

CSC7322: Object Oriented Development

J Paul Gibson, A207

`paul.gibson@it-sudparis.eu`

<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7322/>

Assessed Project

[.../~gibson/Teaching/CSC7322/Project-CSC7322-2011.pdf](http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7322/Project-CSC7322-2011.pdf)

Overview

The application to be developed is a pedagogic tool to aid in the teaching of programming concepts to school children.

The goal is to provide a framework for programming artificial players of simple games. Our framework will support the following steps which we follow when programming a game with the children:

- Identify the rules for play
- Simulate random play
- Add intelligence by formulating rules for playing
- Illustrate the importance of ordering the rules
- Allow children to program using the rule
- Allow the children to see how well their program plays
- Introduce the children to Java
- Implement the children's rules in Java

Learning Objectives

This project is concerned with testing that you have met the following learning objectives:

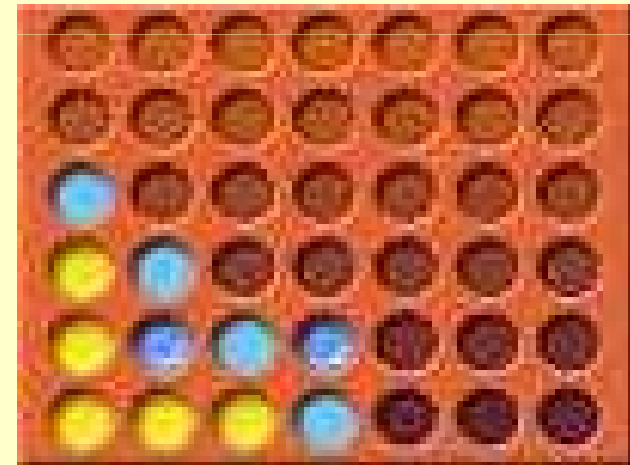
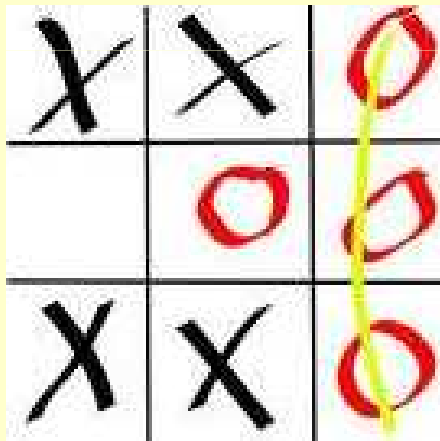
- Understand fundamental OO concepts
- Use OO modelling to represent requirements
- Use OO modelling to evaluate alternative designs
- Transform an OO design into OO code (written in Java)
- Transform requirements into test code (written in Java)
- Demonstrate ability to re-use different types of artefact –
 - Code/tests
 - Designs/Patterns/Architectures
- Demonstrate ability to build reusable artefacts.

Two Games

In order to demonstrate the re-use of artefacts in your development,
You will be required to develop two game applications:

Noughts and Crosses

and Connect-4



Development Steps: for each game

1. A text/terminal based game with a **random AI player** and 3 modes of play:
 - I. AI vs AI
 - II. Human vs AI
 - III. Human vs Human
2. A text/terminal based game (with 3 modes of play) where **the programmed AI player(s)** can be read in from a text file in which a sequence of rules is specified. For example:
 1. *If you can win then win*
 2. *If you can block a win then block a win*
 3. *If the center is free then play in the center*
 4. *Choose one of the remaining free positions randomly*
3. You must facilitate a library of different players and be able to select any of these players for the AI. Note that you should be able to play 2 different types of player AI against each other. Note also that each rule must be implemented in Java (otherwise the AI player will not be able to play)
4. A GUI for playing the game
5. A GUI for programming a player (as a sequence of rules from a library)

Marking Scheme (100%):

XO

Connect 4

Correctly functioning code (50%)

- | | | |
|--|---|---|
| 1. A text/terminal based game with a random AI player and 3 modes of play | 5 | 5 |
| 2. A text/terminal based game (with 3 modes of play) where the programmed AI player(s) can be read in from a text file in which a sequence of rules is specified. | 5 | 5 |
| 3. You must facilitate a library of different players and be able to select any of these players for the AI | 5 | 5 |
| 4. A GUI for playing the game | 5 | 5 |
| 5. A GUI for programming a player (as a sequence of rules from a library of rules) | 5 | 5 |

The focus of the project is on **good design for re-use (50%)**:

- | | |
|--|----|
| 1. Re-use between the 2 different games | 30 |
| 2. Re-use from secondary sources (properly acknowledged and tested!) | 20 |
| 3. There is a bonus of 10 marks for the development of code that can be reused outside of the games project | |

Assessment Rules

- You are to work on this project alone.
- If you get help then you are to acknowledge this in your code/design/test documentation. You can help a colleague but if you do so then you must verify that this is acknowledged in their documentation.
- If you re-use code/design/models then you must fully acknowledge what you used, and will be rewarded for such re-use provided that what you re-use is *doing what it is supposed to do*. You will be penalised for re-using something without demonstrating its suitability
- You should document your code as best you can.
- You should test your code as best you can.
- You should document your design and comment on decisions that were taken to facilitate re-use.
- You must submit all documentation and source code/models on/before Monday 28th March.

NOTE: As part of the assessment procedure you may be asked to attend an interview where you will be questioned about the project.