



# Symposium On Applied Computing Track on Intelligent, Interactive and Innovative Learning Environments

## Amê: An Environment to Learn and Analyze Adversarial Search Algorithms Using Stochastic Card Games

Ana Beatriz Cruz, Sabrina Serique, Leonardo Preuss, Angélica Ogasawara,  
João Quadros, Eduardo Bezerra, Uéverton Souza, Eduardo Ogasawara



**Federal Center of Technological  
Education of Rio de Janeiro  
CEFET/RJ**



# *Computer Agents in Adversarial Games*

- Computer Agent (Agent) as an opponent
- Agent targets
  - Minimizing the impact of opponent's actions
  - Maximizing the return of agent's actions
- Artificial Intelligence Principles
  - Development of [Search Strategies](#)

# *Adversarial AI Search Algorithms*

- Basic search strategies
  - Breadth First
  - Deep First
  - A\* Search
- Adversarial search strategies
  - Deterministic (Full observable)
    - Minimax
    - Alpha-Beta Pruning
    - H-Minimax
  - Stochastic (Partial observable)
    - Expectiminimax



# Stochastic Adversarial Games

## Card games

More fun to play

### Bridge



















































Four players...

### Poker



Bets and Bluffing!

# Japanese Hanafuda Cards (flower cards)

Family	Kô	Tane	Tan	Kasu	Kasu
<i>Matsu</i>					
<i>Ume</i>					
<i>Sakura</i>					
<i>Fuji</i>					
<i>Shoubu</i>					
<i>Botan</i>					
<i>Hagi</i>					
<i>Susuki</i>					
<i>Kiku</i>					
<i>Momiji</i>					
<i>Yanagi</i>					
<i>Kiri</i>					

- Twelve Families
- Family
  - month of the year
  - basic flower
  - Four cards
- Cards can only be combined with family\*
- Kasu cards are valueless
- Tan are striped cards (10 points)
- Tane cards have more information and worth 10 points
- Kô cards worth 50 points

# Hanafuda Combos

Family	Kô	Tane	Tan	Kasu	Kasu
<i>Matsu</i>					
<i>Ume</i>					
<i>Sakura</i>					
<i>Fuji</i>					
<i>Shoubu</i>					
<i>Botan</i>					
<i>Hagi</i>					
<i>Susuki</i>					
<i>Kiku</i>					
<i>Momiji</i>					
<i>Kiri</i>					
<i>Yanagi</i>					

- 10 different combos (such as Nizoro)
- 2 of them lead to immediately win (Nanatan and Shiko)

Nanatan

Shiko

Nizoro



# Playing Hanafuda



Agent's  
earned cards

Agent's cards  
in hand

Cards  
on stock

Cards on  
table

Player's  
cards in hand

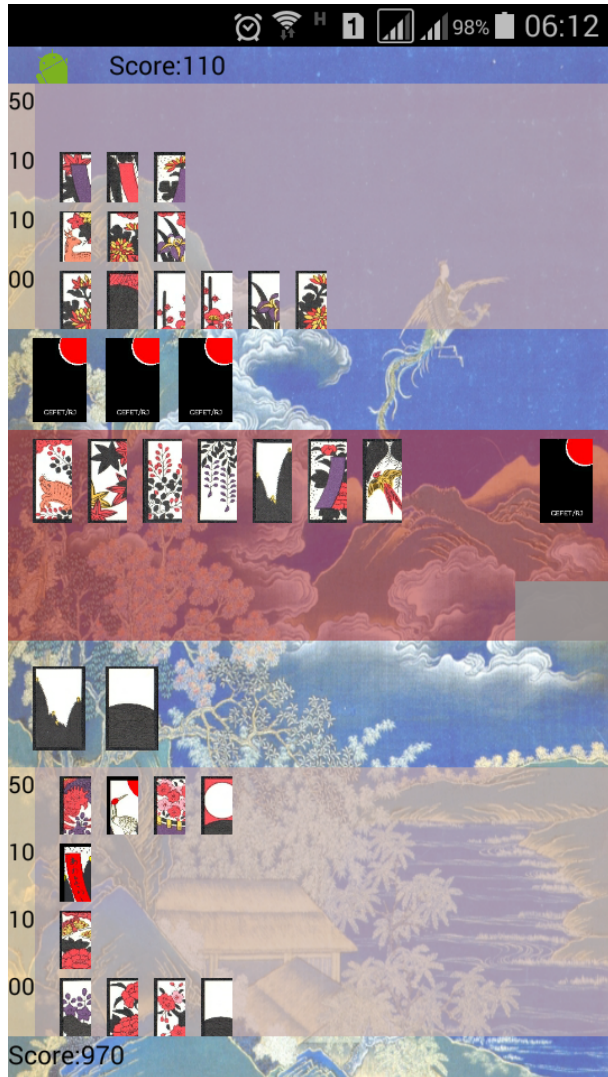
Player's  
earned cards

Player and Agent  
alternate a sequence of  
two movements:

1. Pick one card of hand and try to combine it with table to earn them. Leave it on table if no match found.
2. Pick a card from stock and try to combine it with table to earn them. Leave it on table if no match found.



# Benefits of Hanafuda

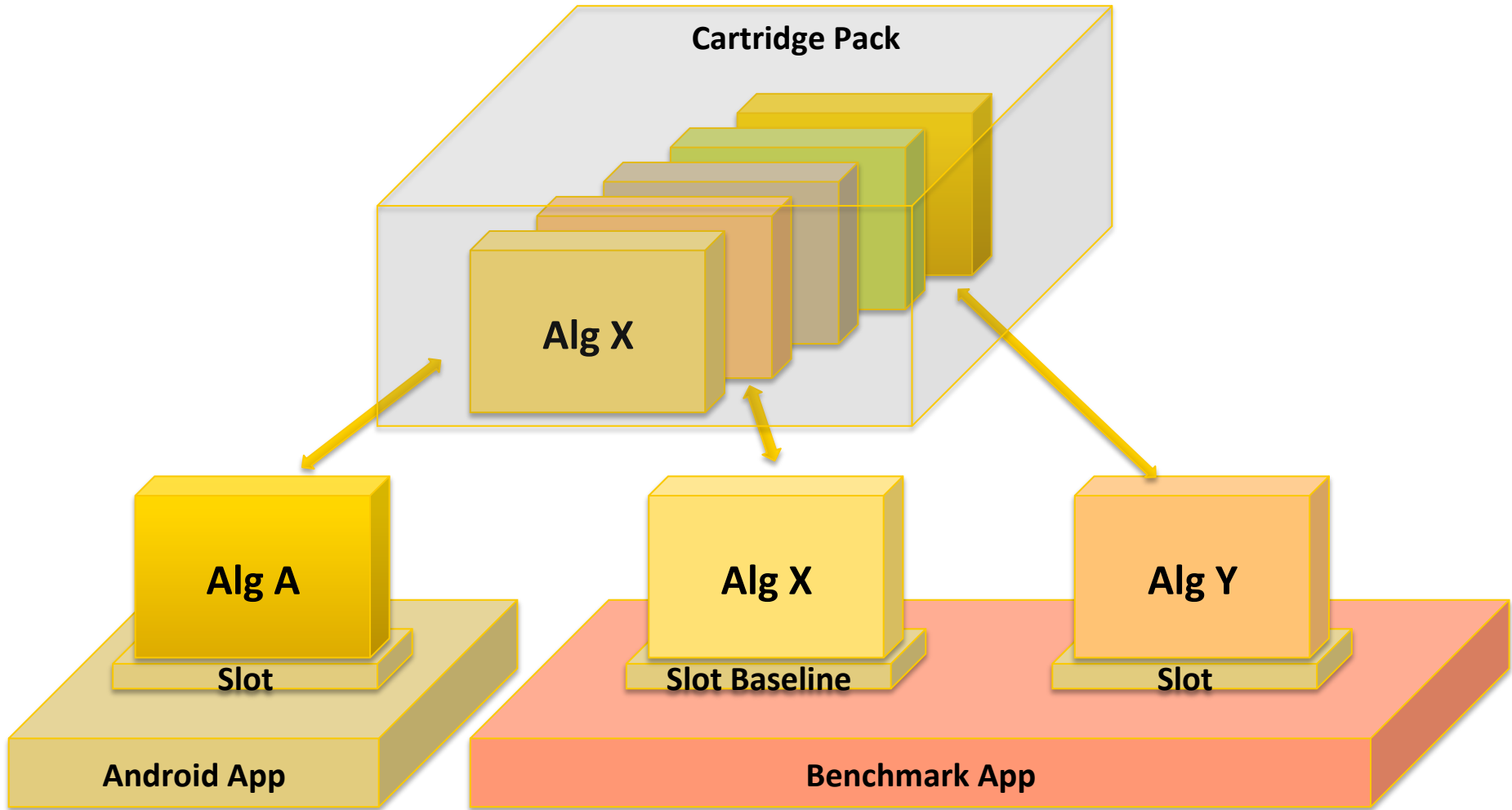


- Different ways to win such as Poker, *i.e.*, empowers the development of multiple strategies for search algorithms
- Fun to play
- No drawbacks of bets and bluffing

# *Amê: An Novel Environment to Learn and Analyze Adversarial Search Algorithms*

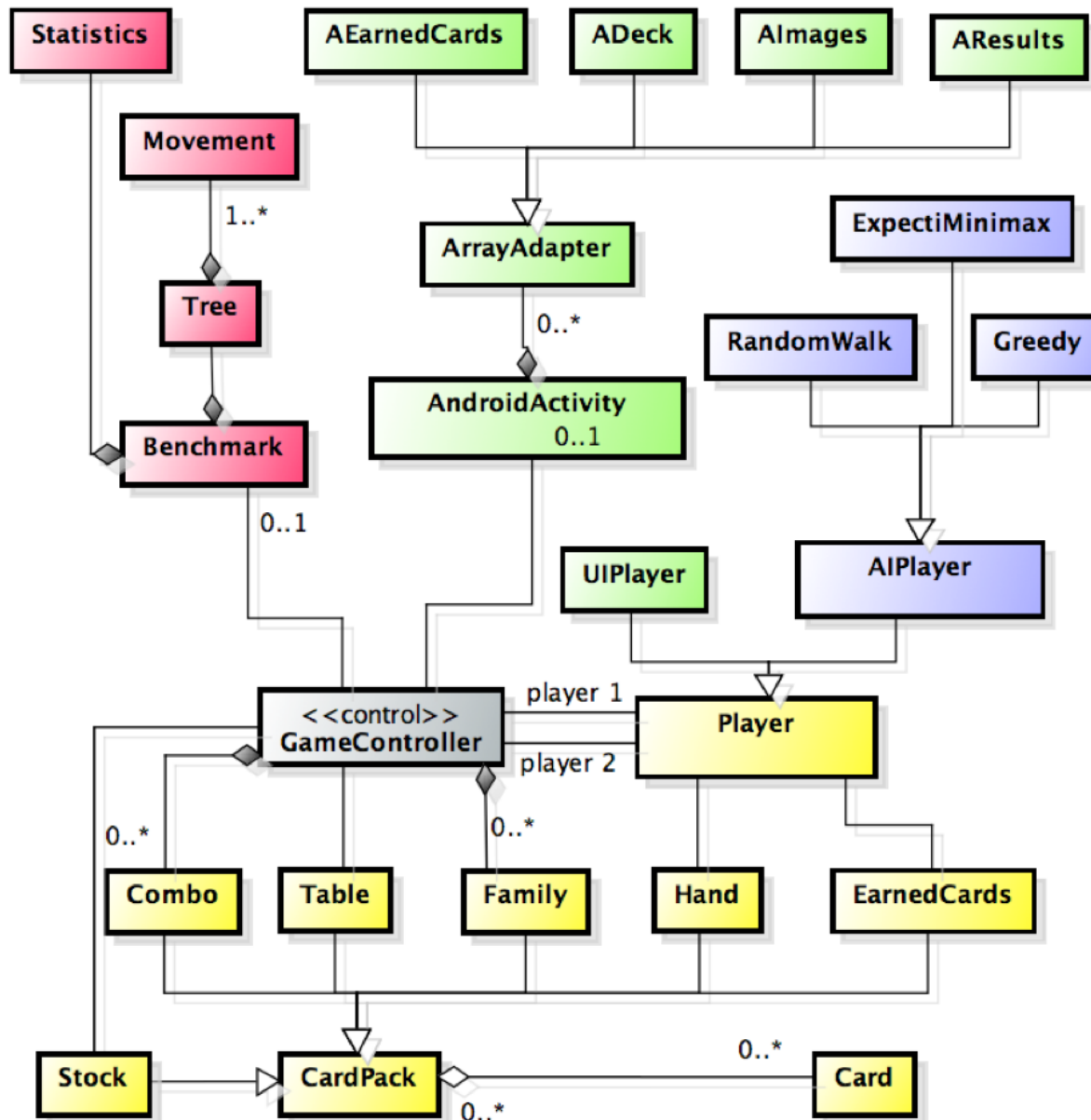
- Context of stochastic card games
- Algorithms are developed considering two perspectives
- Interactive environment
  - Mobile application
  - Fun to develop
  - Fun to play
- Benchmark environment
  - Terminal/Console application
  - Play AI algorithms against each other
  - Empower algorithms analysis (complexity and performance)

# *Amê Architecture*



Project available at [sourceforge](https://sourceforge.net/projects/ame/)

# Amé Class Diagram



- Game play
- AI algorithms
- Benchmark
- Interactive

# Android Application

The image shows a browser window displaying the Google Play Store page for the 'Hanafuda' app. The browser's address bar shows the URL: <https://play.google.com/store/apps/details?id=br.gpca.hanafuda.android&hl=en>. The Google Play logo is visible in the top left. The navigation bar includes 'Apps', 'Categories', 'Home', 'Top Charts', and 'New Releases'. The left sidebar lists 'My apps', 'Shop', 'Games', and 'Editors' Choice'. The main content area features the app's title 'Hanafuda', developer 'GPCA - March 29, 2015', and a green 'L' icon. A green 'Installed' button is present, along with a compatibility message: 'This app is compatible with all of your devices.' The app has a rating of 4.5 stars from 7 users. The app icon depicts a character in a kimono surrounded by colorful flowers and cards.

Available at Google Play Store

## *Benchmark Application*

- Compare AI cartridges running many matches
  - Random Walk
  - Greedy
  - Greedy - Shiko
  - Greedy - Nanatan
  - Cut-Expectminimax
- Collects statistics from matches

## *Hanafuda Search Space*

- Game starts with 8 cards on the table and 8 cards in each player's hand
- The cards of the adversary are not known by the algorithm, so all possible combinations should be considered, as each card can be combined with 3 other cards in the deck , general search when round  $\neq 1$ :

$$\sum_{i=1}^{\lceil d/2 \rceil} 21^i \cdot 36^{i-1} + \sum_{j=0}^{\lceil d/2 \rceil} 21^j \cdot 36^j$$

depth	1	2	3	4
nodes	22	756	16632	588136



# Cut-Expectiminimax

A variation of expectiminimax considering a cut off depth

**function** Cut-Expectiminimax(*state*, *d*) returns an action  
    **return** MAX-VALUE(MIN-VALUE(RESULT(*state*, *a*), *d*))

---

**function** MAX-VALUE(*state*, *d*) returns a expected value  
    **if** CUTOFF-TEST(*state*,*d*) **then**  
        **return** EXPECTED-VALUE(*state*)  
     $v \leftarrow (-\infty)$   
    **for each** *a*  $\in$  ACTIONS(*state*) **do**  
         $v \leftarrow \text{MAX}(v, \text{CHANCE}(\text{"min"}, (\text{RESULT}(\text{state}, a), d)))$   
    **return** *v*

---

**function** MIN- VALUE(*state*, *d*) returns a expected value  
    **if** CUTOFF-TEST(*state*, *d*) **then**  
        **return** EXPECTED-VALUE(*state*)  
     $v \leftarrow \infty$   
    **for each** *a*  $\in$  ACTIONS(*state*) **do**  
         $v \leftarrow \text{MIN}(v, \text{CHANCE}(\text{"max"}, (\text{RESULT}(\text{state}, a), d)))$   
    **return** *v*

**function** CHANCE(*node*, *state*, *d*) returns a expected value  
    **if** CUTOFF-TEST(*state*, *d*) **then**  
        **return** EXPECTED-VALUE(*state*)  
    **if** *node* = "min" **then**  
         $\text{sum} \leftarrow 0$   
        **for each** *a*  $\in$  ACTIONS(*state*) **do**  
             $\text{val} \leftarrow \text{MIN-VALUE}(\text{RESULT}(\text{state}, a), d)$   
             $\text{sum} \leftarrow \text{sum} + \text{Pr}(\text{val}) * \text{val}$   
        **return**  $\text{sum}$

**if** *node* = "max" **then**  
         $\text{sum} \leftarrow 0$   
        **for each** *a*  $\in$  ACTIONS(*state*) **do**  
             $\text{val} \leftarrow \text{MAX-VALUE}(\text{RESULT}(\text{state}, a), d)$   
             $\text{sum} \leftarrow \text{sum} + \text{Pr}(\text{val}) * \text{val}$   
        **return**  $\text{sum}$

## *Experimental Evaluation*

- Benchmark comparison among four implementation
  - Random walk algorithm (RWA)
  - Greedy algorithm (GA)
  - Cut-Expectiminimax (2) - *CutEM(2)*
  - Cut-Expectiminimax (3) - *CutEM(3)*
- 100,000 matches

# Experimental Evaluation

## Benchmark of Greedy

Criteria	Greedy versus
	Random Walk
Wins (%)	54195 (55,2%)
Draws	1961
Average score difference in wins	223
Average score difference in losses	164

## Benchmark of CutEM(d) level 2 and 3

Criteria	CutEM(2) versus		CutEM(3) versus	
	Random Walk	Greedy	Random Walk	Greedy
Wins (%)	55619 (56,7%)	49743 (50,7%)	55703 (56,7%)	49853 (50,8%)
Draws	1978	1963	1904	1969
Average score difference in wins	232	227	232	228
Average score difference in losses	154	208	156	209

# Experimental Evaluation

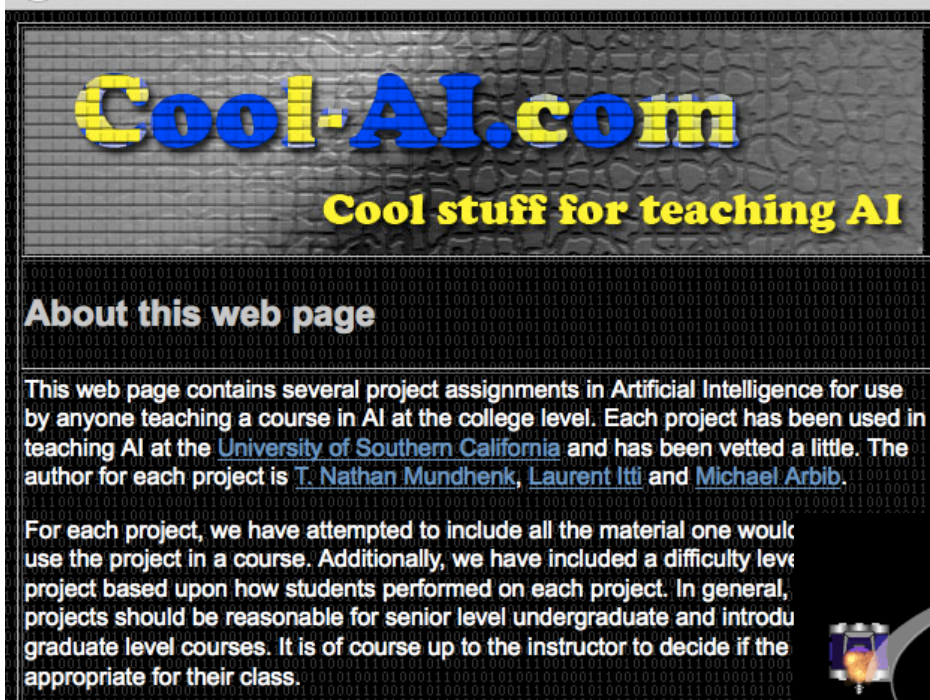
Benchmark of CutEM(d) level 2 and 3 Considering a Perfect Observable Scenarios

Criteria	CutEM(2) versus		CutEM(3) versus	
	Random Walk	Greedy	Random Walk	Greedy
Wins (%)	57940 (59,1%)	52651 (53,6%)	58262 (59,3%)	52858 (53,8%)
Draws	1948	1915	1851	1927
Average score difference in wins	236	231	238	236
Average score difference in losses	135	180	135	180

## Analysis of influential states

Actions of the Game	Average value (per match)	Percentage
Combining a hand's card with a board's card	10.2	71,4%
Discard a hand's card	4.1	28,6%
Combining a deck's card with a board's card	5.1	35,6%
Discard a deck's card	8.3	58,8%

## Related works



**Cool-AI.com**  
**Cool stuff for teaching AI**

**About this web page**

This web page contains several project assignments in Artificial Intelligence for use by anyone teaching a course in AI at the college level. Each project has been used in teaching AI at the [University of Southern California](#) and has been vetted a little. The author for each project is [T. Nathan Mundhenk](#), [Laurent Itti](#) and [Michael Arbib](#).

For each project, we have attempted to include all the material one would use the project in a course. Additionally, we have included a difficulty level project based upon how students performed on each project. In general, projects should be reasonable for senior level undergraduate and introductory graduate level courses. It is of course up to the instructor to decide if the appropriate for their class.



Many initiatives  
for interactive AI  
applications

Benchmark initiative

## *Conclusions*

- Amê is an environment to study and develop adversarial search algorithms for stochastic card games
- It is intended to be a laboratory for discovering and exploring search techniques
- We have conducted a preliminary study with computer science students
- It is a unique learning environment that provides opportunities to evaluate algorithms from a user perspective (Android application) and from a computer perspective (Benchmark application)



# Symposium On Applied Computing Track on Intelligent, Interactive and Innovative Learning Environments

## Amê: An Environment to Learn and Analyze Adversarial Search Algorithms Using Stochastic Card Games

Ana Beatriz Cruz, Sabrina Serique, Leonardo Preuss, Angélica Ogasawara,  
João Quadros, Eduardo Bezerra, Uéverton Souza, Eduardo Ogasawara



**Federal Center of Technological  
Education of Rio de Janeiro  
CEFET/RJ**

