

ACM International Collegiate Programming Contest Asia Regional Contest 2006,  
Yokohama Japan at KEIO University

## Java Challenge Contest

### CodeBear - Strategy to get delicious salmons!

#### Contents:

- What is CodeBear?
  - About the challenge
  - Rule
  - Game screen
  - Objects in the game
- How to get archive, rewrite and compile the source, and run
  - For eclipse users
  - For non-eclipse users
  - Directory Structure
- How to implement your bear
  - Instructions
  - Sample code
- API document
  - Bear
  - BearInfo
  - Salmon
  - Block
- Submission
- Tournament Information

# What is CodeBear?

## About the challenge :

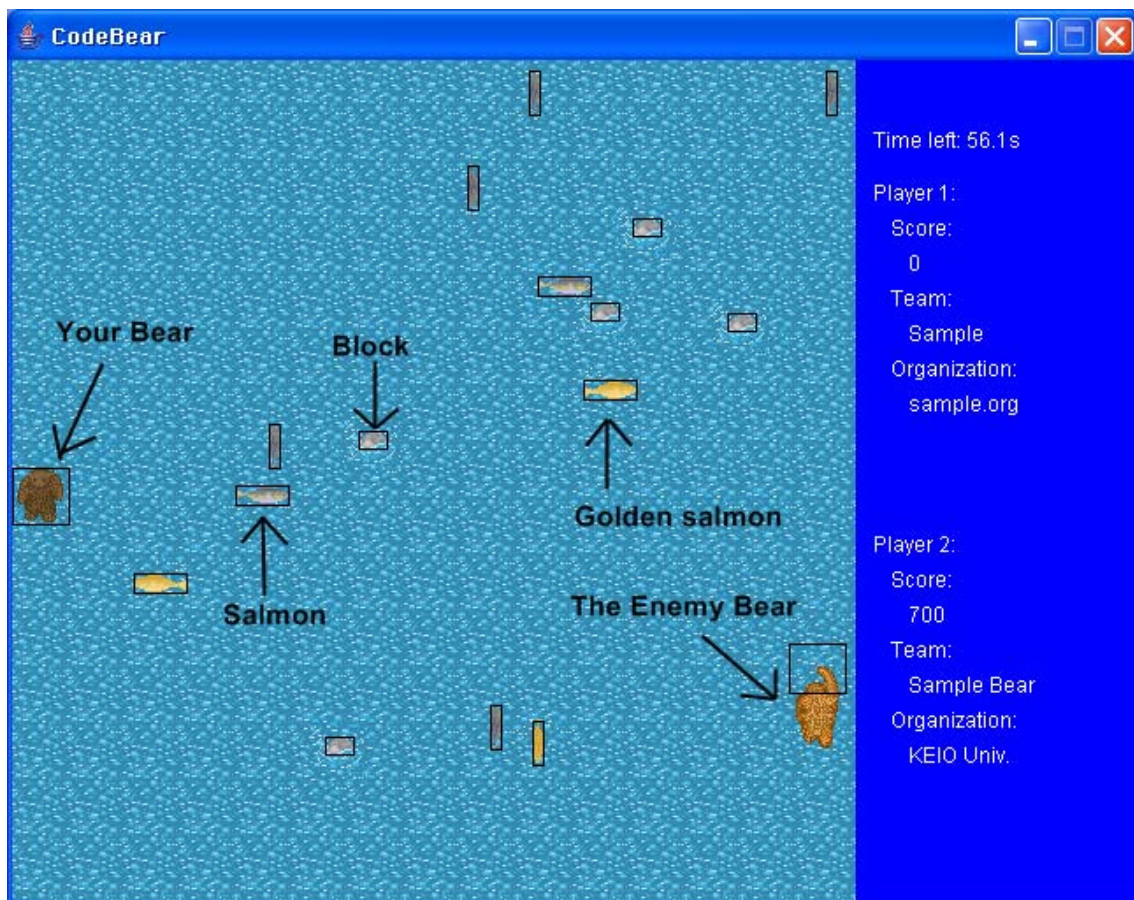
The main objective of the game is to get delicious salmons as many as possible by controlling your bear.

## Rule:

You can earn 100/500 points when you get a normal/golden salmon. You can win the game if your bear gets more points than the opponent within the time limit.

## Game Screen :

The game field size is 480x480 pixels. Bears and salmons move in this game field.



## Objects in the game:

- Your Bear: The bear you control. The size of the bear is 32x32.
- Opponent Bear: The bear the opponent team controls.
- Normal Salmon: You get 100 points if your bear catches a normal salmon.  
Primary food for the bear.
- Golden Salmon : You get 500 points if your bear catches a golden salmon. Golden salmon are more delicious but move more quickly than normal salmon.
- Block : Bears can not go over blocks.

## How to get the archive, rewrite and compile the source, and run

### For Eclipse Users:

Get and import the archive :

1. Get <http://server/CodeBear-1101.zip>
2. Invoke Eclipse. Select “File” and then “Import” from the menu.
3. Select “General” and then “Existing Projects into Workspace”. Click the “Next.”
4. Click “Browse” to the right of “Select archive file” and select the downloaded archive file.
5. Confirm “CodeBear” appears in “Projects” and it is checked. Then, click the “Finish”.

Implement your strategy for getting salmons:

Select “CodeBear” → “codebear.user” → “MyBear1.java”.

Rewrite MyBear1.java to implement your strategy for getting salmons.

Run your program :

Select “CodeBear” → “codebear.user.” Right-click on “Main.java.”

Select “Run as...” → “Java Application” to run the program.

### For Non-eclipse Users:

- 1 Get and save <http://server/CodeBear-1101.zip> in /home/user.
- 2 Execute the following command to expand the archive.
  - `unzip CodeBear-1101.zip -d /home/user`
- 3 Move to the directory, CodeBear, which is created under /home/user.
  - `cd /home/user/CodeBear`

Implement your strategy for getting salmons :

Rewrite codebear/user/MyBear1.java

Then compile the source by executing the following commands.

- `cd /home/user/CodeBear`
- `javac codebear/user/MyBear1.java`

Run your program :

- `cd /home/user/CodeBear`
- `java codebear.user.Main`

## Directory Structure:

- /home/user/CodeBear/ (or /home/user/workspace/ if you are an eclipse user.)
  - system/ : CodeBear System Library. You must not change any sources there.
  - user/ : CodeBear user's directory.
    - ✧ MyBear1.java : Your bear must be implemented here
    - ✧ MyBear2.java : A java source of a sample opponent
- Main.java : A java source to run CodeBear.

## How to implement your bear:

### Instructions :

- You rewrite codebear.user.MyBear1 class to implement the movement of your bear.
- Rewrite register() method to register your team name and affiliation. (Your team name and affiliation is displayed on the game screen.)
- Then, implement think() method.
  - think() method is periodically invoked from the CodeBear game engine.
  - Write the code in think() method that moves your bear by taking the game situation into consideration.
  - The CodeBear game engine provides detailed information on all salmons and blocks to think() method.
- A new salmon appears randomly in the game field if you get a salmon.
- The initial location of your bear is determined randomly in the Java challenge tournament. Don't write the program that depends on the initial location of your bear.
- To facilitate debugging, the CodeBear engine in the archive shows a box around a bear, salmon, and block. The CodeBear engine uses this box to judge collisions of bears, salmons, and blocks. These boxes are not displayed in the Java challenge tournament.

### Sample code:

```
package codebear.user;
import java.util.*;
import codebear.system.*;

public class MyBear1 extends MyBearBase implements IMyBear {
    public void register(){
        setTeamName("Sample"); // Name of your team
        setOrganizationName("sample.org"); // Your organization
    }

    int direction = 0;
    void think(ArrayList<Salmon> salmons, ArrayList<Block> blocks){
        //Sample: Only up-down moving thinking with blocks
        Iterator<Block> blockIterator = blocks.iterator();
        while(blockIterator.hasNext()){ // Checking all blocks
```

```

        Block b = blockIterator.next();
        if(this.hitTest(b)){ // Am I blocked?
            direction = 1;
        }
    }
    if(getTop()<=0){direction = 0;}
    if(getBottom()>=480){direction = 1;} /
    if(direction == 1){moveUp();}
    else moveDown();
}
}

```

MyBear2.java is a more complex example.

## API document:

### The APIs of your bear:

Mybear class has the following methods.

- int getLeft()
  - Returns the leftmost X-coordinate of the box around the bear.
- int getRight()
  - Returns the rightmost X-coordinate of the box around the bear.
- int getTop()
  - Returns the uppermost Y-coordinate of the box around the bear.
- int getBottom()
  - Returns the lowermost Y-coordinate of the box around the bear.
- Point getCenterPosition()
  - Returns the center point of the bear.
- boolean hitTest()
  - Returns true if the bear meets another object in the game field. Otherwise, returns false.
- BearInfo getInfo()
  - Returns a BearInfo object regarding your bear.
- BearInfo getEnemyInfo()
  - Returns a BearInfo object regarding the opponent bear.

BearInfo class has the following methods.

- Point getCenterPosition()
  - Returns the center point of the bear.
- int getScore()
  - Returns the current score of the bear.
- String getTeamName()
  - Returns the team name.
- String getOrganizationName()
  - Returns the organization name.

Note : The following methods are for moving a bear. Only the first invocation of these methods is effective during one invocation of think() method. Once your bear starts moving by calling moveXX() method, it continues moving unless you explicitly invoke stopMove() or attackSalmon().

- void moveUp()
  - Moves a bear upward.
- void moveDown()
  - Moves a bear downward.
- void moveLeft()
  - Moves a bear to the left.
- void moveRight()
  - Moves a bear to the right.
- void stopMove()
  - Stops moving a bear.
- void attackSalmon()
  - Tries to catch a salmon in the direction your bear is heading in.
  - If there is a salmon within the bear's arm reach, the salmon is caught.
    - ◇ The length of the bear's arm is 20 pixels. (See Fig.1.)

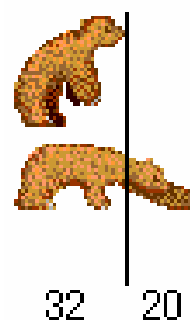
### The APIs of salmon:

A Salmon class object represents a salmon.

“ArrayList<Salmon> salmons” is passed to think() method as parameter. You can get a Salmon object by calling get() method on “salmons.”

Salmon class has the following methods:

Fig.1 The size of your bear and his arm



- `int getLeft()`
- `int getRight()`
- `int getTop()`
- `int getBottom()`
- `Point getCenterPosition()`

You can get the location of a salmon by calling these methods in the same way as bears.

- `int getScore()`
  - Returns the point you can get if this salmon is successfully caught.

### The APIs of block:

A Block class object represents a block in the game field. “`ArrayList<Block> blocks`” is passed to `think()` method as parameter. You can get a Block object by calling `get()` method on “`blocks.`”

Block class has the following methods :

- `getLeft()`
- `getRight()`
- `getTop()`
- `getBottom()`
- `getCenterPosition()`

You can get the location of a block by calling these methods in the same way as bears.

## Submission:

After the Java Challenge program development, the code the participants wrote is collected. Note that only the following file is collected:

- `/home/user/workspace/codebear/user/MyBear1.java` (For eclipse users.)
- `/home/user/CodeBear/codebear/user/MyBear1.java` (For non-eclipse users.)

## Tournament Information:

Java Challenge is done in tournament play. The time limit is one minute. After the quarter final matches, the time limit is two minutes. If the game is draw, another round of a match is played. If the match is draw again, the winning team is selected randomly.