

Forum:[Hack Forums](#)

Thread: [The Empire | Java Game | Tutorial #1 - The Basics of Game Development in Java](#)

Canonsx130 - September 21, 2013, 9:41 am

This tutorial is aimed at programmers who can understand and interpret basic java codes, construct basic programs and have a very fluent and solid idea of all the basic concepts of java. If you are not one of them, it is highly recommended you take bit of your time to learn the basics and then look at this thread. It is expected that you understand the code as you take it down and not just write it all down, compile and have your game. I'll try and explain every single line as detailed as possible. We will start out with the basics and by the end, create couple of very basic java games.

Before starting to create the actual game, make sure you have the workplace setup. I use [Eclipse](#) and it is preferred you use it too. However it is upto you to decide any IDE you are comfortable with. Create the project and create a class giving whatever name you like. And you should be somewhere around here right now:

Now that you have made the class, quickly type this out on Eclipse. Yeah this is the first piece of code you are ever going to write for making the game! Don't worry, I'll be explaining everything. And yeah, I'm a sneaky bastard, I didn't want you to copy paste everything so I decided to take a screenshot of the code ?

Code public class Prototype extends JFrame {

Now what the hell is this line. Well what we did is first create the class giving the name and then we extended JFrame. So what is extends and what is JFrame. Well JFrame is another class, like our Prototype class. It exists already and we just do not see it. Now using extends JFrame will allow our poor class named Prototype to use the functions and use any other functionality the JFrame class had, in our class. So yeah, it kinda helps a lot of time of you from creating a new JFrame class.

Code public Prototype() {

What we just made with the above line is called a 'constructor'. As you might know already(yeah you're smart), constructor is more like a function, but that anything inside it is automatically initialized during the run. So you start the program and BAM, things inside it gets initiated. Just know that anything inside this stuff, gets automatically assigned when the program is run. There is nothing external to be done on it to wake up or apply the values to our program. That's a simple enough explanation which will suffice for now.

Code add(new BoardClass());

Now BoardClass is another class that we will make in a moment. BoardClass is going to be a class essentially like a board, or a panel to display our actual contents. And using the add keyword will set the BoardClass to the center of our JFrame component. So we have the JFrame and the board or the panel is set to the center of it to display anything which is needed.

Code setDefaultCloseOperation(EXIT_ON_CLOSE);

This is a vital line of code you'll probably use in every GUI program you make. This code will allow you to close the program using the X button in the top right corner. Else, the process will just sit there and you will not be able to close it out. So yeah, using this line of code will set the default operation on clicking the close button to exit out from the running program.

Code `setTitle("ProtoType");`

This is quite easy and I assume you have guessed what this does already. As easy and explanatory it is, it simply sets the title of our window to ProtoType. Just like you are seeing the thread title on your browser tab right now, whatever title we specify will be displayed on top of the window.

Code `setLocationRelativeTo(null);`

This is another simple line of code. Using this line, we simply center the window with the screen. You do not want the window to appear at one corner and annoy you right? Yeah, so we use this line of code to make sure when you run the program, the window appears in the dead center of your window, clean and neat!

Code `setSize(500, 260);`

Come on, use your brain kids! 10 points to anyone who can guess what this does. Yeah! Well done. You got it damn right. This line simply sets the size of our window to 500px in length and 260 in height. You can obviously change this value to any value you want depending on the program you want to make. It does not actually matter what size you choose as of now since we have not started to make the actual game.

Code `setVisible(true);`

Without this line of code, your entire code is pretty useless. This code displays the actual window on screen. If you set the visibility of the window to false, you'll obviously see nothing when you run the program. So make sure you never forget this line of code while writing a GUI-based program or nothing will appear on your screen. To repeat again, this simply makes the window visible on your screen.

Code `setResizable(false);`

Another self-explanatory line of code. This line of code simply makes the window you made non-resizable. Unless you want others to use their mouse to punish your window by scrolling and disfiguring it, you should probably use this little function. So setting it to false will make sure none of those cruel people can come out with their mouses and torture your little game window by deforming it.

Code `public static void main(String[] args) {`

This is probably like the first thing you learned in java, the main function. The main function runs first wherever it is within the program. Without the main function, the program will not run.

Code `new Prototype();`

And finally we simply create an object for the class using the new keyword. Notice that we are creating the object explicitly, hence the usage of the keyword new.

Like we've said earlier, now we will need to create the BoardClass class. Why we are doing this? to display our actual stuff. This class will act as a panel to display anything we need. When we go about creating the game, it will happen inside this board, just like any other board, say for a chess game. Without this board where the game takes place, you can't do anything. So now we'll see the code first and then the explanation. First start off by creating a new class in the same package and

name it BoardClass.

Code import javax.swing.JPanel;

JPanel like we have told is a container to hold our actual game. So like we imported JFrame earlier, we have imported JPanel into our class from the javax package so that we can use it for good use. Please note that JPanel is inside javax, within the swing package and hence we included swing in between.

Code public class BoardClass extends JPanel {

We simply defined the class with the proper name and extended the JPanel class so that we'll be able to use the JPanel functionalities in our class, if needed.

Code public BoardClass() {

Again, this is a constructor. Notice that constructors will always have the same name of the class it is in. In this case since there is nothing else to initialize, there is no need to enter anything inside the constructor.

With this, we conclude our first tutorial. In the next tutorial, we will display basic shapes and images onto our panel. If you liked this thread, please do leave comments which are highly appreciated. Now it's my favorite part. Let's make sure all the hardwork we did pays off. Do not wait another second, run down the cliff and click the run button to see the wonderful piece of art, a true souvenir, your first ever Java GUI ?

MrRapidz - September 21, 2013, 9:44 am

Very nice tutorial, are you gonna add more parts to this? :)

Phaxol - September 21, 2013, 9:46 am

Pretty basic, I actually find this ironic because I just picked up Java again yesterday and am looking into LibGDX. Is this tutorial going to use that by any chance?

Canonsx130 - September 21, 2013, 9:46 am

Yes, take some time to read, you'll see that I mentioned the next parts coming soon enough, probably tomorrow or the day after. By the end I'll be aiming at teaching everyone to make basic games, like Snake, Tetris or Space invader etc.

@Phaxol: No, since this is the pure basics, I'll stick to pure java code and will not use any extra libraries.

Sparta. - September 21, 2013, 9:56 am

Not really the basics of game development. You should go over and explain the game structure of a game first, not just show how to make a JFrame/JPanel as the first tutorial.

Canonsx130 - September 21, 2013, 9:58 am

(Sep 21, 2013 09:56 AM)Sparta. Wrote: [\[CITE\]](#) Not really the basics of game development. You should go over and explain the game structure of a game first, not just show how to make a JFrame/JPanel as the first tutorial.

This tutorial was more aimed at people having no idea about Java GUI or the swing package. I thought it would not hurt if I chirped in a tutorial the very beginning showing how to display a basic

window and explaining a bit of JFrame functions before starting to do the actual code. Thanks for your feedback though. I appreciate constructive criticism. I'm sure you'll like me coming tutorials :)

Bonfire - September 21, 2013, 10:01 am

All you did was convert this tutorial to text

<http://www.youtube.com/watch?v=PopdTUzizDA>

I mean, I'm sure that text better for some people, and that this is the simplest way to do it, but in-case anyone wanted a video tutorial, that would be it.

Swirl - September 21, 2013, 10:03 am

(Sep 21, 2013 10:01 AM)Bonfire. Wrote: [\[CITE\]](#) All you did was convert this tutorial to text

<http://www.youtube.com/watch?v=PopdTUzizDA>

I mean, I'm sure that text better for some people, and that this is the simplest way to do it, but in-case anyone wanted a video tutorial, that would be it.

That's actually plagiarizing, he didn't cite his sources in the slightest. I think somebody should tell xadamxdk about this.

Canonsx130 - September 21, 2013, 10:04 am

(Sep 21, 2013 10:01 AM)Bonfire. Wrote: [\[CITE\]](#) All you did was convert this tutorial to text

<http://www.youtube.com/watch?v=PopdTUzizDA>

I mean, I'm sure that text better for some people, and that this is the simplest way to do it, but in-case anyone wanted a video tutorial, that would be it.

Well you could say so but you see, this is the base code you use to create a base window and set some common properties to it so you'll find atleast 100+ links where 99% of the lines are present. So there are not much changes to make in it and I'm not surprised to see this video either. And yeah, some people might like it in video format a bit more. There is absolutely no other way you could display a window. It's like saying 'hey your syntax for creating the main function is similar to mine' to make it more generic.

DarkShadowz - September 21, 2013, 10:12 am

(Sep 21, 2013 10:04 AM)Canonsx130 Wrote: [\[CITE\]](#) Well you could say so but you see, this is the base code you use to create a base window and set some common properties to it so you'll find atleast 100+ links where 99% of the lines are present. So there are not much changes to make in it and I'm not surprised to see this video either. And yeah, some people might like it in video format a bit more. There is absolutely no other way you could display a window. It's like saying 'hey your syntax for creating the main function is similar to mine' to make it more generic.

There definitely is the method of making a window by initiating one through LWJGL, rather than starting off with the slow and limited renderer given by Java. To be honest, it would be preferable to start off the whole game development series off hardware-accelerated graphic platforms so that it would become an ease in practice later on.

Bonfire - September 21, 2013, 10:13 am

(Sep 21, 2013 10:04 AM)Canonsx130 Wrote: [\[CITE\]](#) Well you could say so but you see, this is the base code you use to create a base window and set some common properties to it so you'll find atleast 100+ links where 99% of the lines are present. So there are not much changes to make in it and I'm not surprised to see this video either. And yeah, some people might like it in video format a bit more. There is absolutely no other way you could display a window. It's like saying 'hey your syntax for creating the main function is similar to mine' to make it more generic.

Hence why I said "and that this is the simplest way to do it."

I understand that there might not be other ways to do it.

I don't want to throw this thread off-topic though.
Nice tutorial.

Canonsx130 - September 21, 2013, 10:14 am

(Sep 21, 2013 10:12 AM)DarkShadowz Wrote: [\[CITE\]](#) There definitely is the method of making a window by initiating one through LWJGL, rather than starting off with the slow and limited renderer given by Java. To be honest, it would be preferable to start off the whole game development series off hardware-accelerated graphic platforms so that it would become an ease in practice later on.

But then that's not how I planned it for my tutorial. I agree LWJGL is much more efficient but this is the core basics. I really feel like I should stick to just java for now. Maybe after I finish this small series and complete couple of games like snake or tetris, I'll start a LWJGL series which might interest even more members :)

_wind - September 21, 2013, 10:36 am

Very nice, detailed for starters wanting to make games in Java.

Sparta. - September 21, 2013, 1:38 pm

(Sep 21, 2013 10:01 AM)Bonfire. Wrote: [\[CITE\]](#) All you did was convert this tutorial to text <http://www.youtube.com/watch?v=PopdTUzizDA>

I mean, I'm sure that text better for some people, and that this is the simplest way to do it, but in-case anyone wanted a video tutorial, that would be it.

...that's not even close to the tutorial. That is the proper way to make a JFrame, so why the fuck is it a big deal? Just because it's on someone else's tutorial doesn't mean he copied it, even I know that people usually use JFrame like that.

(Sep 21, 2013 10:03 AM)Swirl Wrote: [\[CITE\]](#) That's actually plagiarizing, he didn't cite his sources in the slightest. I think somebody should tell xadamxdk about this.

He didn't copy it, stop making up bull shit. You make me want to prove everyone how much of a fucking idiot you are. ?

Bonfire - September 21, 2013, 2:48 pm

(Sep 21, 2013 01:38 PM)Sparta. Wrote: [\[CITE\]](#) ...that's not even close to the tutorial. That is the proper way to make a JFrame, so why the fuck is it a big deal? Just because it's on someone else's tutorial doesn't mean he copied it, even I know that people usually use JFrame like that.

Where did I say he copied it? I simply said he converted it to text, then thanked him for the nice tutorial. Now stop derailing this thread.

Sparta. - September 21, 2013, 3:57 pm

(Sep 21, 2013 02:48 PM)Bonfire. Wrote: [\[CITE\]](#) Where did I say he copied it? I simply said he converted it to text, then thanked him for the nice tutorial. Now stop derailing this thread.
?

"Derailing"

Who was the one who bitched about how he converted a "video into text". You're pathetic, please.

OP: I suggest making a few more and combining them into one thread.

Canonsx130 - September 21, 2013, 6:13 pm

Thank you sparta. for your concerns. I felt the same way on seeing the reply by swirl but I guess it's alright. After all, I know I was right in making this tutorial so I just felt like not replying to such

comments. And yes, like I said, I'll be continuing this series and by the end, making proper working games(basic), like Tetris, snake etc. Thanks :)

Han. - September 22, 2013, 3:24 am

(Sep 21, 2013 10:03 AM)Swirl Wrote: [\[CITE\]](#) That's actually plagiarizing, he didn't cite his sources in the slightest. I think somebody should tell xadamxdk about this.

Let me give you a little info on how programming works, because you seem to be confused about the definition of plagiarism.

Generally, tens of thousands of people have the same ways of doing things, so just because Canon made a window the same as some other guy, doesn't mean he plagiarised, it means he programmed a window in a way that happened to be very common. Get it?

Nice tutorial Canon <3

Annointed - September 24, 2013, 2:40 pm

Hope you continue this, I'm really interested in these threads. Thanks!

Canonsx130 - September 25, 2013, 5:51 am

Thank you annointed for your comment. I'll definitely start working on hte part 2 and 3 of this one ASAP and post it. Glad you're enjoying this :)

D?rkStorm - September 25, 2013, 1:36 pm

If you were going to focus on the rendering side of game development in Java, you could have at least explained canvases with double-buffering or OpenGL...

leto12 - December 18, 2013, 9:57 am

Thanks a lot for this tutorial.

I will wait for you next tutorial ;)