**Name:** Thomas Fisher
**Student Number:** 831021

# CS-235: Assignment 6 – Assignment Report

My game suite application is built on Scott-Cheg (Group 2)'s application (my group in A5).

## Compiling Source Code

To compile the project, import it into eclipse / other IDE as there are multiple packages.

To run the application, use MainMenu.java in the gamesuite package.

## Implemented Features

The following section states the features that I have implemented that are asked for on the feedback form.

### Working Features

**New Functionality**

A user can play a game of Ludo to completion

The user can switch back and forth between Ludo and Minesweeper

Up to 4 machine players are supported for Ludo

The game logic for Ludo

The user can set an interval timer between computer player moves for any real number between 0 and 10 seconds.

**Functionality from A5**

The user can play a game of Kablewie to completion

The game logic for Kablewie

Both AI and Human players are supported

Games can be saved and re-loaded

**Common Functionality**

The winner of each game is indicated clearly at the end of the game

The number of active / played pieces for each player is shown

The elapsed play time is shown

The software has instructions for each game in the menu bar

**Name:** Thomas Fisher
**Student Number:** 831021

### Not-Working Features

Kablewie tile animation - The animated reveal of the tiles causes a bug with the AI. If the AI is quicker than the rate of the animation, it can make moves mid-animation. This causes problems with the reveal algorithm. The code has been left in, but a non-animated reveal is being used while this doesn't work.

## Non-Implemented Features

The Ludo pieces are not currently animated when they move. Unfortunately my biggest mistake of this coursework was designing the software with animation as an after-thought, thinking it would be easy to add in. The way I've made pieces move, the location is only updated after the whole move is made. This prevents my initial plan for animation which was to render the pieces after each move and then wait a short amount of time before going to the next space. To go back and change how I move the pieces now, at the time of completing all the other sections of the assignment would be too time consuming to have a finished program in time.

Besides this all the features on the feedback form have been implemented.

## List of Classes

### New Classes for A6

**'gamesuite' Package**

Includes features that are common to various different games (including Kablewie and Ludo).

New classes:

- Board
- Dice
- GameAi
- MainMenu
- Player
- Tile

**'ludo' Package**

Includes the features required for ludo.

New classes:

- HomeArea
- HomeAreaTile
- HomeColumnTile
- LudoAi
- LudoBoard
- LudoController
- LudoPlayer
- LudoSettings
- LudoTile
- NormalTile
- Piece
- StartingArea
- StartingAreaTile
- StartingSlot
- StartingTile

**Name:** Thomas Fisher
**Student Number:** 831021

## Modified Classes from A5

For some classes (such as Tile, Board, Player, ComputerPlayer), a more abstract version was created so that they could be re-used for games other than Kablewie (not just Ludo). The original classes were then changed to implement or be subclasses of these.

Classes with notable changes:

KablewieBoard – Now uses gamesuite.Board, a more abstract, re-usable board concept.

KablewieAi – Now uses gamesuite.GameAi, an interface that includes the methods common to any game AI.

KablewieSettings – Fixed hard to look at / read colour scheme.

Also, a big change that I made to all the classes is that the board used an ArrayList of an ArrayList of tiles, which made things very messy when trying to retrieve a particular tile. This has been changed to a 2D array of tiles to make things easier to understand.

Besides these classes, almost all of the other classes were changed in smaller ways, such as removing violations of coding conventions that were overlooked in A5.