



BREDER LANGUAGE

Bernardo Breder

ÍNDICE

- Definição da Linguagem Breder
- Projetos Relacionados
- Funcionalidades Principais
- Funcionalidades Futuras
- Demonstrações

DEFINIÇÃO

DEFINIÇÃO

- Linguagem Orientado a Objeto, Fortemente Tipada com Escopo Estático
- Multiplataforma rodando em Windows, Linux e MacOS para 32 e 64 bits
- Ambiente Mobile rodando em iPhone
- Sintaxe similar a Linguagem Java com alguns recursos de Lua, Python e C++
- Desalocação de memória automática pelo Coletor de Lixo

DEFINIÇÃO

- Compilação para bytecode, sendo executado pela Máquina Virtual Breder
- Otimizada para operações nativas [futuro]
- Recurso de Orquestragem de outras Linguagens de Programação [futuro]

PROJETOS RELACIONADOS

PROJETOS

- Breder Language Web Site
- Breder Compiler
- Breder Virtual Machine
- Breder SDK
- Breder Language Test
- Breder Debugger
- Breder Processor
- Breder Eclipse Plugin

BREDER.ORG

The screenshot shows a web browser window with the following details:

- Title Bar:** Breder Programming Language - Official Website
- Address Bar:** http://www.breder.org/lang/
- Search Bar:** Google
- Page Content:**
 - ## Breder Language

[main](#) - [doc](#) - [api](#) - [bni](#) - [project](#) - [download](#) - [email](#)
 - Welcome message: "Welcome to the page of the new Breder Language built by Bernardo Breder. The index of this page is :"
 - target**:

The Breder Language is a high level language with strong typing and static scope. This best thinks of this language is to build a project that you need to use a high level language and a low level language.

For example, the target is to get a project with many data struct, Object Orientation, Gargage Collection and has many processing in the C Language Environment. With this, every big processing will run in the C Language Environment and every struct processing will run in the Breder Language.

This language was builded for PC and Mobile. With this, the Breder Language need be very fast with processing and many code of API is implemented in the C Language. For all this work with performance, the Breder Virtual Machine need be very fast with the change of Breder Language to C Language and the chance of C Language for Breder Language.
 - document**:

In this index, you can find a documentation of the Breder Language. The documentation can explain how to use, how it work, where can find thinks with same subject. This page is very importante, because there are many trips that you need to know for programming in this language.
 - download**:

In this index, you can find the Breder Language for download. Because of this language is multi plataform, in this link you can find download for Windows 32, Windows 64, Linux 32, Linux 64, MacOS 32, MacOS 64.
 - email**:

In this Index, you can send email for the developer of the Breder Language. If you have same doubt or interessed in study this Language, you can send email. This email is the main email of the Breder Language has. Because of that, you can send email with no problem, that we will anwser early.

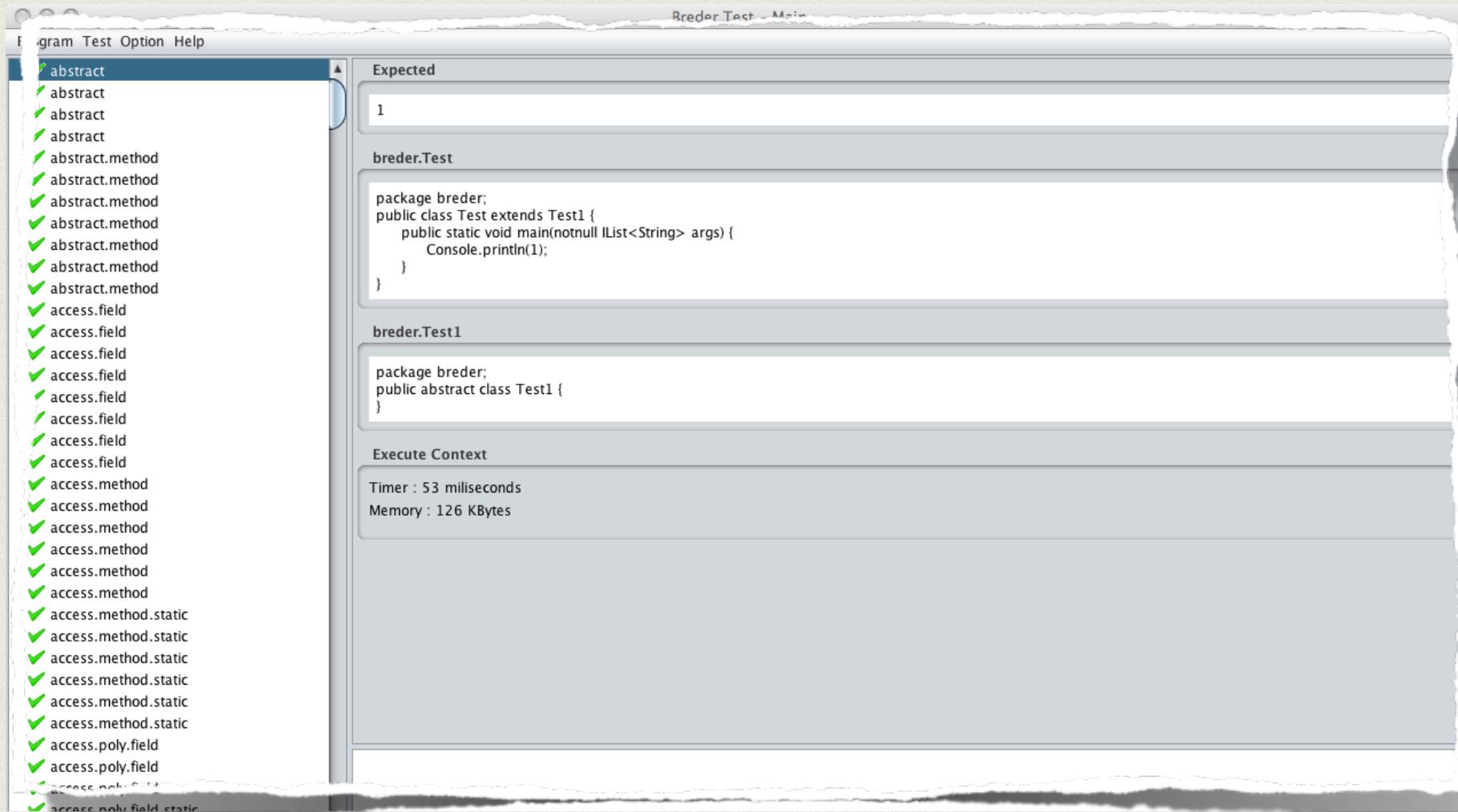
COMPILER E BVM

- **Breder Compiler** é responsável pela geração de arquivo bytecode através do código fonte de um projeto
- **Breder Virtual Machine** é responsável por executar um arquivo bytecode

BREDER SDK

- Linguagem Breder Disponibiliza um SDK básico :
 - **breder.lang** - classes da Linguagem
 - **breder.util** - classes Utilitárias
 - **breder.io** - classes de Entrada e Saída
 - **breder.net** - classes de Network
 - **breder.gui** - classes de Interface Gráfica

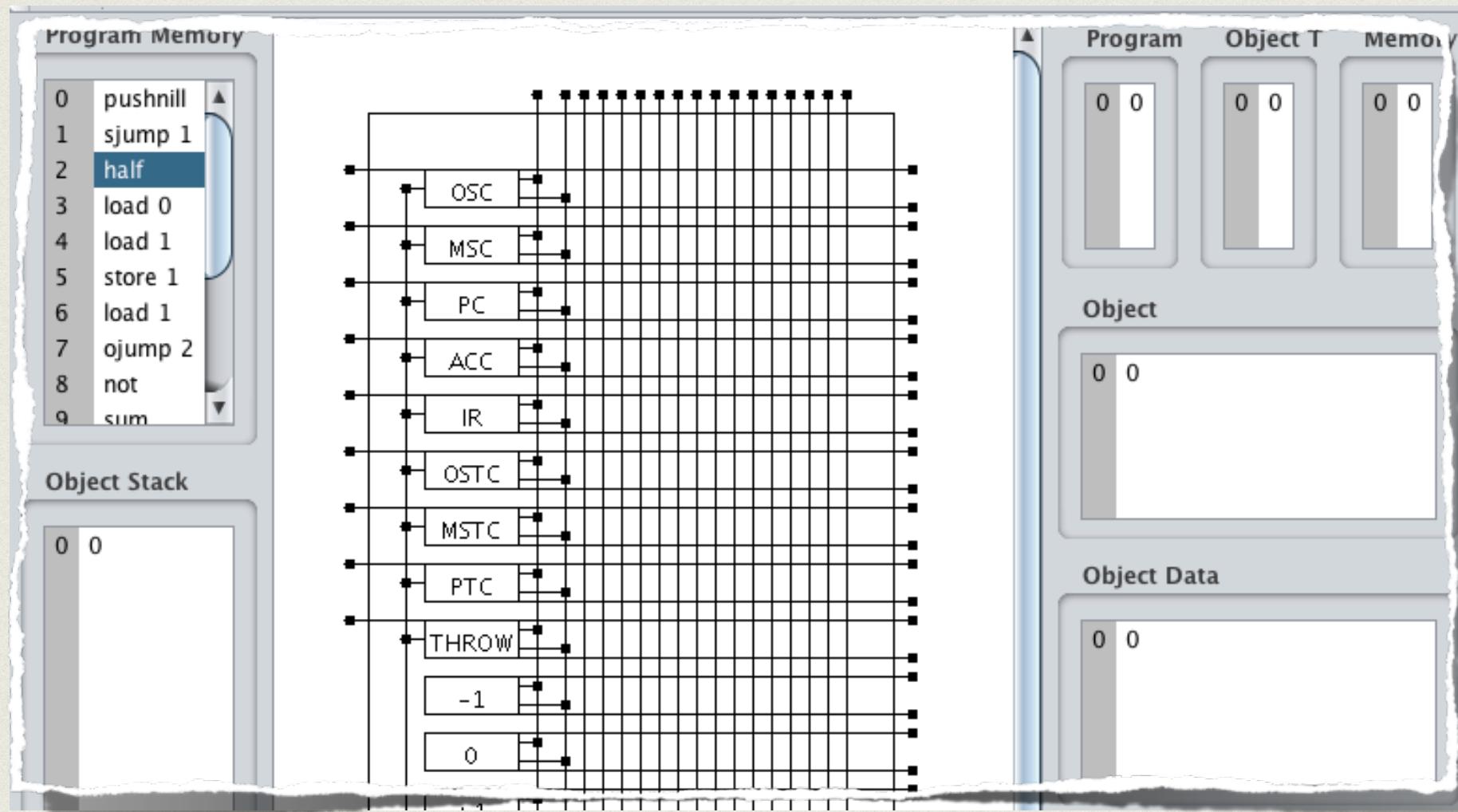
BREDER TESTER



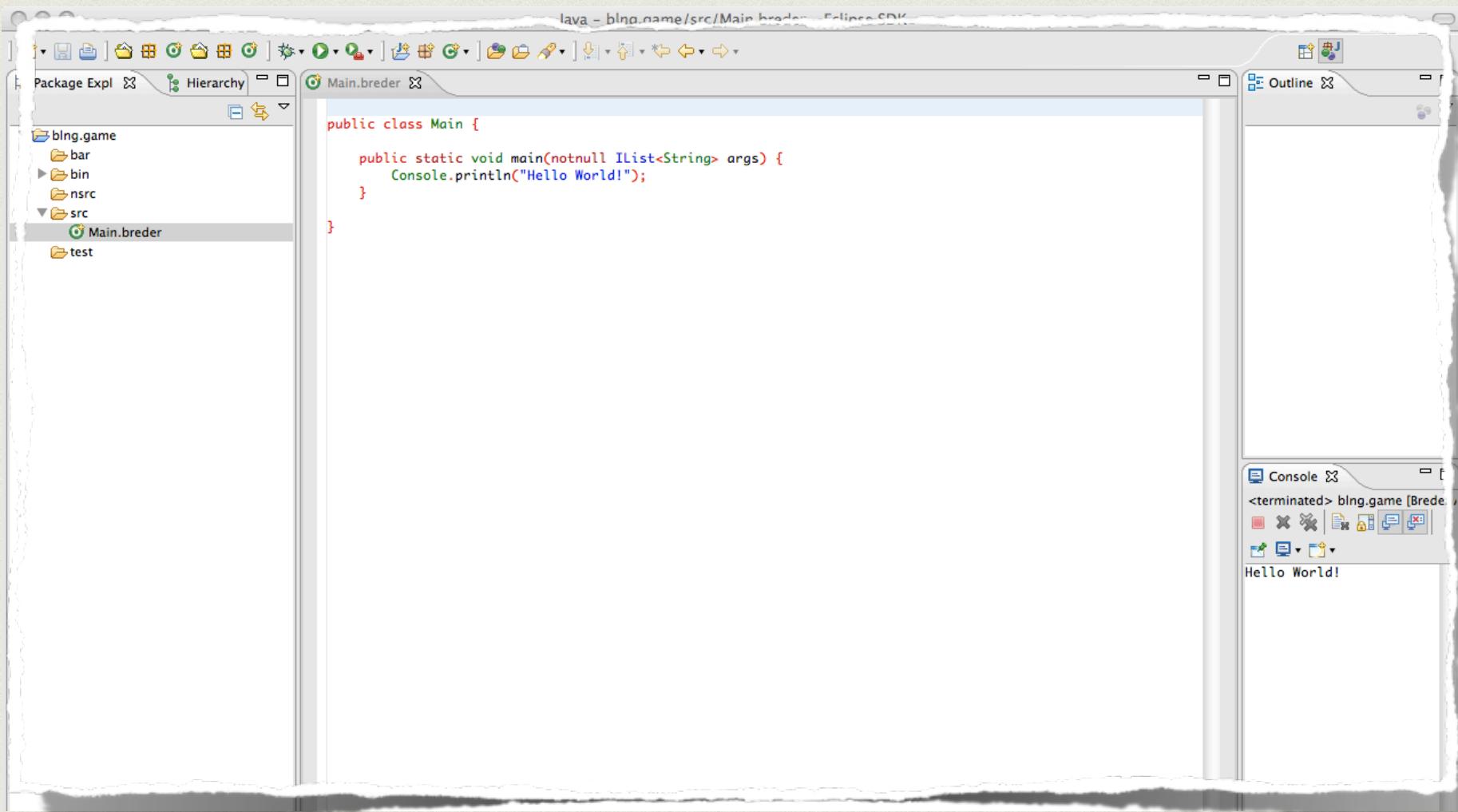
BREDER DEBUGGER

- **Ambiente de Depuração** para analisar a execução do aplicativo
- **Ambiente de Trace** para analisar a trajetória de execução do aplicativo

BREDER PROCESSOR



BREDER ECLIPSE



FUNCIONALIDADES PRINCIPAIS

OPERAÇÃO NATIVA

- Linguagem Breder foi desenvolvida para possuir uma grande performance em operações nativas
- Assim, todo o SDK da Linguagem Breder está implementado nativamente
- A maior parte do processamento está no ambiente compilado para a máquina real

ESPECIFICAÇÃO

- O SDK da Linguagem Breder possui diversas especificações desde o pacote básico breder.lang até o pacote mais complexo breder.gui
- Todas as especificações são armazenadas em forma de Interfaces
- Dessa forma, a Linguagem Breder possui um tratamento mais avançado em cima das interfaces

COLETOR DE LIXO

- A Linguagem Breder possui 2 Coletores de Lixo
- Real Time Garbage Collector - Mais adequado para aplicativos Real Time
- Event Orientation Garbage Collector [futuro] - Mais adequado para aplicativos que dependam de interação do Usuário

FUNCIONALIDADES FUTURAS

ORQUESTRAÇÃO DE LINGUAGENS

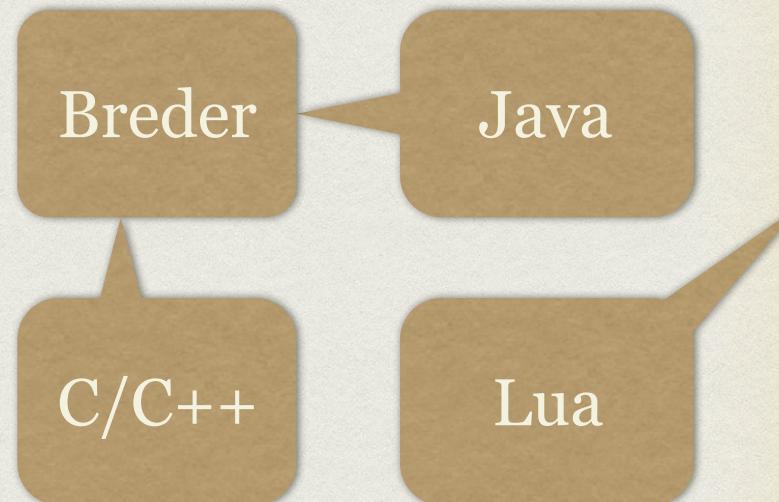
- Linguagem Breder possui o recurso de orquestrar outras Linguagens
- Através disso é possível na Linguagem Breder efetuar chamadas de operações implementadas em outras Linguagens
- Por exemplo, na Linguagem Breder podemos chamar um método implementado na Linguagem Lua

ORQUESTRAÇÃO DE LINGUAGENS

- Abaixo temos exemplos de declarações de métodos implementados em outras Linguagens de Programação :
 - **public native void test ()**
 - **public lua void test ()**
 - **public java void test ()**

ORQUESTRAÇÃO DE LINGUAGENS

- A ilustração mostra que a Linguagem Breder pode comunicar com outras Linguagens
- A comunicação deve ser transparente para o desenvolvedor da Linguagem Breder
- Projeto implementado em outra Linguagem pode ser anexada a um projeto da Linguagem Breder



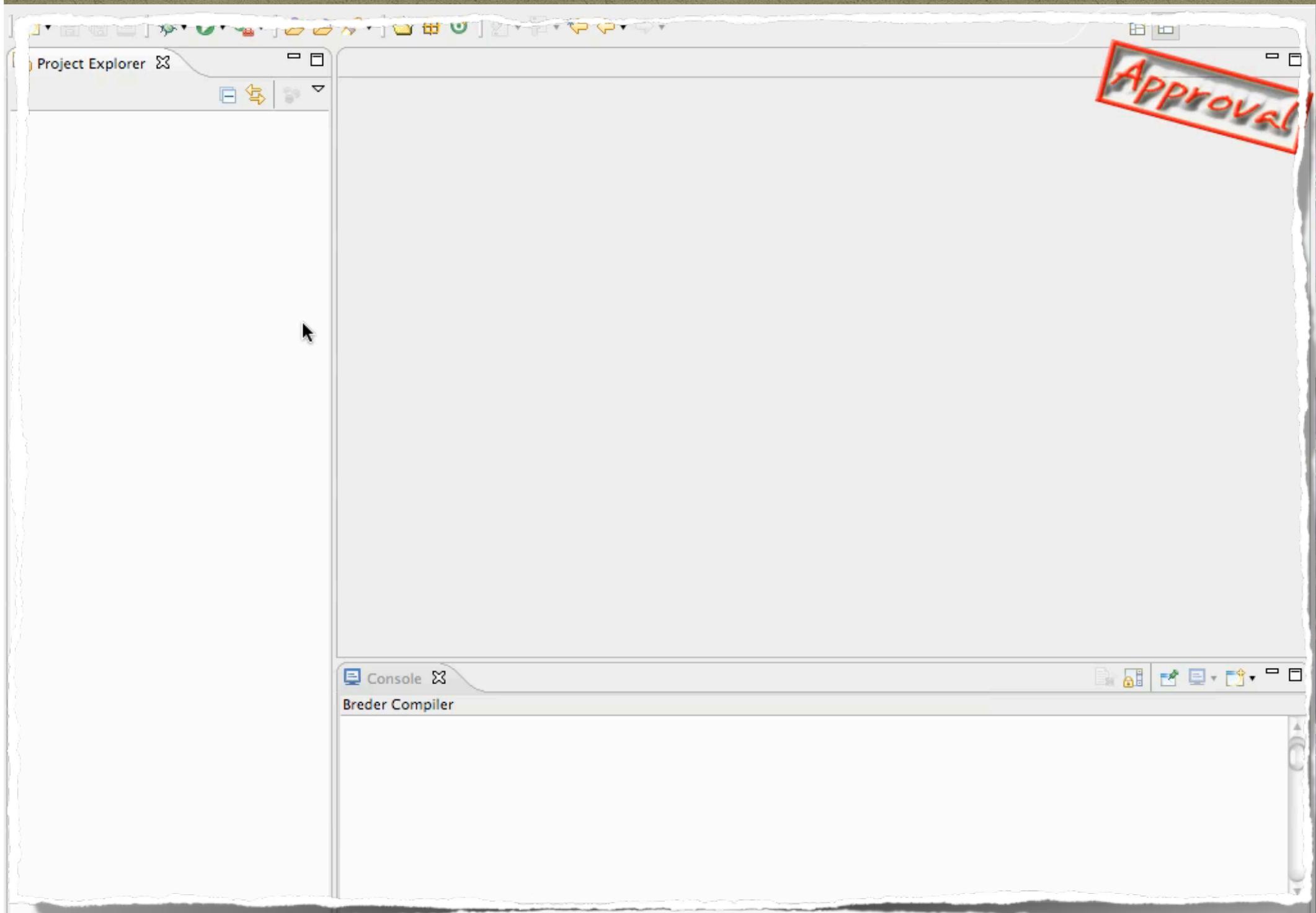
DEMONSTRAÇÃO

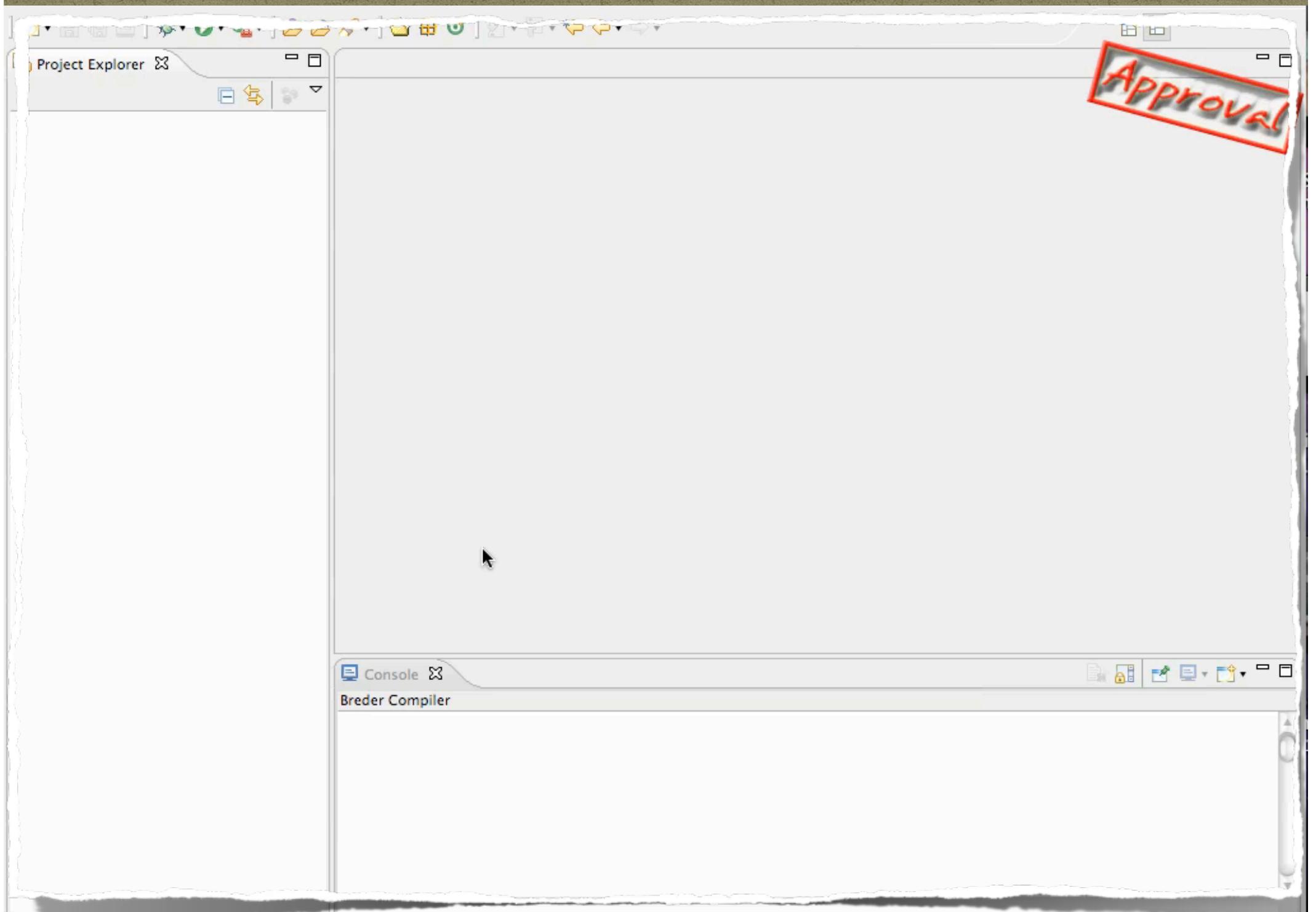
DEMO DE JOGO

java - Breder_gui_OpenGL_api/src/Main.breder - Eclipse - /Users/bernardobreder/breder/eclipse/project

```
42     GLNative.pop();
43     GLNative.push();
44     GLNative.color(0, 0, 255);
45     for (Index u = 1 ; u <= 5 ; u += 1) {
46         GLNative.translate(0, 0, i);
47         GLNative.drawWireCube(i);
48     }
49     GLNative.pop();
50 }
51 if (true) {
52     GLNative.materialShininess(false, false, 1);
53     GLNative.color(255, 0, 0);
54     GLNative.rect(this.x, this.y, this.x + this.rs,
55 );
56 if (false) {
57     GLNative.push();
58     GLNative.rotateZ(this.x);
59     GLNative.beginPath();
60     Number size = 2*Math.pi();
61     Number z = 0;
62     for (Number angle = 0 ; angle <= size ; angle +
63         Number x = 50 * Math.sin(angle);
64         Number y = 50 * Math.cos(angle);
65         GLNative.vertex(x, y, z);
66         z += 0.5;
67     }
68     GLNative.end();
69     GLNative.pop();
70 }
71 if (true) {
72     GLNative.push();
73     GLNative.enableLighting();
74     GLNative.lightModelAmbient(254, 255, 255);
75     GLNative.materialAmbientDiffuse(false, true, 10);
76     GLNative.shade(true);
77     GLNative.beginTriangle();
78     GLNative.color(255,0,0);
79     GLNative.vertex(0,200,0);
80     GLNative.color(0,255,0);
81     GLNative.vertex(200,-70,0);
82     GLNative.color(0,0,255);
83     GLNative.vertex(-200,-70,0);
84     GLNative.end();
85 }
```

String Not Found





OBRIGADO