3.3 Assessment 1: Greenfield Development, due: noon, Sem1/7/Mon

Assessment 1 covers the main stages of the software engineering lifecycle – from requirements elicitation to implementation – in a greenfield setting. As part of the assessment, each team will need to elicit requirements, develop an architecture, identify appropriate software engineering methods and techniques, identify risks and their mitigation, and implement a first version of the system specified in the scenario.

3.3.1 Deliverables for Assessment 1

Your team will submit a website plus a single .zip file. The requirements for the deliverables to be included in the zipfile are summarised in Table 1.

Table 1: Assessment 1 zipfile contents

	Deliverable	**Max. mark**	**Page limit**	**File name and format**
1.	Website (submit only the URL)	3	—	url1.txt
2.	Requirements	20	$1 + 3$	Req1.pdf
3.	Architecture	22	6	Arch1.pdf
4.	Method selection and planning	10	$2 + 1 + 2$	Plan1.pdf
5.	Risk assessment and mitigation	10	$1 + 3$	Risk1.pdf
6.	Implementation	25	1	Impl1.pdf + Code + Executable JAR

1. Website [3 marks]

   a) The submitted URL must link to the website that is the “public face” of your team’s project, and will be updated as you proceed.

   b) The website must include links to all the PDF documents listed in Table 1, to the executable JAR of your game, and to the version control repository of your team’s code in a clear and accessible way.

   c) The “management” and other teams can use the website at any time after the submission deadline to access the material above.

   d) In this deliverable, it is the website structure that is marked. You will be penalised if material is not easily locatable and accessible.

2. Requirements [20 marks]:

   a) Write a succinct introduction explaining how requirements were elicited and negotiated, and why they are presented as they are. Your submission should evidence research into requirements specification and presentation (4 marks, $\leq$ 1 *page*).

   b) Give a systematic and appropriately-formatted statement of user and system requirements (16 marks, $\leq$ 3 *pages*).

   Note that you will need a requirements referencing system, and may need to update this for subsequent assessment deliverables.

3. Architecture [22 marks]:

   Give diagrammatic representations (structural and behavioural diagrams) of the architecture of the team's product, with a brief statement of the specific languages (for instance, relevant parts of UML) and the tool(s) used to create these representations. Include a systematic justification for this architecture and describe how it was initially designed and how it evolved over the course of the project. Provide evidence of the design process followed (e.g. interim versions of architectural diagrams, CRC cards) on your team's website and link to them from your report. Relate the architecture clearly to the requirements, using your requirements referencing for identification, and consistent naming of constructs to provide traceability (22 marks, $\leq$ 6 *pages*).

4. Method selection and planning [10 marks]:

   a) Give an outline and justification of the team's software engineering methods, and identify any development or collaboration tools that the team has used to support the project or the team working. Justify the fitness of the selected tools with the team's software engineering methods and discuss alternatives considered. (3 marks, $\leq$ 2 *pages*).

   b) Outline the team's approach to team organisation, and explain why the chosen approach is appropriate for both the team and the project (2 marks, $\leq$ 1 *page*).

   c) Give a systematic plan for the project. Your plan should lay out the key tasks, their starting and finishing dates, as well as task priorities and dependencies. Provide weekly snapshots of the plan on your team's website and discuss how the plan evolved throughout the duration of the project (5 marks, $\leq$ 2 *pages*).

5. Risk assessment and mitigation: [10 marks]

    a) Describe and justify the risk management process followed by your team and the format of your team's risk register (3 marks, $\leq 1\ page$).

    b) Give a systematic tabular presentation of risks (risk register) to the project, their likelihood, impact, mitigation and ownership (7 marks, $\leq 3\ pages$).

    ENG1 is a small project, developing non-critical software. Keep your likelihood and impact measures simple.

6. Implementation [25 marks]:

    a) Provide documented code for a working implementation of the part of the game that meets the remit, requirements and architecture for Assessment 1. The code and an executable JAR of the game, that includes all external dependencies, must be included in the zipfile. (20 marks)

    b) List any 3rd-party libraries or assets you may have used in your implementation and the licenses under which they are made available. Briefly discuss the suitability of these licenses for your project. State explicitly any of the features required for Assessment 1 that are not (fully) implemented, using your requirements referencing for identification, and consistent naming of constructs to provide traceability. (5 marks, $\leq 1\ pages$)

See Section 2 for information on peer assessment. [10 marks]

You may use additional pages for a bibliography. Any other content that overruns the page limit will not be considered by the markers and will not receive any marks.