

CW2 Software Maintenance Spec Sheet

Academic Year 2018/2019

CW2 is about maintaining and extending a modern version of a classic retro game (Pacman). You can play around with a clone of the original game downstairs at one of the play stations at the back door. The modern version has never been completed, but at least it runs.

To get started download zip file "BestPacmanEverV5_src.zip". The zip file only contains a "src" folder, so you have to set up your own JavaFX project in your preferred IDE and import the src folder.

The marks will be split as follows:

- 15% for git use
- 30% for refactoring
- 30% for additions
- 15% for documentation (Javadocs + class diagram)
- 10% for a marketing video

You should roughly spend 50 hours on this coursework

What to produce:

- There is no report to be written. The documentation should be delivered in form of a Javadoc folder inside your project folder. We will mainly look at the Javadocs to find out how you maintained and extended the game. If it is not obvious from there we might miss it. Also we have only a limited amount of time to look at each coursework submitted. So, please make sure to provide informative but concise Javadocs.
- In addition you need to provide a high level class diagram that shows the structure of the final version of your game (considering only classes, interfaces, relationships, and multiplicity). If you use software to reverse engineer your class diagram, make sure the delivered diagram is correct and follows the above requirements.
- Finally you have to make a video showing your software in action and demonstrating and explaining its functionality. You could also highlight 2-3 achievements you are very proud of.

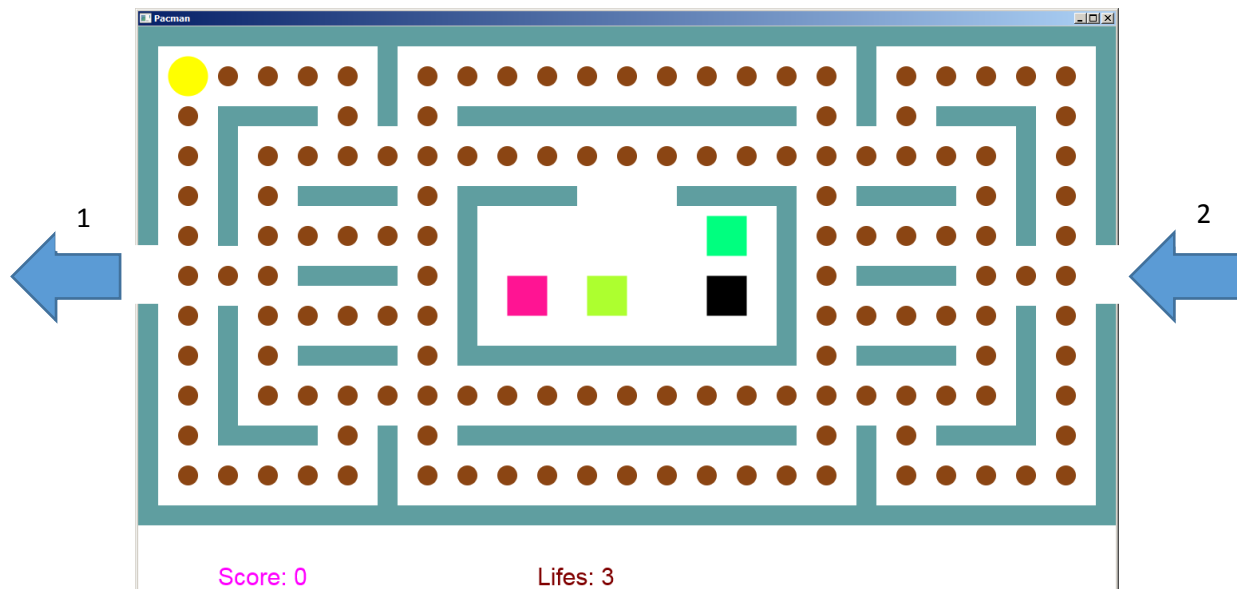
Important:

- This coursework is about maintaining and extending existing code. So, for the maintenance part you have to use the existing code as a basis, and not write your own Pacman game from scratch.

For moderate marks:

- Set up a git repository on the school's git server <https://projects.cs.nott.ac.uk/> and use it for version control activities
- Do some basic maintenance of the delivered code base (e.g. adding meaningful Javadocs, organising files in a meaningful way into packages, breaking up large classes in a meaningful way to support the idea of single responsibility, improving encapsulation)
- Extend the delivered code base by adding:
 - A START screen, displaying a picture related to the game and a button that allows going to a SETUP screen
 - A SETUP screen: Allowing to choose background and wall colour for the game field (allowing a choice of at least 8 colours) and a button to go back to the START screen
 - A high score popup, appearing at the end of each round, showing the scores from each round, highest at the top

- Two doors at the side of the playfield to allow the pacman and the ghosts to transit between them (leaving at 1 and immediately reappearing at 2 and vice versa).
 - See figure below and ask during the labs if this is not clear



For higher marks, in addition to the previous:

- Refactor the code by adding some design patterns to enhance maintainability
- Considering organising the code to adhere to the MVC pattern
- Create a permanent high score list (using a file to store scores)
- Load level descriptions from file and allow running the game with different layouts
- Load proper game character pictures
- ...

<https://powcoder.com>

Add WeChat powcoder

You can use either Java 8 or Java 10 for the implementation.

The project files you are submitting need to be either compatible with Eclipse or IntelliJ.

Deliverables: Three separate files, uploaded to Moodle:

- A zip file containing your entire project (including your Javadocs). It needs to be possible to import your project into one of the IDEs on the lab machines in A32 (either Eclipse or IntelliJ) using the import wizard of that IDE. So please test your file there before submitting it, to avoid disappointment. Please name your zip file as follows: "Surname_FirstName_IDE.zip", where IDE represents the name of the IDE you used (i.e. either Eclipse or IntelliJ).
- Your final class diagram as pdf. Please name your pdf as follows: "Surname_FirstName.pdf"
- A video of up to 2 minutes, as described above, in a common format that can be played on the lab machines in A32 Please test your video on the lab machines before submitting it, to avoid disappointment. Please name your pdf as follows: "Surname_FirstName.EXT", where EXT represents the extension related to your video format (e.g. mp4; mpg; avi).

Note: You are gently reminded that we are at liberty to use plagiarism detection software on your submission. Plagiarism will not be tolerated, and academic offences will be dealt with in accordance with university policy and as detailed in the student handbook. This means you may informally discuss the coursework with other students but your submission must be your own work. Please also note that it is not permitted for you to copy and paste text from another source without correct referencing. If you are unclear about this process please discuss with the module convenors during one of our lab sessions or at the end of a teaching session.

Assessment Details:

- This coursework is worth 50% of this module and as such is marked out of 50.

Questions:

- Dr Siebers and Dr Greensmith will of course be happy to answer questions and give high level interim formative feedback on your assignment during the lab sessions.

Good luck, have fun, code well :).

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder