School of Computer Science

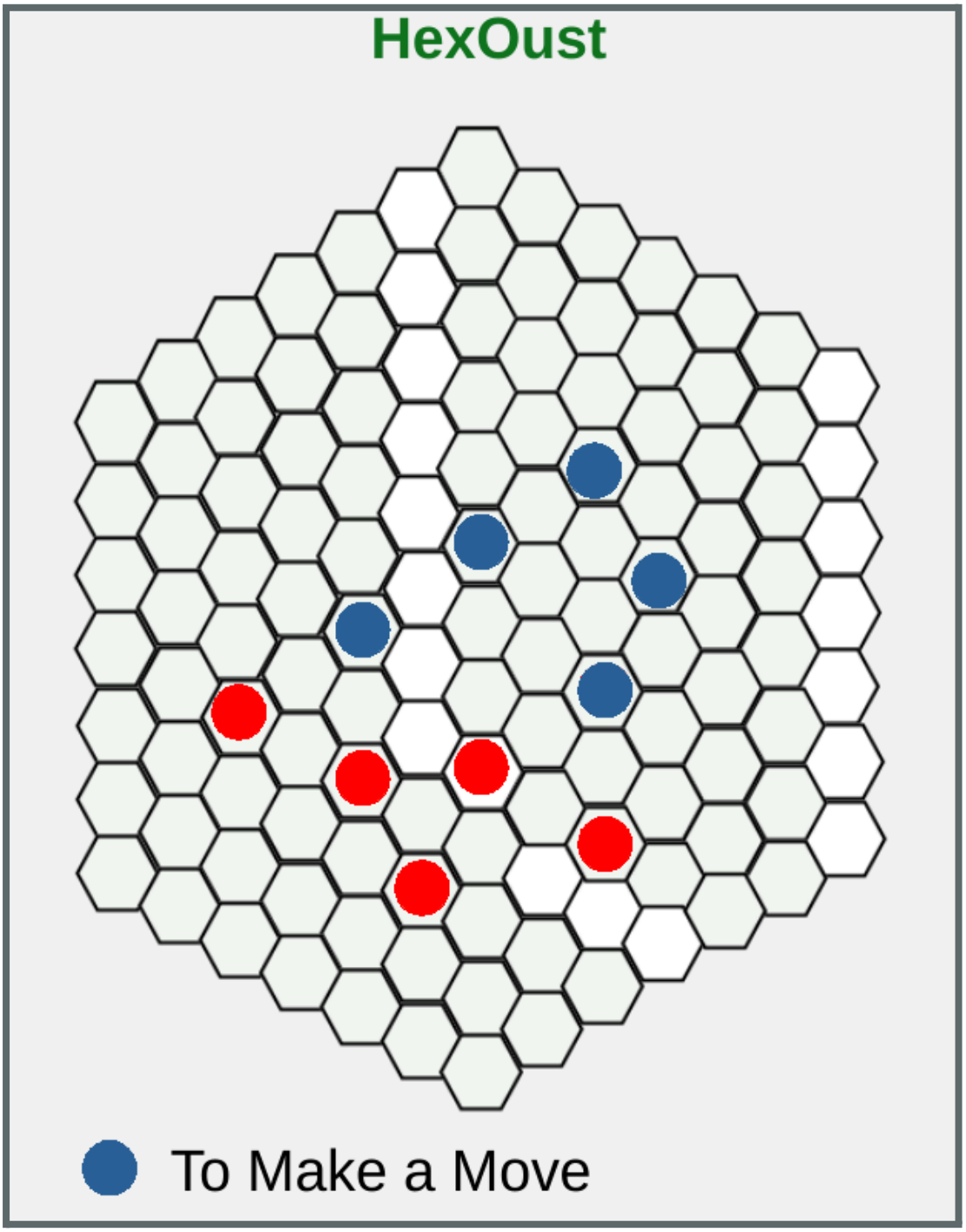
COMP20050: Software Engineering Project-II

Project Handbook

Module Coordinator	**Ravi Reddy Manumachu**
Email	**ravi.manumachu@ucd.ie**
Date	**2025 Spring**

Introduction

In this project, you (in a group of three members) will develop a software implementation of the board game, **Hex Oust (base-7 Hexagonal Board)**, using Java programming language.



The rules of the game are in the document on BrightSpace in the link below:

My Learning > Software Project Materials
https://brightspace.ucd.ie/d2l/le/content/249619/viewContent/2853914/View

Please follow the steps below before commencing the project.

The following are regarding selecting your group:

- Self-select your group at **My Class > My Groups** on **Brightspace**.
- After the group selection expiry date (next week), any class members who have not self-selected will be automatically and randomly allocated to a group.

Follow the instructions below for project source version control:

- Git and GitHub will be the source code repo and version control for this project.
- Setup Git on your computer. Setup your GitHub account online.
- Setup a GitHub repo for the project. **THIS REPO MUST BE PRIVATE.**
- The repo name should include your Group numbers allocated on Brightspace.
- Share the repo with your team members.
- Give repo access to our TA (user name: **gillanimaryam**) and the Module Coordinator (user name: **ravimanumachu**).
- **GitHub must be used for source code control for the duration of the project.**

All your submissions will be via Brightspace.

Project Calendar

Week	Week Starting Date	Project Activity	Deadlines
1	21 Jan	Group creation, GitHub account setup, Software Project Introduction	
2	29 Jan	Software Architectural Design Using Problem-based Learning (PBL) approach	
3	5 Feb	Software Architectural Design continuation, Formulating Project Plan divided into four Scrum sprints	**10 Feb** **Architecture design submission,** **Project plan submission**
4	10 Feb	Sprint 1	
5	17 Feb		**23 Feb** **Sprint 1 submission**
6	24 Feb	Sprint 2	
7	3 Mar		**9 Mar** **Sprint 2 submission**
	9 - 23 Mar	***Study Week***	
8	24 Mar	Sprint 3	
9	31 Mar		**6 Apr** **Sprint 3 submission**
10	14 Apr	Sprint 4	
11	21 Apr		**20 Apr** **Sprint 4 submission**
13	28 Apr	Project Conclusion	**5 May** **Final Project Submission**

Software Architectural Design and Project Plan (Architectural Sprint)

You will propose a software architectural design. You are allowed to use a modelling tool (like UML) or diagram software for this activity.

You will use your design to formulate a project plan that breaks the development of the project into four sprints.

Following are the Brightspace submission details for the architectural design and project plan:

- Submit a zip file with the name, **<yourgroupnumber>_<yourgroupname>_hla.zip**.
- A **report** in PDF format (1 page) including:
    - Group number, name, and Student IDs.
    - State the relative amount of work done by each group member. If the workload distribution is not evenly divided among the members, please explain why.
    - **If the workload distribution is even, state "All members contributed equally".**
- Software architectural design document (pdf).
- Project plan document (pdf).

You will be given a rubric containing analytical criteria on how this submission is graded.

Sample Project Plan

A sample project plan is provided below. The features are mapped to the requirements presented in Lecture 2 on Software Requirements.

Sprint	Features
1	1). A feature that implements the board layout. **(SR1)** 2). A feature displaying a **RED** sphere followed by a text "**To Make a Move**" to indicate the **RED** player's turn or a **BLUE** sphere followed by the same text to indicate the **BLUE** player's turn. This placement need not be *necessarily below the board layout*. **(SR1.3)** 3). A feature that allows exiting the game at any point. For a GUI, the user can close the window to terminate the game. For a TUI, the user is given an option to quit. **(SR1.4)**

2	Following are features for *Non-capturing Placements (NCP)* and *Invalid Placements*: 1). A feature allowing a player (**RED** or **BLUE**) to place her stone in a valid cell by drag-and-drop or clicking the cell. **(SR2)** 2). A feature where if the player makes an NCP, a stone of the opponent player's color followed by a text shall be displayed to indicate the opponent player's turn. **(SR2.2)** 3). A feature that displays an error message if a player (**RED** or **BLUE**) places her stone in an invalid cell. The error message shall be **Invalid Cell Placement.** **(SR4)**
3	Following are features for *Capturing Placements (CP)*: 1). A feature where if the player makes a CP, then all the opponent's groups that are captured shall be removed from the board. **(SR3.1)** 2). A feature where if the player makes an CP, a stone of the capturing player's color followed by a text shall be displayed to indicate the capturing player's turn. **(SR3.2)** 3). A feature covering interesting cases of capture of opponent's groups (see Lecture 2 slides on CP).
4	Following are features for *Winning Move*: 1). A feature where if a player (**RED** or **BLUE**) places her stone that eliminates all the opponent's groups, then a message then a message shall be displayed mentioning that the player won (**RED** or **BLUE**). **(SR5)** 2). A feature that displays the message **RED wins** if the RED player wins. Otherwise, **BLUE wins**. **(SR5.1)**

Sprint Submissions

The submission details for each sprint follow:

- Submit a zip file with the name, **<yourgroupnumber>_<yourgroupname>_sprint<N>.zip**, Where <N> stands for the sprint number.
- A **report** in PDF format (1 page) including:
    - Group number, Group name, and Student IDs.

    - GitHub link.
    - State the relative amount of work done by each group member. If the workload distribution is not evenly divided among the members, please explain why.
    - **If the workload distribution is even, state "All members contributed equally".**
- An **executable JAR file** of the board game.
- A **document** on what features have been developed and tested.
    - Note the software implementation for a sprint should have all the features planned in the sprint working and tested well.
    - Specify any deviations from the project plan such as what features have been removed or what new features were added.
- **GitHub or Trello Kanban Board** for the sprint for each team member.
- A **document** on instructions on how to launch and play the board game.

Note that the report would typically be the same for all sprints for a well-designed project unless there is a change in sprint activities and member contributions. *Therefore, you can just submit the same report if there are no significant changes.*

You are provided a rubric containing analytical criteria on how a sprint submission is graded.

Final Project Submission

You will be given a rubric containing analytical criteria on how your final project submission is graded.

Apart from the working functionality, all of the following will be considered towards grading:

- **Code quality:**
    - Project is well structured into classes, classes are well structured into methods (small classes, highly cohesive classes).
    - Clean code practices for functions (small functions, stepdown rule, DRY principle).
    - Naming conventions are clear and meaningful.
    - Comments are useful.
    - Proper formatting (size of classes, length of lines).
- **Test quality:**
    - Clean testing principles (F.I.R.S.T, Use of asserts).
    - A robust test strategy with **JUnit** for at least 2 non-trivial classes.
- **Documentation:**

    - **Javadoc** used correctly for at least 2 non-trivial classes.

The final project submission details are below:

- Submit a zip file with the name, **<yourgroupnumber>_<yourgroupname>_final.zip**.
- A **report** in PDF format (1 page) including:
    - Group number, name, and Student IDs.
    - GitHub link.
    - State the relative amount of work done by each group member. If the workload distribution is not evenly divided among the members, please explain why.
    - **If the workload distribution is even, state "All members contributed equally".**
- An **executable JAR file** of the board game.
- A **user manual** instructions on how to launch and play the board game.
- A **video** (max 5 minutes) in .mp4 format of a screen recording (with voice over if you like). The video should contain the following:
    - Run the board game showing the working features.
    - Give a high-level code walkthrough.
    - Explain your test strategy.

Grading Scheme

Assessment	Marks
High-level Design	15
Project Plan	5
Sprint 1 Submission	10
Sprint 2 Submission	10
Sprint 3 Submission	10
Sprint 4 Submission	10
Final Project Submission	40

The numerical value is converted to a grade using the Alternative Linear Conversion Grade Scale 40% Pass.

https://www.ucd.ie/students/exams/gradingandremediation/understandinggrades/

Please read the following policy on late submission.

Late Submission of Coursework Policy
https://brightspace.ucd.ie/d2l/le/content/249619/viewContent/2824569/View

- Coursework submitted up to **10 working days** past the due date will be graded but a **grade penalty** will be applied.

- Coursework submitted at any time up to and including **5 working days after the due date** will have the grade awarded reduced by **one grade point**.

- Coursework submitted more than **5 working days but up to and including 10 working days** after the due date will have the grade reduced by **two grade points**.

- Coursework received more than **10 working days after the due date** will **not be accepted or graded**.

Following are the other penalties:

- Based on review of the comments in the report on the relative work done and on review of the group member contributions on GitHub, the grades for individual students in a group may be adjusted.
- GitHub not used, up to **2 grade point deduction**.
- For the final project submission,
    - No report will result in **1 grade point deduction**.
    - For the final project submission, no video has a **1 grade point penalty**.
    - No JAR file has a **1 grade point penalty**.

Final Comments

Do not delete or make public your GitHub repository until the end of the semester plus 3 months.