right information in the string).</p>
<i>Hero</i>	<i>Destroy</i>	<i>public</i>	<p><i>Method</i></p> <p>This function won't take any parameters and won't return anything either. Its job is to cleanly clear all the memory allocated when a hero instance got created.</p>
<i>MagicSpell</i>	<i>name</i>	<i>public</i>	<p><i>String</i></p> <p>This string is used to give the spell a name.</p>
<i>MagicSpell</i>	<i>mana</i>	<i>public</i>	<p><i>Integer</i></p> <p>This integer is used as a mana counter.</p>
<i>MagicSpell</i>	<i>damage</i>	<i>public</i>	<p><i>Number</i></p> <p>This number represents how much damage the spell does to enemies.</p>
<i>MagicSpell</i>	<i>Constructor</i>	<i>public</i>	<p><i>Method</i></p> <p>The constructor will take 3 parameters (String, Integer and Integer) that will be used to initialize its member variables. These parameters will help the user initialize the spell instance's members at creation time.</p>
<i>Level</i>	<i>Update</i>	<i>public</i>	<p><i>Hero Class</i></p> <p>This method checks for key presses and reacts accordingly. If the space key is triggered the hero's CastSpell metod should be called. If the N key is triggered the hero's SwitchToNextSpell method should be called. In both cases the text on the screen should be updated.</p>

Note:

- Updating the text on the screen is done by calling the **onScreenText** instance's **setText** method which takes any String value and will change the text label on the screen.

Example: `onScreenText.setText("Hello");` -->displays Hello on the screen.

- You have to create the **Hero** and **MagicSpell** classes from scratch.
- Both the **Hero** and **MagicSpell** classes have to fall under the **gameplay** package.

What to submit

You must submit the full Java project (**MagicSpellCaster**) containing all the java files in a single .zip file named "**Assignment23.zip**" (go to the class page on moodle and you will find the assignment submit link). **Do not submit any other files than the ones listed.**

Special note:

The due date/time posted is the positively latest you are allowed to submit your code. Since the assignment can easily be completed well before the deadline, you should strive to turn it in as early as possible. If you wait until the deadline, and you encounter unforeseen circumstances (like being sick, or your car breaking down, or something else), you may not have any way to submit the assignment on time. **Moral: Don't wait until the last day to do your homework.**