



**Caribbean
Finals of the
ACM - ICPC**



The 2015 ACM-ICPC Caribbean Finals



Cuban Site hosted by UCI
Sponsored by UCI, Albet, OxiSold and XETiD
Havana, November 13th and 14th

3rd Caribbean Tournament of Virtual Players "ICPChallenge"

THE ICPCHALLENGE TOURNAMENT

The Tournament consists on programming virtual players capable of automatically playing, i.e. without human interaction. Each participant team of the 2015 ACM-ICPC Caribbean Finals may submit one virtual player implementation. A virtual player must be implemented in Java using a single file with the source code describing a single class. To submit a virtual player, the source code must be sent before November 8th to the address acm-icpc@uci.cu. The message should contain the subject: ICPChallenge "Team official name" (for example: ICPChallenge "UH++").

The implementation should be done considering a tournament with up to 6 players. The final tournament will be held with the 6 best implementations.

"THE CONQUERORS" GAME

"The Conquerors" game was intended to be easy to use and easy to understand its overall dynamics. In that sense it can be described quickly as a strategy tournament where each player has a finite group of buildings with a population that grows up to a limit. Each building can form armies compounds half of the population in the building when executing the action, with different characteristics of speed, attack and defense that dependent on the building type. Armies can be formed with two targets: attack an opponent building for conquer it, or reinforce an own building with more people.

THE GAME FIELD

The field is a two-dimensional space, on which the game is developed. The field is considered as a plane with 35 squares (points, spaces) in length and 16 in width. The buildings are placed at any point (X, Y) such that $0 \leq X \leq 34$ and $0 \leq Y \leq 15$.



The game field.

THE BUILDINGS

There are four types of buildings able to generate different kinds of armies. At the beginning, each player has four buildings, one of each type. There are N buildings on the field that belonging to the computer. Such buildings will not be able to attack nor increase his level.

Each building has a maximum population that depends on the level thereof. Each level offers to buildings a total of 10 people more in the population limit. To increase its level, a building must have at least half of the maximum population. Once the building is commanded to increase level, the number of people is reduced by half (if possible). The maximum level is 5 for each building.

Example:

Building with Level 1, maximum population of 10 people.

Current population is 9 (the building receives one command)

Building with Level 2, maximum population of 20 people; current population is 4.

CREATE ARMY

A player can have 5 armies by each building he owns. The armies are only created in the buildings. To create an army, the current population of the building is divided into two parts: one of them remains in the building and the other becomes part of the army.

Each army corresponds to a point of arrival. In case of own building, increases the population of the building in the same amount as the size of the army (the building is reinforced). Otherwise the combat is made. If the army wins, the building is moved to the winner player, restarting its level and updating as the current population the surviving soldiers of the attack.

FEATURES OF THE BUILDINGS AND SOLDIERS

SPEED

The speed of each army, when it moves, is defined as follows:

- COWHOUSE = 0.08 (pts. / iteration)
- FORGE = 0.02 (pts. / iteration)
- BARRACKS = 0.02 (pts. / iteration)
- CASTLE = 0.02 (pts. / iteration)

DEFENSE

The defense of the army in combat is defined as follows:

- COWHOUSE = 80
- FORGE= 90
- BARRACKS = 10

- CASTLE = 50

ATTACK

The attack of the army in combat is defined as follows:

- COWHOUSE = 50
- FORGE= 90
- BARRACKS = 10
- CASTLE = 70

SHELTER

The shelter represents an additional value that is added to each soldier in his defense and attack while he is protected inside a building:

- COWHOUSE = 20
- FORGE = 5
- BARRACKS = 2
- CASTLE = 30

RECRUITING TO OFFICIALS

In the game have been defined a set of actions that allow to collect coins, which can be used for recruiting to officers.

BUILDER

At a cost of 30 coins, the builder gives the possibility of constructing a new building that appears in a random position of the terrain. The player can choose the type of building to build. The field has a limit of 100 buildings.

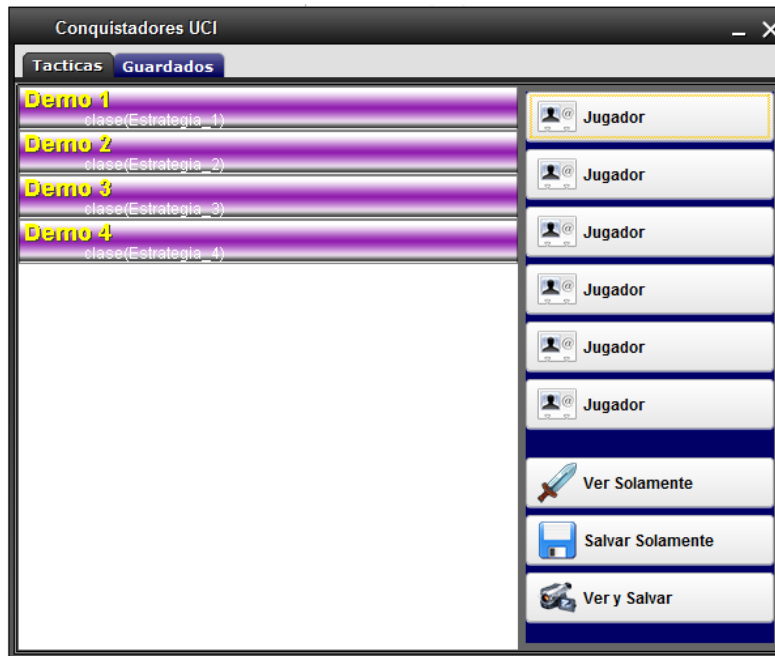
COACH

At a cost of 50 coins, the coach gives to all buildings and armies a plus in defense (+50) and attack (+50). The armies that were created before the purchase will not receive the plus.

THE CONQUERORS LIBRARY

The **uci.conquerors** library¹ implements a computational variant of the "Los Conquistadores" game and is distributed through the **conquerors.jar** file. The library documentation is also available².

With the command line **java -jar conquerors.jar** appears a window where you can choose up to 6 implemented strategies and play.



THE API

The main goal of the contestants is to develop a tactical of game, which should coordinate actions for attaining conquer all opposing buildings on the field. The API was designed for that purpose and is composed of a set of classes that allow the development of solutions. Those classes can be studied in more details in the available documentation.

¹<http://coj.uci.cu/downloads/final-carib/2015/uci.conquerors.rar>

²<http://coj.uci.cu/downloads/final-carib/2015/uci.conquerors.javadoc.zip>

IMPLEMENTING A PLAYER

The **uci.conquerors** library allows competition between virtual players that can be implemented using the library itself.

To create a tactic is only necessary to write a class that implements the interface **uci.conquerors.jugador.Jugador**. That interface contains two methods: **MyName** and **Execute**.

```
public interface Jugador {  
  
    /**  
     * Recibe los datos del estado del Juego y retorna las acciones del jugador  
     * @param ej Estado del juego  
     * @param et Estado de la Tienda  
     * @return  
     */  
    public List<Accion> Execute(EstadoJuego ej, EstadoTienda et);  
  
    /**  
     * Nombre de la implementacion  
     * @return  
     */  
    public String MyName();  
}
```

The **MyName** method should return the name of the tactics.

The **Execute** method is executed by the game engine at each iteration, updating (by parameters) to the tactics of the current situation of the game and providing information that can be found in the library documentation.

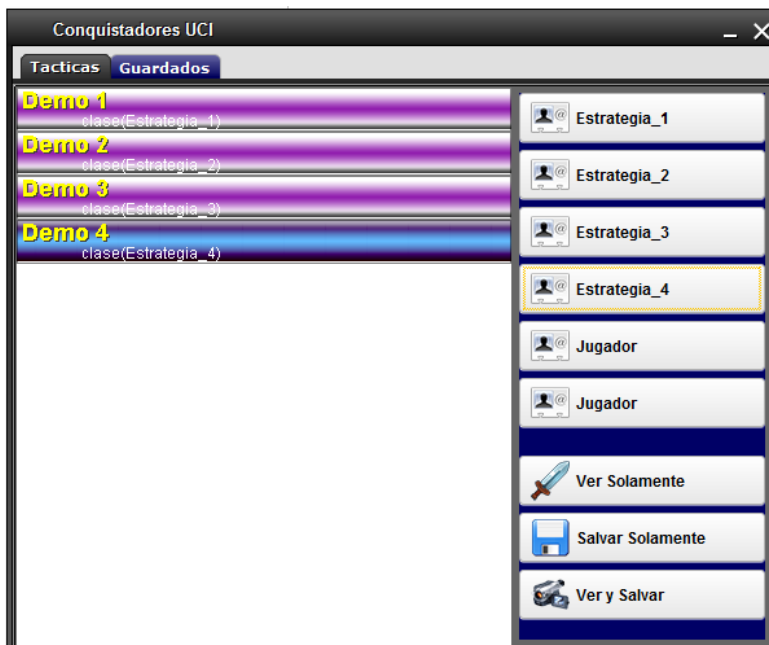
A tutorial video is also available³. Furthermore, some demonstrative classes are provided⁴.

³<http://coj.uci.cu/downloads/final-carib/2015/conquerors-videolow.rar>

⁴<http://coj.uci.cu/downloads/final-carib/2015/demo-players.rar>

PLAYING WITH THE VIRTUAL PLAYER

Once the virtual player is implemented, it is very easy to play with it: you should only ask the **uci.conqueros** library to start the game. After running the game, the following window is displayed with the different tactics implemented:



In the right area of the application, a tactic must be selected, and then execute commands with the button that corresponds to the action. Later appears a window with the simulation of the game.

CONTACT INFORMATION

In case of any doubt or concerns related to this document, please write to acm-icpc@uci.cu. You may also contact us directly:



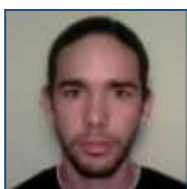
Dovier A. Ripoll Méndez

General Director
2015 ACM-ICPC Caribbean Finals
+53.7837.2762
daripoll@uci.cu



Tomás O. Junco Vázquez

Assistant Director, Cuban Site
2015 ACM-ICPC Caribbean Finals
+537.837.2709
tomaso@uci.cu



Ernesto A. Yero Oses

Director, "ICPChallenge" Tournament
2015 ACM-ICPC Caribbean Finals
yero@uci.cu