A Game PROJECT ON

DX BALL

GUIED BY

**SANJIDA KHATUN**

**AFSANA AHMED MUNIA**

SUBMITTED BY

Shoumik Das Bibon

Naimul Mukit

Arefin Naveed



Submitted to

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

YEAR

2012-2013



**The Game**

**Introduction**

* **DX-ball is a 2D game . It is a game on pc.**
* **Such kind of games like DX-ball are played for having fun and as a part of recreation.**
* **It can be played when we are free and pass our leisure time.**
* **It is very easy to play. Just only mouse and sometimes keyboards are needed to play it.**
* **Dx-ball is a nice game to pass the leisure times.**
* **People of all ages and Everyone can Play as it is easy to play.**
* **Today the Dx-ball Game is on Mobile too. So We can enjoy it on Mobiles and Have a lot fun . At anywhere, any place.**
* **After the Creation it Had advanced A lot.**
* **A lot of Game Developers Has made Different Approach making this game.**
* **It has Reached Everyones on to Reach. And Day by day it is Advancing.**
* **Gradually its interests and Demands are being noticed in the marketplaces, And even if in the Internets**
* **A pc version was made Firstly. But later on the game developers Thought that it should be Spread on the intenets.**
* **They Uploaded this interesting game in the internet and Later on this process Elaborated.**
* **It is now Found majority sites in the Internets. People Download them and play then and have fun.**
* **So necessity of CD’s And DVD’s are decreased after this.**
* **Online gaming is a familiar term . Dxball has become a online game too for all the people. Everyone can play it online too.**
* **It is a nice game indeed. Espacially Children have a lot of fun playing it.**
* **Different Computer Languages like C,Java,Python,c++ was used to make Dxball.**
* **In Our project we have Used Java Language to make this game.**
* **By Using Java a full Dxball game can be made through.**
* **As we did in Our Game creating Project.**
* **Hope Everybody Would like it and will share there positive comments.**

****

**Game History**

***DX-Ball*** is a [freeware](http://en.wikipedia.org/wiki/Freeware) [computer game](http://en.wikipedia.org/wiki/Computer_game) for the [PC](http://en.wikipedia.org/wiki/IBM_PC) first released in 1996 by [Michael P. Welch](http://en.wikipedia.org/w/index.php?title=Michael_P._Welch&action=edit&redlink=1).

The game, an updated version of an earlier series of [Amiga](http://en.wikipedia.org/wiki/Amiga) games known as

*Megaball*, is patterned after classic ball-and-paddle arcade games such as [*Breakout*](http://en.wikipedia.org/wiki/Breakout_%28arcade_game%29)

and [*Arkanoid*](http://en.wikipedia.org/wiki/Arkanoid). It became a massive [cult classic](http://en.wikipedia.org/wiki/Cult_classic) in the [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) freeware gaming

community during the late 1990s. A [level editor](http://en.wikipedia.org/wiki/Level_editor) was made available as well.

*DX-Ball*

[**Developer(s)**](http://en.wikipedia.org/wiki/Video_game_developer) [Michael P. Welch](http://en.wikipedia.org/w/index.php?title=Michael_P._Welch&action=edit&redlink=1)

[**Designer(s)**](http://en.wikipedia.org/wiki/Game_designer) [Michael P. Welch](http://en.wikipedia.org/w/index.php?title=Michael_P._Welch&action=edit&redlink=1)

**Version** 1.09 (December 1998)

[**Platform(s)**](http://en.wikipedia.org/wiki/Computing_platform) [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows)

**Release date(s)** [1996](http://en.wikipedia.org/wiki/1996_in_video_gaming)

[**Genre(s)**](http://en.wikipedia.org/wiki/Video_game_genres) [Arcade](http://en.wikipedia.org/wiki/Arcade_game)

**Mode(s)** [Single player](http://en.wikipedia.org/wiki/Single_player)

*DX-Ball* [**Developer(s)**](http://en.wikipedia.org/wiki/Video_game_developer) [Michael P. Welch](http://en.wikipedia.org/w/index.php?title=Michael_P._Welch&action=edit&redlink=1)

[**Designer(s)**](http://en.wikipedia.org/wiki/Game_designer) [Michael P. Welch](http://en.wikipedia.org/w/index.php?title=Michael_P._Welch&action=edit&redlink=1)

**Version** 1.09 (December 1998)

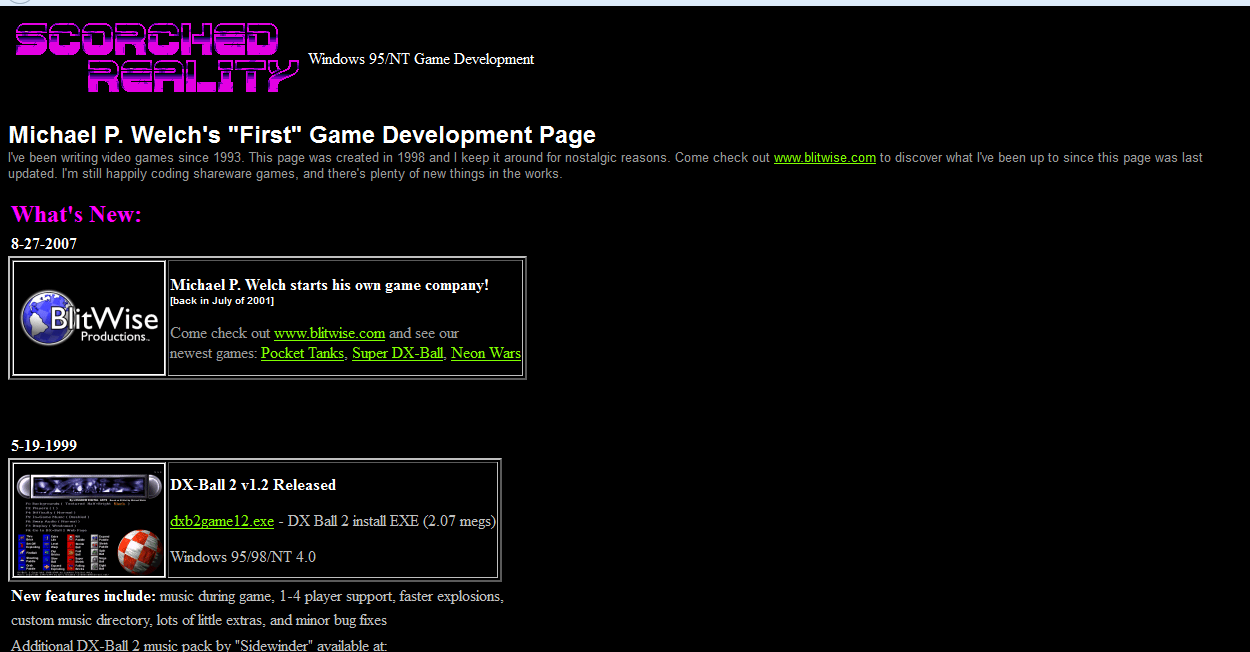
[**Platform(s)**](http://en.wikipedia.org/wiki/Computing_platform) [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows)

**Release date(s)** [1996](http://en.wikipedia.org/wiki/1996_in_video_gaming)

[**Genre(s)**](http://en.wikipedia.org/wiki/Video_game_genres) [Arcade](http://en.wikipedia.org/wiki/Arcade_game)

**Mode(s)** [Single player](http://en.wikipedia.org/wiki/Single_player)[**Michael P. Welch**](http://en.wikipedia.org/w/index.php?title=Michael_P._Welch&action=edit&redlink=1)**’s . Dxball Game ScreenShots**

As we Know that the Inventer of the Dx-ball game is [**Michael P. Welch**](http://en.wikipedia.org/w/index.php?title=Michael_P._Welch&action=edit&redlink=1)**.**





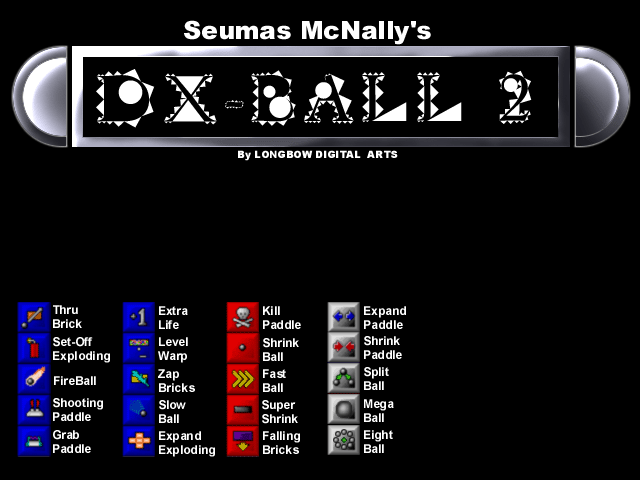






Later on Seumas Macnally Created his game Dx-Ball after **Michael p welch**

**Seumas Macnally’s . Dxball Game ScreenShot**



**Game Play**

* The game is basically a [Breakout clone](http://en.wikipedia.org/wiki/Breakout_clone):
* you bounce a ball off a paddle at the bottom hitting different color blocks on the top of the screen.
* Hitting all the blocks results in completing the level and going to the next.
* There are 50 levels to complete.
* Unlike Breakout, however, is the inclusion of powerups other than extra balls.
* When you hit a brick, there is a chance that a powerup will float downwards towards the paddle, and can be picked up by touching it with the paddle.
* Certain powerups have positive effects, while others have negative, making it important to try to collect the beneficial powerups while avoiding the detrimental powerups.
* There are 18 power-ups, four of them being either good or bad.
* For example, if you were to get the fast ball power-up.
* the ball will simply gain speed.
* The Ball also gradually picks up speed as the game continues.
* Some good power-ups include the Zap Brick which reveals the hidden bricks and also makes the unbreakable bricks breakable, the grab paddle, which is able to catch the ball and aim it at any brick that you want to hit, and also the Extra Life, whose powers give you an extra life at the cost of most of your power-ups, and Level Warp, which lets you advance to the next level.

**SeQels**

*DX-Ball* has been followed up by two direct sequels: [*DX-Ball 2*](http://en.wikipedia.org/wiki/DX-Ball_2) by Longbow Digital Arts in 1998, and

[*Super DX-Ball*](http://en.wikipedia.org/wiki/Super_DX-Ball) by Michael P. Welch himself in 2004. While *DX-Ball 2* was later succeeded by *Rival Ball*

in 2001, the latter merely relates as a spiritual successor to the original *DX-Ball*, being developed on

the branch of Longbow Digital Arts. Unlike *DX-Ball*, these games are not freeware.

**External links**

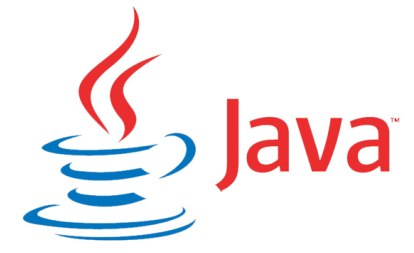
* [Homepage of *DX-Ball*](http://www.blitwise.com/Scorched_Reality/index.html) - The game and level editor can be legally downloaded here.
* [DX-Ball Online](http://dx-ball.ru/) - web version.
* [*DX-Ball*](http://www.mobygames.com/game/dx-ball) at [MobyGames](http://en.wikipedia.org/wiki/MobyGames)

**How to Play Dxball**

* **The game is basically a**[**Breakout clone**](http://en.wikipedia.org/wiki/Breakout_clone)**: you bounce a ball off a paddle at the bottom hitting the bricks on the top of the screen. Hitting all the bricks results in completing the level and going to the next. There are (?) levels to complete. Unlike Breakout, however, is the inclusion of power ups other than extra balls.**
* **If you fail to catch the ball then it will go below and will result in Game over.**
* **Score Increases by hitting the Bricks. And when All the bricks are hit then you mastered the game.**
* **Game differs by Changing of levels. In fact it becomes harder by increasing levels.**
* **In this way you have to play this interesting game.**

**Components used in developing Our game**

1. **Eclipse compiler**
2. **NetBeans 7.0.1 compiler**
3. **Java Language(Object Oriented programming,GUI)**
4. **Necessary Game Images**



**Components DesCription**

Eclipse Compiler



**Eclipse** is a multi-language [software development environment](http://en.wikipedia.org/wiki/Software_development_environment) comprising a

[workspace](http://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](http://en.wikipedia.org/wiki/Plug-in_%28computing%29) system. It is written mostly in [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29).

It can be used to develop applications in Java and, by means of various plug-ins, other

[programming languages](http://en.wikipedia.org/wiki/Programming_language) including [Ada](http://en.wikipedia.org/wiki/Ada_%28programming_language%29), [C](http://en.wikipedia.org/wiki/C_%28programming_language%29), [C++](http://en.wikipedia.org/wiki/C%2B%2B), [COBOL](http://en.wikipedia.org/wiki/COBOL), [Fortran](http://en.wikipedia.org/wiki/Fortran), [Haskell](http://en.wikipedia.org/wiki/Haskell_%28programming_language%29), [Perl](http://en.wikipedia.org/wiki/Perl), [PHP](http://en.wikipedia.org/wiki/PHP), [Python](http://en.wikipedia.org/wiki/Python_%28programming_language%29), [R](http://en.wikipedia.org/wiki/R_%28programming_language%29), [Ruby](http://en.wikipedia.org/wiki/Ruby_%28programming_language%29) (including [Ruby on Rails](http://en.wikipedia.org/wiki/Ruby_on_Rails) framework), [Scala](http://en.wikipedia.org/wiki/Scala_%28programming_language%29), [Clojure](http://en.wikipedia.org/wiki/Clojure), [Groovy](http://en.wikipedia.org/wiki/Groovy_%28programming_language%29), and [Scheme](http://en.wikipedia.org/wiki/Scheme_%28programming_language%29). It can also be used to develop packages for the software [Mathematica](http://en.wikipedia.org/wiki/Mathematica).

Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial [codebase](http://en.wikipedia.org/wiki/Codebase) originated from [IBM VisualAge](http://en.wikipedia.org/wiki/IBM_VisualAge).[[2]](http://en.wikipedia.org/wiki/Eclipse_%28software%29#cite_note-VisualAge-2) The Eclipse SDK (which includes the Java development tools) is meant for Java developers.

Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the [Eclipse Public License](http://en.wikipedia.org/wiki/Eclipse_Public_License), Eclipse [SDK](http://en.wikipedia.org/wiki/Software_development_kit) is [free and open source software](http://en.wikipedia.org/wiki/Free_and_open_source_software) (although it is incompatible with the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License)[[3]](http://en.wikipedia.org/wiki/Eclipse_%28software%29#cite_note-3)).

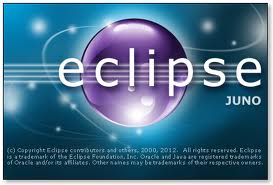
It was one of the first IDEs to run under [GNU Classpath](http://en.wikipedia.org/wiki/GNU_Classpath) and it runs without problems under [IcedTea](http://en.wikipedia.org/wiki/IcedTea).

A **Java compiler** is a [compiler](http://en.wikipedia.org/wiki/Compiler) for the [Java programming language](http://en.wikipedia.org/wiki/Java_%28programming_language%29). The most common form of output from a Java compiler is [Java class files](http://en.wikipedia.org/wiki/Class_%28file_format%29) containing platform-neutral [Java bytecode](http://en.wikipedia.org/wiki/Java_bytecode). There exist also compilers emitting optimized [native machine code](http://en.wikipedia.org/wiki/Machine_code) for a particular hardware/[operating system](http://en.wikipedia.org/wiki/Operating_system) combination.

Most Java-to-bytecode compilers, [Jikes](http://en.wikipedia.org/wiki/Jikes) being a well known exception, do virtually no [optimization](http://en.wikipedia.org/wiki/Optimization_%28computer_science%29), leaving this until [run time](http://en.wikipedia.org/wiki/Run_time_%28program_lifecycle_phase%29) to be done by the [JRE](http://en.wikipedia.org/wiki/JRE)[[*citation needed*](http://en.wikipedia.org/wiki/Wikipedia:Citation_needed)].

The [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) (JVM) loads the class files and either [interprets](http://en.wikipedia.org/wiki/Interpreter_%28computing%29) the [bytecode](http://en.wikipedia.org/wiki/Bytecode) or [just-in-time](http://en.wikipedia.org/wiki/Just-in-time_compilation) compiles it to [machine code](http://en.wikipedia.org/wiki/Machine_code) and then possibly optimizes it using [dynamic compilation](http://en.wikipedia.org/wiki/Dynamic_compilation).

A standard on how to interact with Java compilers programmatically was specified in [JSR](http://en.wikipedia.org/wiki/Java_Community_Process) 199.



**Net Beans 7.0.1**

**C:\Users\Uesr\Desktop\150px-NetBeans.svg.png**

**NetBeans** is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an [application platform](http://en.wikipedia.org/wiki/Platform_%28computing%29) framework for [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29) desktop applications and others.

The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible [JVM](http://en.wikipedia.org/wiki/Java_Virtual_Machine).

The NetBeans Platform allows applications to be developed from a set of modular [software components](http://en.wikipedia.org/wiki/Software_component) called *modules*. Applications based on the NetBeans Platform (including the NetBeans IDE itself) can be extended by [third party developers](http://en.wikipedia.org/wiki/Third_party_developer).[[1]](http://en.wikipedia.org/wiki/NetBeans#cite_note-1)

NetBeans IDE 6.0 introduced support for developing IDE modules and rich client applications based on the NetBeans platform, a Java Swing [GUI](http://en.wikipedia.org/wiki/GUI) builder (formerly known as "Project Matisse"), improved [CVS](http://en.wikipedia.org/wiki/Concurrent_Versions_System) support, [WebLogic](http://en.wikipedia.org/wiki/WebLogic) 9 and [JBoss](http://en.wikipedia.org/wiki/JBoss) 4 support, and many editor enhancements. NetBeans 6 is available in official repositories of major Linux distributions.

NetBeans IDE 6.5, released in November 2008, extended the existing [Java EE](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) features (including Java Persistence support, EJB 3 and JAX-WS). Additionally, the NetBeans Enterprise Pack supports development of Java EE 5 enterprise applications, including [SOA](http://en.wikipedia.org/wiki/Service-oriented_architecture) visual design tools, XML schema tools, web services orchestration (for BPEL), and [UML](http://en.wikipedia.org/wiki/Unified_Modeling_Language) modeling. The NetBeans IDE Bundle for C/C++ supports C/C++ and FORTRAN develop

## NetBeans Platform

The **NetBeans Platform** is a reusable [framework](http://en.wikipedia.org/wiki/Software_framework) for simplifying the development of [Java Swing](http://en.wikipedia.org/wiki/Java_Swing) desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download [digitally signed](http://en.wikipedia.org/wiki/Digital_signature) upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

* User interface management (e.g. menus and toolbars)
* User settings management
* Storage management (saving and loading any kind of data)
* Window management
* Wizard framework (supports step-by-step dialogs)
* NetBeans Visual Library
* Integrated development tools

NetBeans IDE is a free, open-source, cross-platform IDE with built-in-support for Java Programming Language.

## NetBeans IDE

**NetBeans IDE** is an [open-source](http://en.wikipedia.org/wiki/Open_source) integrated development environment. NetBeans IDE supports development of all Java application types ([Java SE](http://en.wikipedia.org/wiki/Java_Platform,_Standard_Edition) (including [JavaFX](http://en.wikipedia.org/wiki/JavaFX)), [Java ME](http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition), [web](http://en.wikipedia.org/wiki/Web_application), [EJB](http://en.wikipedia.org/wiki/EJB) and [mobile](http://en.wikipedia.org/wiki/MIDlet) applications) out of the box. Among other features are an [Ant](http://en.wikipedia.org/wiki/Apache_Ant)-based project system, Maven support, [refactorings](http://en.wikipedia.org/wiki/Refactoring), [version control](http://en.wikipedia.org/wiki/Version_control_system) (supporting [CVS](http://en.wikipedia.org/wiki/Concurrent_Versions_System), [Subversion](http://en.wikipedia.org/wiki/Subversion_%28software%29), [Mercurial](http://en.wikipedia.org/wiki/Mercurial_%28software%29) and [Clearcase](http://en.wikipedia.org/wiki/Clearcase)).

**Modularity**: All the functions of the IDE are provided by modules. Each module provides a well defined function, such as support for the [Java language](http://en.wikipedia.org/wiki/Java_%28programming_language%29), editing, or support for the [CVS](http://en.wikipedia.org/wiki/Concurrent_Versions_System) versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, [Sun Studio](http://en.wikipedia.org/wiki/Sun_Studio_Compiler_Suite), Sun Java Studio Enterprise, and [Sun Java Studio Creator](http://en.wikipedia.org/wiki/Sun_Java_Studio_Creator) from [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) are all based on the NetBeans IDE.

**License**: From July 2006 through 2007, NetBeans IDE was licensed under Sun's [Common Development and Distribution License](http://en.wikipedia.org/wiki/Common_Development_and_Distribution_License) (CDDL), a license based on the [Mozilla Public License](http://en.wikipedia.org/wiki/Mozilla_Public_License) (MPL). In October 2007, Sun announced that NetBeans would henceforth be offered under a [dual license](http://en.wikipedia.org/wiki/Dual_license) of the CDDL and the [GPL](http://en.wikipedia.org/wiki/GPL) version 2 licenses, with the [GPL linking exception](http://en.wikipedia.org/wiki/GPL_linking_exception) for [GNU Classpath](http://en.wikipedia.org/wiki/GNU_Classpath)[[7]](http://en.wikipedia.org/wiki/NetBeans#cite_note-7)

### NetBeans JavaScript editor

The NetBeans JavaScript editor provides extended support for [JavaScript](http://en.wikipedia.org/wiki/JavaScript), Ajax, and [CSS](http://en.wikipedia.org/wiki/CSS).[[10]](http://en.wikipedia.org/wiki/NetBeans#cite_note-10)[[11]](http://en.wikipedia.org/wiki/NetBeans#cite_note-11)

JavaScript editor features comprise [syntax highlighting](http://en.wikipedia.org/wiki/Syntax_highlighting), [refactoring](http://en.wikipedia.org/wiki/Code_refactoring), [code completion](http://en.wikipedia.org/wiki/Autocomplete) for native objects and functions, generation of JavaScript class skeletons, generation of [Ajax](http://en.wikipedia.org/wiki/Ajax_%28programming%29) [callbacks](http://en.wikipedia.org/wiki/Callback_%28computer_science%29) from a template; and automatic [browser compatibility](http://en.wikipedia.org/wiki/JavaScript#Compatibility_considerations) checks.

CSS editor features comprise [code completion](http://en.wikipedia.org/wiki/Autocomplete) for styles names, quick navigation through the navigator panel, displaying the CSS rule declaration in a List View and file structure in a Tree View, sorting the outline view by name, type or declaration order (List & Tree), creating rule declarations (Tree only), refactoring a part of a rule name (Tree only).



**Java Language**

**Java** ([Indonesian](http://en.wikipedia.org/wiki/Indonesian_language): ***Jawa***) is an [island](http://en.wikipedia.org/wiki/Island) of [Indonesia](http://en.wikipedia.org/wiki/Indonesia). With a population of 135 million (excluding the 3.6 million on the island of [Madura](http://en.wikipedia.org/wiki/Madura) which is administered as part of the provinces of Java), Java is the world's [most populous island](http://en.wikipedia.org/wiki/List_of_islands_by_population), and one of the most [densely-populated](http://en.wikipedia.org/wiki/Densely-populated) places on the globe. Java is the home of 60 percent of the Indonesian population. The Indonesian [capital](http://en.wikipedia.org/wiki/Capital_%28political%29) city, [Jakarta](http://en.wikipedia.org/wiki/Jakarta), is located on western Java. Much of Indonesian history took place on Java. It was the center of powerful [Hindu](http://en.wikipedia.org/wiki/Hindu)-[Buddhist](http://en.wikipedia.org/wiki/Buddhist) empires, the [Islamic sultanates](http://en.wikipedia.org/wiki/The_spread_of_Islam_in_Indonesia), and the core of the [colonial](http://en.wikipedia.org/wiki/Colony) [Dutch East Indies](http://en.wikipedia.org/wiki/Dutch_East_Indies). Java was also the center of the [Indonesian struggle for independence](http://en.wikipedia.org/wiki/Indonesian_National_Revolution) during the 1930s and 40s. Java dominates Indonesia [politically](http://en.wikipedia.org/wiki/Politics_of_Indonesia), [economically](http://en.wikipedia.org/wiki/Economy_of_Indonesia) and [culturally](http://en.wikipedia.org/wiki/Culture_of_Indonesia).

Formed mostly [as the result of volcanic eruptions](http://en.wikipedia.org/wiki/Volcanic_island), Java is the [13th largest island](http://en.wikipedia.org/wiki/List_of_islands_by_area) in the world and the fifth largest island in Indonesia. A chain of volcanic mountains forms an east-west spine along the island. It has three main languages, though [Javanese](http://en.wikipedia.org/wiki/Javanese_language) is dominant, and it is the native language of about 60 million people in Indonesia, most of whom live on Java. Most of its residents are [bilingual](http://en.wikipedia.org/wiki/Multilingualism), with [Indonesian](http://en.wikipedia.org/wiki/Indonesian_language) as their first or second languages. While the majority of the people of Java are [Muslim](http://en.wikipedia.org/wiki/Muslim), Java has a diverse mixture of religious beliefs, ethnicities, and cultures.

Java is divided into four provinces, [West Java](http://en.wikipedia.org/wiki/West_Java), [Central Java](http://en.wikipedia.org/wiki/Central_Java), [East Java](http://en.wikipedia.org/wiki/East_Java), and [Banten](http://en.wikipedia.org/wiki/Banten), and also two special regions, [Jakarta](http://en.wikipedia.org/wiki/Jakarta) and [Yogyakarta](http://en.wikipedia.org/wiki/Yogyakarta_%28special_region%29).

[James Gosling](http://en.wikipedia.org/wiki/James_Gosling), Mike Sheridan, and [Patrick Naughton](http://en.wikipedia.org/wiki/Patrick_Naughton) initiated the Java language project in June 1991.[[12]](http://en.wikipedia.org/wiki/Java_language#cite_note-12) Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.[[13]](http://en.wikipedia.org/wiki/Java_language#cite_note-13) The language was initially called [*Oak*](http://en.wikipedia.org/wiki/Oak_%28programming_language%29) after an [oak](http://en.wikipedia.org/wiki/Oak) tree that stood outside Gosling's office; it went by the name *Green* later, and was later renamed *Java*, from [Java coffee](http://en.wikipedia.org/wiki/Java_coffee), said to be consumed in large quantities by the language's creators.[[14]](http://en.wikipedia.org/wiki/Java_language#cite_note-14) Gosling aimed to implement a [virtual machine](http://en.wikipedia.org/wiki/Virtual_machine) and a language that had a familiar [C](http://en.wikipedia.org/wiki/C_%28programming_language%29)/[C++](http://en.wikipedia.org/wiki/C%2B%2B) style of notation.[[15]](http://en.wikipedia.org/wiki/Java_language#cite_note-15)

[Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) released the first public implementation as Java 1.0 in 1995.[[1]](http://en.wikipedia.org/wiki/Java_language#cite_note-oraclejavahistory-1) It promised "[Write Once, Run Anywhere](http://en.wikipedia.org/wiki/Write_once,_run_anywhere)" (WORA), providing no-cost run-times on popular [platforms](http://en.wikipedia.org/wiki/Computing_platform). Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run [*Java applets*](http://en.wikipedia.org/wiki/Java_applet) within web pages, and Java quickly became popular. With the advent of *Java 2* (released initially as J2SE 1.2 in December 1998 – 1999), new versions had multiple configurations built for different types of platforms. For example, *J2EE* targeted enterprise applications and the greatly stripped-down version *J2ME* for mobile applications (Mobile Java). *J2SE* designated the Standard Edition. In 2006, for marketing purposes, Sun renamed new *J2* versions as [*Java EE*](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition), [*Java ME*](http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition), and [*Java SE*](http://en.wikipedia.org/wiki/Java_Platform,_Standard_Edition), respectively.

In 1997, Sun Microsystems approached the [ISO/IEC JTC1](http://en.wikipedia.org/wiki/ISO/IEC_JTC1) standards body and later the [Ecma International](http://en.wikipedia.org/wiki/Ecma_International) to formalize Java, but it soon withdrew from the process.[[16]](http://en.wikipedia.org/wiki/Java_language#cite_note-16) Java remains a [*de facto*](http://en.wikipedia.org/wiki/De_facto) standard, controlled through the [Java Community Process](http://en.wikipedia.org/wiki/Java_Community_Process).[[17]](http://en.wikipedia.org/wiki/Java_language#cite_note-17) At one time, Sun made most of its Java implementations available without charge, despite their [proprietary software](http://en.wikipedia.org/wiki/Proprietary_software) status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System. Sun distinguishes between its [Software Development Kit (SDK)](http://en.wikipedia.org/wiki/Software_development_kit) and [Runtime Environment (JRE)](http://en.wikipedia.org/wiki/HotSpot) (a subset of the SDK); the primary distinction involves the [JRE's](http://en.wikipedia.org/wiki/Java_virtual_machine) lack of the compiler, utility programs, and header files.

On November 13, 2006, Sun released much of Java as [free and open source software](http://en.wikipedia.org/wiki/Free_and_open_source_software), (FOSS), under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL). On May 8, 2007, Sun finished the process, making all of Java's core code available under [free software](http://en.wikipedia.org/wiki/Free_software)/open-source distribution terms, aside from a small portion of code to which Sun did not hold the copyright.[[18]](http://en.wikipedia.org/wiki/Java_language#cite_note-18)

Sun's vice-president Rich Green said that Sun's ideal role with regards to Java was as an "evangelist."[[19]](http://en.wikipedia.org/wiki/Java_language#cite_note-19) Following [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation)'s acquisition of Sun Microsystems in 2009–2010, Oracle has described itself as the "steward of Java technology with a relentless commitment to fostering a community of participation and transparency".[[20]](http://en.wikipedia.org/wiki/Java_language#cite_note-20) This did not hold Oracle, however, from filing a lawsuit against Google shortly after that for using Java inside the Android SDK (see Google section below). Java software runs on [laptops](http://en.wikipedia.org/wiki/Laptop) to [data centers](http://en.wikipedia.org/wiki/Data_center), [game consoles](http://en.wikipedia.org/wiki/Video_game_console) to scientific [supercomputers](http://en.wikipedia.org/wiki/Supercomputer). There are 930 million [Java Runtime Environment](http://en.wikipedia.org/wiki/Java_%28software_platform%29) downloads each year and 3 billion [mobile phones](http://en.wikipedia.org/wiki/Mobile_phone) run Java.[[21]](http://en.wikipedia.org/wiki/Java_language#cite_note-21) On April 2, 2010, James Gosling resigned from Oracle.[[22](http://en.wikipedia.org/wiki/Java_language#cite_note-22)

There were five primary goals in the creation of the Java language:[[23]](http://en.wikipedia.org/wiki/Java_language#cite_note-23)

1. It should be "simple, object-oriented and familiar"
2. It should be "robust and secure"
3. It should be "architecture-neutral and portable"
4. It should execute with "high performance"
5. It should be "interpreted, threaded, and dynamic"

### Versions

Main article: [Java version history](http://en.wikipedia.org/wiki/Java_version_history)

Major release versions of Java, along with their release dates:

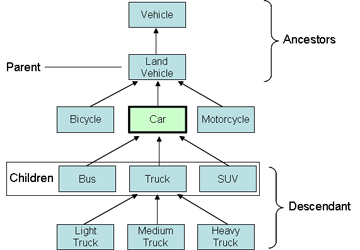
* JDK 1.0 (January 21, 1996)
* JDK 1.1 (February 19, 1997)
* J2SE 1.2 (December 8, 1998)
* J2SE 1.3 (May 8, 2000)
* J2SE 1.4 (February 6, 2002)
* J2SE 5.0 (September 30, 2004)
* Java SE 6 (December 11, 2006)
* Java SE 7 (July 28, 2011)

****

**Object oriented programming**

**Object-oriented programming** (**OOP**) is a [programming paradigm](http://en.wikipedia.org/wiki/Programming_paradigm) that represents concepts as "[objects](http://en.wikipedia.org/wiki/Object_%28computer_science%29)" that have [data fields](http://en.wikipedia.org/wiki/Field_%28computer_science%29) (attributes that describe the object) and associated procedures known as [methods](http://en.wikipedia.org/wiki/Method_%28computer_science%29). Objects, which are [instances](http://en.wikipedia.org/wiki/Instance_%28computer_science%29) of [classes](http://en.wikipedia.org/wiki/Class_%28computer_science%29), are used to interact with one another to design applications and computer programs.[[1]](http://en.wikipedia.org/wiki/Object-oriented_programming#cite_note-1) [[2]](http://en.wikipedia.org/wiki/Object-oriented_programming#cite_note-2)

**Example of a Object Oriented Progamming**



**Dx-ball Games Necessary principles**

**Game Features**

* ***NEW GAME***
* ***LOAD GAME***
* ***SAVE GAME***
* ***SETTINGS***
* ***ABOUT***
* ***HELP***
* ***EXIT***

Description of the Game main menu window

***New game:*** *\* Single player****Load game:*** *Start a previously saved game****Settings****:  
\*sound: sets the sound off or on  
\*music: a music that will be being played through the whole time  
On/off  
\*sfx: sound of Brick hitting and Ball throwing etc.  
On/off****Help****:  
\*instructions: how to play the game  
\*Levels: details about the levels after levels  
\*History: The story on which the game is based on****About****:  
Info about the game developer****Exit****:  
Exit the game window*

Game Objects

***1.Bar:***

***A rectangle shaped bar will be at the bottom of the game window with a constant colour.***

***Bar Size: Small and constant.***

***Bar movement: Bar will move by pressing the keyboard movement (left sticks). And that will move from left to the right.***

***Number of bar: one***

***2. Bricks:***

***The bricks will be placed in a particular position. They will be rectangle in size. There number will be more than 10 or more.  
Size: Small/medium***

***Colour: Yellow, Red, green***

***Movement: Boxes won’t move.***

***3. Ball:***

***A ball with a circle shape will be always moving when it touches the bar.***

***Size: Small  
Hitpoints: same for all boxes***

***Colour: Same for all levels .***

***4. Screen:***

***Screen colour will change by the change of levels.***

***Size: Large***

***5. Levels: 2 levels***

**In Game Necessary images**

* ***Bar:***

******

* ***Ball:***

**

* *Brick:*

******

* ***Background:***

**

*Lifes:*

**

*Game\_Over\_Image:*

**

Rockets:



Bullets:



Ball\_On\_hury\_image:

C:\Users\Uesr\Desktop\game images\ball_on_hury.jpg

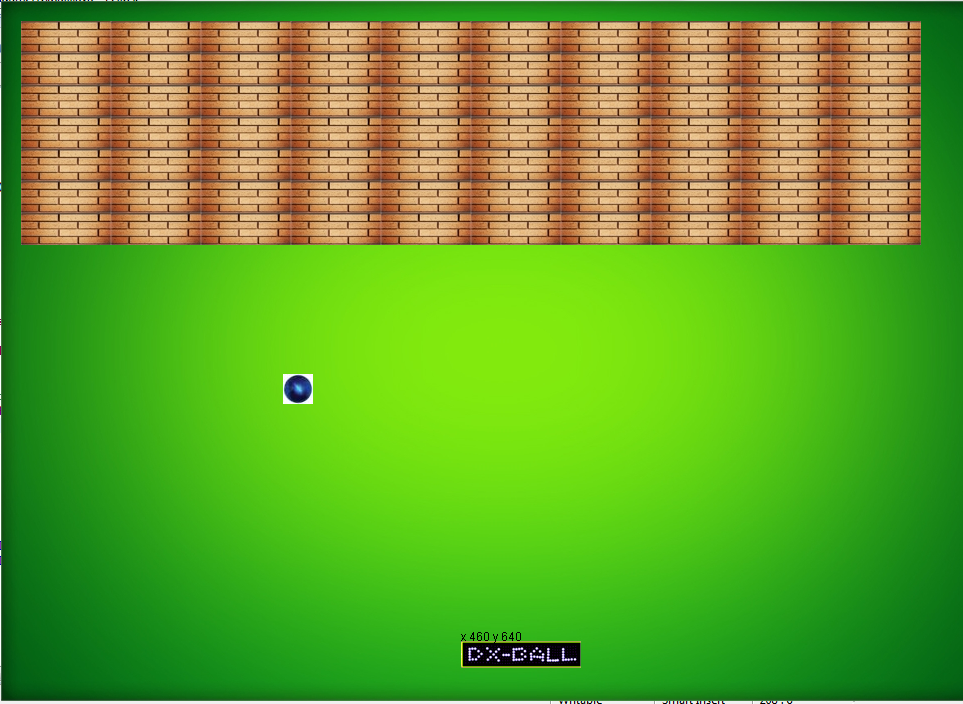
Bar\_Size\_increase:

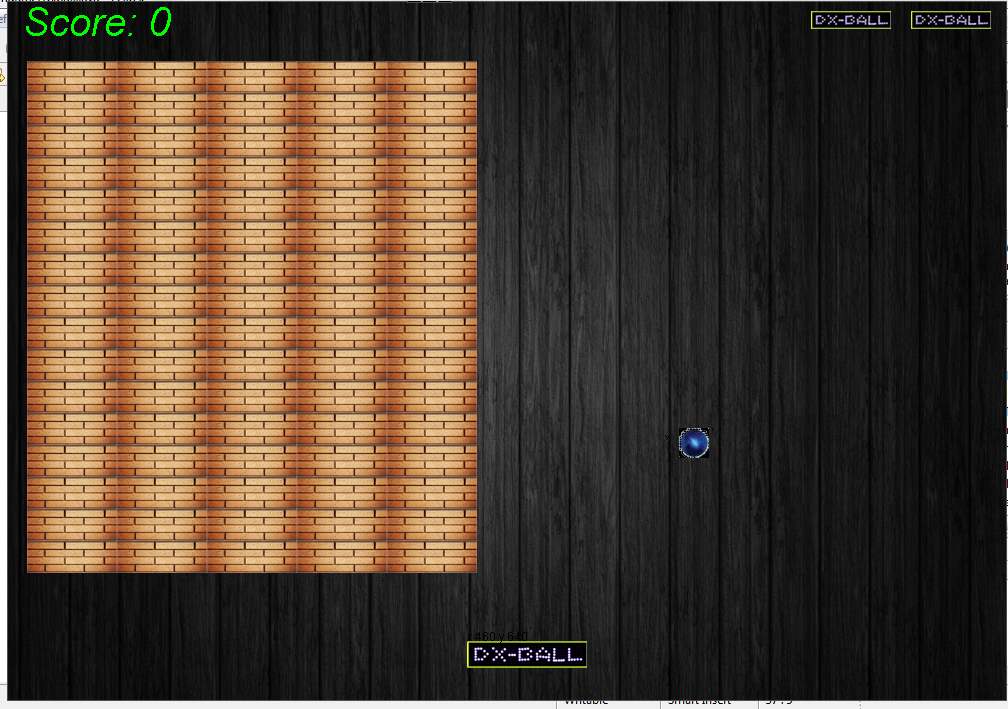
C:\Users\Uesr\Desktop\game images\bar_size_increase2.png

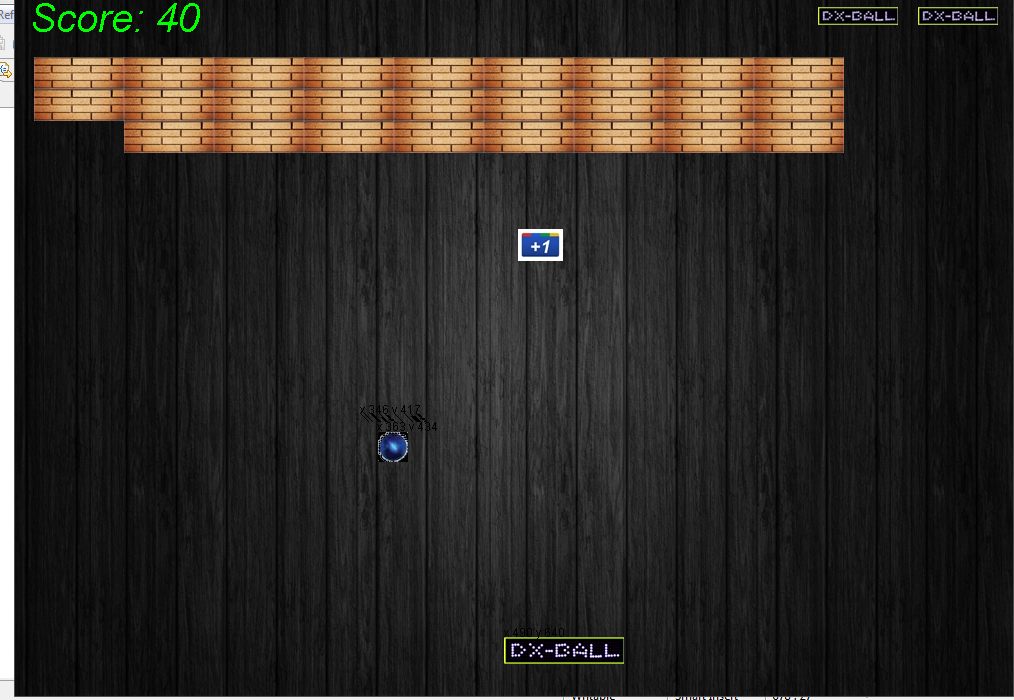
Bar\_Size\_Decrease:

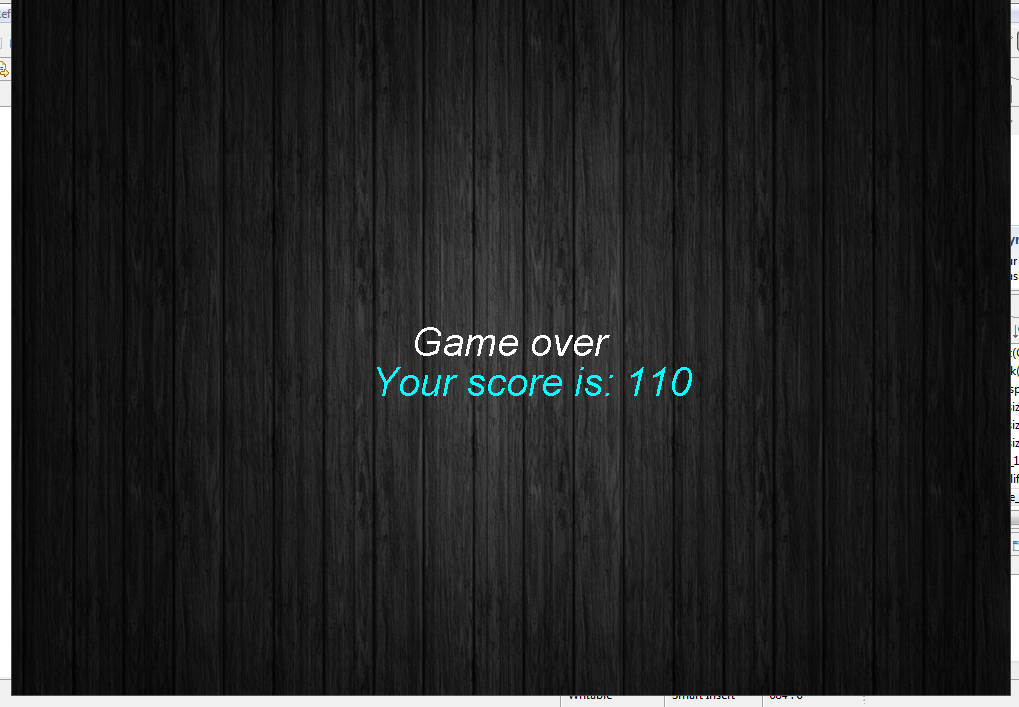
C:\Users\Uesr\Desktop\game images\bar_size_decrease.png

Game ScreenShots played at different times:









**In Game Code Description:**

**We used multiple classes in the game**

**Such as**

* **Dxball.java**
* **Mainclass.java**
* **Mainmenu.java**
* **Sound.java**
* **Elements.java**
* **Input.java etc..etc**

**Our Dx-ball.java class is a JFrame**

**Code :**

**class Dxball extends JFrame // it is now a JFrame**

**{**

**}**

**We took a common characteristics of our game, putted them into the Element.java class.**

**And then by Object Oriented principle we called each and every variables.**

**Which made our work easier.**

**package maingame;**

**import java.awt.Graphics;**

**import java.awt.Image;**

**import javax.swing.ImageIcon;**

**public class Elements //Elements.java class**

**{**

**int x;**

**int y;**

**Image img;**

**public boolean isalive;**

**public Elements(String filepath,int x,int y,boolean isalive) //Elements class constructor**

**{**

**img =new ImageIcon(filepath).getImage();**

**this.x=x;**

**this.y=y;**

**this.isalive=isalive;**

**}**

**public Elements(Image img,int x,int y,boolean isalive) //another constructor**

**{**

**this.img =img;**

**this.x=x;**

**this.y=y;**

**this.isalive=isalive;**

**}**

**public int getwidth() //code for getting image width**

**{**

**if(img==null)**

**return 0;**

**else**

**return img.getWidth(null);**

**}**

**public int getHeight() //code for getting image height**

**{**

**if(img==null)**

**return 0;**

**else**

**return img.getHeight(null);**

**}**

**public void draw(Graphics g) //code for drawing any images**

**{**

**if(img!=null && isalive)**

**g.drawImage(img, x, y, null);**

**}**

**public void draw\_bar(Graphics g,int width) //code for drawing bar**

**{**

**if(img!=null && isalive)**

**g.drawImage(img,x, y, width, 27, null);**

**}**

**We created another class named Input.java where we just only took input from the keyboard.**

**For our gameplay it is necessary to use keyboard that’s why we used keyboard. And putted the codes in Input.java**

**package maingame;**

**import java.awt.event.KeyEvent;**

**import java.awt.event.KeyListener;**

**public class Input implements KeyListener {**

**public int x\_axis\_increment=30;**

**//static int x=Dxball.getbarx();**

**Elements bar;**

**Elements rocket\_left;**

**Elements rocket\_right;**

**Elements bullet\_at\_rocket\_left;**

**Elements bullet\_at\_rocket\_right;**

**public Input(Elements b,Elements r\_l,Elements r\_r,Elements b\_at\_r\_l,Elements b\_at\_r\_r)**

**{**

**//The keylistener being registered by this constructor**

**rocket\_left=r\_l;**

**rocket\_right=r\_r;**

**bar=b; //Taking the Object of Dxball bar**

**bullet\_at\_rocket\_left=b\_at\_r\_l;**

**bullet\_at\_rocket\_right=b\_at\_r\_r;**

**}**

**@Override**

**public void keyPressed(KeyEvent e) {**

**int k=e.getKeyCode();// TODO Auto-generated method stub**

**if(k==KeyEvent.VK\_ESCAPE)**

**System.exit(0); //game will terminate**

**else if(k==KeyEvent.VK\_LEFT && bar.x>5) //bar movement**

**{**

**bar.x=bar.x-x\_axis\_increment;**

**rocket\_left.x=rocket\_left.x-x\_axis\_increment;**

**rocket\_right.x=rocket\_right.x-x\_axis\_increment;**

**bullet\_at\_rocket\_left.x=bullet\_at\_rocket\_left.x-x\_axis\_increment;**

**bullet\_at\_rocket\_right.x=bullet\_at\_rocket\_right.x-x\_axis\_increment;**

**}**

**else if(k==KeyEvent.VK\_RIGHT && bar.x<871) //bar movement**

**{**

**bar.x=bar.x+x\_axis\_increment;**

**rocket\_left.x=rocket\_left.x+x\_axis\_increment;**

**rocket\_right.x=rocket\_right.x+x\_axis\_increment;**

**bullet\_at\_rocket\_left.x=bullet\_at\_rocket\_left.x+x\_axis\_increment;**

**bullet\_at\_rocket\_right.x=bullet\_at\_rocket\_right.x+x\_axis\_increment;**

**}**

**else if(k==KeyEvent.VK\_ENTER) // game will be started**

**Dxball.isgamestarted=true;**

**}**

**@Override**

**public void keyReleased(KeyEvent arg0) { //not used**

**// TODO Auto-generated method stub**

**}**

**@Override**

**public void keyTyped(KeyEvent arg0) {**

**// TODO Auto-generated method stub**

**}**

**}**

**Code for the ballMove**

**In this code the we initialized ball movement**

**public** **void** ballmove()

{

**if**(**this**.ball\_speed\_increase())

{

**if**(**this**.barupperintersect\_with\_ball\_on\_hury())

{

ball\_speed\_timer--;

Sound.*ball\_speed\_increase\_sound*();

System.*out*.println("one life plus bonus touched");

**this**.ball\_on\_hury.y=**this**.*HEIGHT*+100;

}

**if**(ball\_on\_hury.y==*HEIGHT*)

{

ball\_on\_hury.x=-30;

}

ball\_on\_hury.y++;

repaint();

}

**try** {

Thread.*sleep*(ball\_speed\_timer);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**if**(barleftintersect())

{

System.*out*.println("left true"+signy);

signy\*=(-1);

System.*out*.println("after change"+signy);

signx\*=(-1);

}

/\*

if(barrightintersect())

{

System.out.println("right true"+signy);

signy\*=(-1);

System.out.println("after change"+signy);

signx\*=-1;

}

\*/

**else** **if**(ball.x<5 || ball.x>950)

{

Sound.*wall\_touch\_sound*();

signx=signx\*(-1);

}

**else** **if**(ball.y<5 || barupperintersect())

{

Sound.*wall\_touch\_sound*();

signy=signy\*(-1);

}

// condition for gaming over

**if**(ball.y>=*HEIGHT* && **this**.Life\_count>=0)

{

**this**.Life\_count--;

Sound.*gameover\_sound*();

**if**(**this**.Life\_count==0)

{

msg="Game over";

repaint();

gameover=**true**;

}

signy=-1;

}

**int** x1=ball.x;

**int** y1=ball.y;

//Rectangle r1=new Rectangle(x1,y1,x1+ball.getwidth(),y1+ball.getHeight());

// Rectangle r2=new Rectangle(ball.x,ball.y,ball.x+ball.getwidth(),ball.y+ball.getHeight());

ball.x=ball.x-signx;

ball.y=ball.y+signy;

// Rectangle r=new Rectangle();

// repaint(x1,y1,x1+ball.getwidth(),y1+ball.getHeight(),ball.x,ball.y,ball.x+ball.getwidth(),ball.y+ball.getHeight());

// repaint(ball.x,ball.y,x1+ball.getwidth(),y1+ball.getHeight());

//this.rep

//repaint();

repaint(x1,y1,x1+ball.getwidth(),y1+ball.getHeight()); //repaint the previous position

repaint(ball.x,ball.y,ball.x+ball.getwidth(),ball.y+ball.getHeight());//repaint the new position

}

**void** go()

{

//long gameSpeed=0;

// repaint();

**long** t1,t2;

t1=System.*currentTimeMillis*();

**while**(**true**)

{

System.*out*.println("while");

**if**(*isgamestarted*)

{

System.*out*.println("if");

t2=System.*currentTimeMillis*();

// if(gameSpeed%3000000==0){

ballmove();

**if**(t2-t1>=80){

repaint();

// gameSpeed=0;

//}

//gameSpeed++;

t1=System.*currentTimeMillis*();

}

}

}//while

}//go

**Code for initializing the bricks**

**In this code we initialized our bricks**

**private** **void** initializebricks() // brick initialization codes are here

{

**int** x=BRICK\_FIRST\_X;

**int** y=BRICK\_FIRST\_Y;

brick=**new** Elements[row\_number\_of\_bricks][column\_number\_of\_bricks]; // 7 is row number of brick and 10 is column number of brick by changing this we can change brick numbers

**for**(**int** r=0;r<brick.length;r++)

{

x=BRICK\_FIRST\_X;

**for**(**int** c=0;c<brick[r].length;c++)

{

brick[r][c]=**new** Elements(*BRICK\_IMAGE*,x,y,**true**);

x+=brick[r][c].getwidth();

}

//System.out.println(r);

y+=brick[r][0].getHeight();

}

}

**Paint component of our class**

**In this code we have drawn each and every drawing’s in the code.**

**We also drawn ball,background,brick,and others necessary images and also moved them.**

**We took Graphics g (**import java.awt.Graphics)

**public** **void** paint(Graphics g)

{

//g.clearRect(0, 0, WIDTH,HEIGHT);

background.draw(g);

**this**.game\_over\_image.draw(g);

**if**(**this**.Life\_count==4) // life count condition

{

**this**.small\_bar1.draw(g);

**this**.small\_bar2.draw(g);

**this**.small\_bar3.draw(g);

}

**if**(**this**.Life\_count==3)

{

**this**.small\_bar2.draw(g);

**this**.small\_bar3.draw(g);

}

**if**(**this**.Life\_count==2)

{

**this**.small\_bar3.draw(g);

}

// this.rocket\_left.draw(g);

// this.rocket\_right.draw(g);

//bullet\_at\_rocket\_left.draw(g);

// bullet\_at\_rocket\_right.draw(g);

//bar\_size\_increase.draw(g);

//this.bar\_size\_decrease.draw(g);

**if**(**this**.bar\_size\_decrease())

{

**if**(**this**.barupperintersect\_with\_bar\_size\_decrease())

{

Sound.*bar\_size\_decrease*();

bar\_width=120;

System.*out*.println("one life plus bonus touched");

**this**.bar\_size\_decrease.y=**this**.*HEIGHT*+100;

}

**if**(bar\_size\_decrease.y==*HEIGHT*)

{

bar\_size\_decrease.x=-30;

}

bar\_size\_decrease.y++;

bar\_size\_decrease.draw(g);

}

**if**(**this**.bar\_size\_increase())

{

**if**(**this**.barupperintersect\_with\_bar\_size\_increase())

{

Sound.*bar\_size\_increase\_sound*();

bar\_width=190;

**this**.rocket\_right.x=bar\_width+160;

System.*out*.println("one life plus bonus touched");

bar\_size\_increase.y=-30;

}

**if**(bar\_size\_increase.y==*HEIGHT*)

{

bar\_size\_increase.x=-30;

}

bar\_size\_increase.y++;

bar\_size\_increase.draw(g);

}

**if**(**this**.bar\_size\_increase\_secondtime())

{

**if**(**this**.barupperintersect\_with\_bar\_size\_increase())

{

Sound.*bar\_size\_increase\_sound*();

bar\_width=190;

System.*out*.println("one life plus bonus touched");

**this**.bar\_size\_increase2.y=**this**.*HEIGHT*+100;

}

**if**(bar\_size\_increase2.y==*HEIGHT*)

{

bar\_size\_increase2.x=-30;

}

bar\_size\_increase2.y++;

bar\_size\_increase2.draw(g);

}

**if**(**this**.game\_over\_image\_run())

{

**if**(**this**.barupperintersect\_with\_game\_over\_image() )

{

Sound.*gameover\_sound*();

System.*out*.println("one life plus bonus touched");

**this**.game\_over\_image.x=-100;

}

**if**(**this**.game\_over\_image.y==*HEIGHT*)

{

**this**.game\_over\_image.x=-100;

}

**this**.game\_over\_image.y++;

**this**.game\_over\_image.draw(g);

}

**if**(**this**.one\_life\_plus())

{

**if**(**this**.barupperintersect\_with\_one\_life\_plus() && **this**.Life\_count<5)

{

**this**.Life\_count++;

Sound.*one\_life\_plus\_sound*();

System.*out*.println("one life plus bonus touched");

one\_life\_plus.x=-100;

}

**if**(one\_life\_plus.y==*HEIGHT*)

{

one\_life\_plus.x=-100;

}

one\_life\_plus.y++;

one\_life\_plus.draw(g);

}

//brick.draw(g);

bar.draw\_bar(g,bar\_width);

calcbricks\_drawbricks(g);

g.drawString("x "+bar.x+" y "+ bar.y, bar.x, bar.y);

Rectangle r=g.getClipBounds();

g.setClip(ball.x,ball.y,ball.x+ball.getwidth(),ball.y+ball.getHeight());

ball.draw(g);

g.setClip(r);

g.drawString("x "+ball.x+" y "+ ball.y, ball.x, ball.y);

g.setColor(Color.*GREEN*);

Font font=**new** Font("Arial",Font.*ITALIC*,40); // printing the scores

g.setFont(font);

**if**(score==**null**)

score="0";

g.drawString("Score: "+score, 15, 35);

//String s=String.valueOf(this.Life\_count);

// g.drawString("Life "+s, 500, 350);

**if**(gameover==**true**) //for game over message

{

background.draw(g);

*isgamestarted*=**false**;

g.setColor(Color.*WHITE*);

g.drawString(msg, 400, 360);

g.setColor(Color.*CYAN*);

g.drawString("Your score is: " +score, 360, 400);

}

**if**(level\_1\_complete())

{

// this.setRootPaneCheckingEnabled(false);

//new Dxball\_level2().ballmove();

background.draw(g);

*isgamestarted*=**false**;

g.setColor(Color.*WHITE*);

g.drawString("Score: "+score, 405, 355);

// ob = new Dxball\_Level2();

}

**Bricks intersecting logics logics:**

**We used four Booleans to implement this**

**private** **void** calcbricks\_drawbricks(Graphics g) //if brick intersects then return false, by this brick will be vanished otherwise draw it

{

**for**(**int** r=0;r<brick.length;r++)

**for**(**int** c=0;c<brick[r].length;c++){

**if**(isbrick\_bottom\_intersected(brick[r][c]))

{

//brick[r][c]=null;

brick[r][c].isalive=**false**; // object of the brick is becoming false

signy=1; // after intersecting ball returns

System.*out*.println("bottom intersected");

score\_count+=10;

score=String.*valueOf*(score\_count);

repaint();

System.*out*.println(score\_count);

}

**else** **if**(isbrick\_leftside\_intersected(brick[r][c]))

{

//brick[r][c]=null;

brick[r][c].isalive=**false**; // object of the brick is becoming false

signx=-1; // after intersecting ball returns

score\_count+=10;

score=String.*valueOf*(score\_count);

repaint();

System.*out*.println(score\_count);

//System.out.println("left side intersected");

}

**else** **if**(isbrick\_upper\_intersected(brick[r][c]))

{

//brick[r][c]=null;

brick[r][c].isalive=**false**; // object of the brick is becoming false

signy=-1; // after intersecting ball returns

score\_count+=10;

score=String.*valueOf*(score\_count);

repaint();

System.*out*.println(score\_count);

//System.out.println("left side intersected");

}

/\*

if(isbrick\_rightside\_intersected(brick[r][c]))

{

//brick[r][c]=null;

brick[r][c].isalive=false; // object of the brick is becoming false

signx=1; // after intersecting ball returns

//System.out.println("left side intersected");

}

\*/

**else**

brick[r][c].draw(g);

//System.out.println("not intersected");

}

}

**private** **boolean** isbrick\_bottom\_intersected(Elements brick)

{

**if**(brick!=**null** && signy==-1 && brick.isalive &&( (ball.x>=brick.x && ball.x<=brick.x+brick.getwidth())

|| ball.x+ball.getwidth()>=brick.x && ball.x+ball.getwidth()<=brick.x+brick.getwidth() )

&& (ball.y<=brick.y+brick.getHeight() && ball.y>=brick.y) )

{

Sound.*brick\_break\_sound*();

**return** **true**;

}

**else**

**return** **false**;

}

**private** **boolean** isbrick\_upper\_intersected(Elements brick)

{

**if**(brick!=**null** && signy==1 && brick.isalive && ( (ball.x>=brick.x && ball.x<=brick.x+brick.getwidth())

|| ball.x+ball.getwidth()>=brick.x && ball.x+ball.getwidth()<=brick.x+brick.getwidth() ) &&

(ball.y + ball.getHeight() >=brick.y && ball.y+ball.getHeight()<= brick.y+ brick.getHeight()))

{

Sound.*brick\_break\_sound*();

System.*out*.println(" upper brick intersected");

**return** **true**;

}

**else**

**return** **false**;

}

**private** **boolean** isbrick\_leftside\_intersected(Elements brick)

{

**if**(brick!=**null** && signx==1 && brick.isalive && (ball.x+ball.getwidth()>=brick.x && ball.x+ball.getwidth()<=brick.x+brick.getwidth()) &&

((ball.y>=brick.y && ball.y<=brick.y+brick.getHeight()) || (ball.y+ball.getHeight()>=brick.y && ball.y+ball.getHeight()<=brick.y+brick.getHeight()) ))

{

Sound.*brick\_break\_sound*();

System.*out*.println("left side intersected");

**return** **true**;

}

**else**

**return** **false**;

}

/\*

private boolean isbrick\_rightside\_intersected(Elements brick)

{

if(brick!=null && signx==-1 && brick.isalive && (ball.x<=brick.x+brick.getwidth() && ball.x>=brick.x) &&

((ball.y>=brick.y && ball.y<=brick.y+brick.getHeight()) || (ball.y+ball.getHeight()>=brick.y && ball.y+ball.getHeight()<=brick.y+brick.getHeight())) )

{

System.out.println(" right side intersected");

return true;

}

else

return false;

}

\*/

**Objects of the elements class**

Elements ball;

//Elements brick;

Elements bar;

Elements brick[][];

Elements background;

Elements background2;

Elements one\_life\_plus;

Elements bar\_size\_increase;

Elements bar\_size\_increase2;

Elements bar\_size\_decrease;

Elements game\_over\_image;

Elements ball\_on\_hury;

Elements small\_bar1;

Elements small\_bar2;

Elements small\_bar3;

Elements rocket\_left;

Elements rocket\_right;

Elements bullet\_at\_rocket\_left;

Elements bullet\_at\_rocket\_right;

Dxball\_Level2 ob;

**By using all of this we completed our game DxBall**

**What we couldn’t do:**

In our game we couldn’t call the main method , by which our game starts… from the actionperformed() method . We placed the Thread.sleep() method in the ballmove function and it is in the While loop. So each and every time it is being called.

And the it freezes. So when we call it it doesn’t work.

This was our obstackle.

We also couldn’t give mainmenu for this same reason.

A Slide from Internet is given below:

## The GUI freeze problem

When you write Swing applications, you show a GUI to a user; the user clicks on some components (buttons, menus, etc.) to perform the desired action.  
The code that executes the action is written in event listeners methods, and event listeners are always executed in the **Event Dispatch Thread**.  
The Event Dispatch Thread is responsible for taking one event after another and processing it; the processing involves calling the event listeners's methods, which are then executed. If an event listener requires a long time to be executed, then the Event Dispatch Thread cannot process the next event, which will then be waiting in the Event Queue.  
If the pending event is a repaint event, the GUI cannot be repainted, so it appears to be **frozen**.  
So resizing your window, overlapping it with another window, clicking on other components, all these events are queued but not processed until the time-consuming listener has finished.   
The user feels the application has hung.  
When the time-consuming listener finishes, all pending events are processed, and if they are quick to execute (like repaint events) it appears they're are executed like a storm

**We faced this problem in our gaming code.**

**We also used the code of sound in our game code**

**The code for the Sound is in Sound.java**

**// This was all about our description of the codes //**

**Game Levels**

**Level 1**

*Objective: Hitting all the Boxes in the window   
Location: Green Screen.*

*Description:*

A constant sized ball will be moving and when it comes down it will touch the bar and will return to the top. If the player fails to touch the ball through bar then the ball will fall down , and the game will over. Player will have to hit each and every boxes shown in the window to pass Level 1.

Level 2

*Objective: Hitting more Boxes than Level 1 within a particular time*

*Location: Black Screen.  
Description:*

In this Level Player will have to hit more boxes than level 1. That means the number of Boxes will increase. And there will be a particular time period. In this time period the player will have to finish his game. Otherwise the game will be Over. The rules of Level 2 will be as same as in Level 1.

**We used only two levels in our game code**

**Conclusion**

* **DX-Ball is an interesting game.**
* **It is one of the smallest games in pc.**
* **People of every ages can play it and have fun.**
* **Specially the children have fun playing it.**
* **In fact DX-ball is a good game for all.**
* **Though we did some mistakes in our codes we hope that the player will like to play our game.**
* **Whatever mistakes we did, we hope that we can eradicate this mistakes in future.**
* **We can make better Game.**
* **This was all about our Dx-ball game.**

****

****