University of York

Department of Computer Science

SEPR - Assessment 2

Updates to Assessment 1 Deliverables

Team Craig

Thomas Burroughs

Huw Christianson

Joseph Frankish

Isaac Lowe

Beatrix Vincze

Suleman Zaki

Changes to Requirements

Our requirements now reference Sommerville's stages of requirement elicitation and specification[1] to show our research into requirement elicitation. Following this process the team was able to elicit more precise, unambiguous requirements. Our choice of following Sommerville's elicitation process was justified as it allowed us to abide by our chosen agile methodology by prioritising face-to face meetings and working software.

Although the actions we took to elicit our requirements remained the same we have added justifications and reflections on our choices. For example, justification on our brainstorming method was included to show how the group collaborated to elicit requirements and how this method helped to consolidate our vision for the project's future. We have also added justifications for our inclusion of paper prototyping which was included after research into the IEEE practice of SRS[2]. This practice outlined that prototyping provides quick feedback from the customer as they don't have to read over the full SRS. Also, prototyping helped to elicit unanticipated requirements and ensure that all members had similar visions for the project's future. On reflection, both prototyping and brainstorming proved to be useful strategies for engaging all team members in design decisions, even those less vocal, as well as facilitating communication with the customer.

We also added statements of reflection for particular strategies implemented for requirement specification. For example, our choice to abide by the IEEE practice and inclusion of SSON helps to provide structure and consistency. However, it can also hamper creativity as stricter protocols have to be adhered to, restricting creative freedom. Our reflections may help other teams taking over the project to suitably adapt their requirement specification procedures to avoid problems our team faced. Our feedback suggested that our use of the IEEE practice for SRS [2] was good but justification was needed as to why not all of the practice was used. Our requirements now include detailed justification on our choice to use IEEE's practice and how this allows us to elicit 'good' requirements. This practice helps to factor out inconsistencies which proved detrimental in the precision of our requirements from Assessment 1. Further justification was also added about the omittance of certain aspects of the IEEE practice [2] e.g. security and safety requirements. Changes to the layout of the SRS have also occurred to follow the layout of the IEEE practice for SRS template more strictly. Our SRS now includes a greater range of IEEE's conventions e.g. purpose and further document conventions, including the clarification between 'must' and 'should' in the requirement description. This should help the overall readability of the document for non-team members (i.e customer) and should ease the transition for any group acquiring our game should there be a need for the SRS to evolve.

REQ	Changes
General changes	Requirement Descriptions: Used words "must" and "should" more discerningly. Must is used for integral requirements, while "should" is used for more optional requirements. Requirement ID: Ordered Requirements by type to enhance readability
F1	Fit Criterion: Changed to define what a "novel mechanic" is in terms of the project, and how a "novel mechanic" might affect the project on a wider scale, as the wording was nebulous even to the group.
F2	Fit Criterion: Clarifies that the main game cannot affect the mini-game, and that the two games are distinct, to clear up misunderstandings within the group
F3	Fit Criterion: Attempt to clarify the use and effects of powerups.
F5	Fit Criterion: Clarify what constitutes "different" characters because it was undefined
F6	Fit Criterion: Clarify the differences between a "regular" and "boss" enemy because it was unclear Risks: Clarify the Risks with making bosses feel distinct from regular enemies
F10	Environmental assumptions: Adds Environmental assumptions concerning how the game acts when it is and is not the currently in-use program, to keep in mind when developing further
F11	Risks: Clarified wording to clear up misunderstandings within the team
F12	Risks: Clarified why the player not collecting currency was a bad thing to make Risks clearer as power-ups and their costs are being experimented with
UI1	Environmental assumptions: Made it clear that this assumption applies for all methods of control

Changes to Method Selection and Planning

Within our Assessment 1 Method Selection and Planning deliverable we stated that we planned to share the leadership role as and when the project evolves. It became clear at the end of Assessment 1 that this was neither efficient nor suitable for our team. Issues which arose because of this included underperformance by team members (mainly due to miscommunications), the missing of deadlines and poor communication between members leading to tension between the group during a critical part of the assessment. Some team members clearly needed a leader from whom they could receive direction from and someone who will hold them accountable for missing deadlines or falling short of expected efforts. We still maintained 'experts' for each section who make executive decisions on their allocated work-package, but a group leader has been introduced to mediate problems with underperforming members, organisation of meetings or disagreements between group members. This change to our method was extremely vital for this assessment period due to the heavy workload caused by the introduction of the implementation process and the need to continue with progress over the Christmas Holiday break. Having a leader enabled the team to avoid falling behind when working remotely.

A change to the assignation of roles was also made to reduce miscommunication between members who shared the same role. It became clear during Assessment 1 that team members were completing the same tasks, but independently, leading to the unnecessary replication of work. To combat this, the team decided to create sub-roles within roles to more concretely distribute responsibilities. Roles such as programmer and tester were simply too broad and finer details needed to be included. Problems, such as the one stated above, were exacerbated by poor communication using the communication tools stated in our first Method section. Some team members were not vocal using these tools and felt more comfortable meeting face-to-face. Due to this, a further adaption of our agile method was made to meet twice a week, however, each sprint remained one week long. The extra meeting enabled members to voice issues arising between cycles and therefore reducing time lost due to miscommunications and confusions on tasks. During periods, such as the holiday breaks the team has agreed to use audio communication tools such as google hangouts as discussed in the first assessment to ensure that changes to roles and assignation of tasks are properly understood and welcomed by all members.

An additional tool was added for the second assessment as it became clear that some team members were not yet comfortable using the command window for Git commands and struggled with the concepts of version control. Git Kraken was introduced as it helped struggling members visualise the merging of sperate branches and more easily showed the committing of changed files. This addition was justified as it has enabled less technical team members to provide additional and useful input for the implementation of our game. Previously these members may not have felt comfortable committing/merging prior to Git Kraken's involvement.

Small changes to the 'Detailed Plan for Assessment 2' was made to take into account our chosen testing approaches. Black-box testing will commence towards the end of the implementation process rather than consistently throughout. This was done to allow more full implementations of features to be finalised hopefully allowing the team to use these features more cohesively and sequentially when testing. Another change was made to show the introduction of coding standards for programmers. This was introduced soon after the start of implementation, as it became that our programmers had considerably different coding styles and the lack of commentary made collaboration difficult.

A detailed plan for Assessment 3 has also been included with minor changes to the Gant Chart made. Key findings from Assessment 2 has resulted in a change to our plan for Assessment 3. For example, the writing up of reports must occur concurrently with implementation and testing to prevent such heavy workloads at the end of the Assessment period. More meetings have also been scheduled to reduce miscommunication between team members and a reduction in sprint lengths (now one week long) has been factored in due to the narrow time window for Assessment 3. This will allow the team to have a greater turnover of work and hopefully improve the effectiveness of conflict resolutions.

The development tool section has been edited to be more concise as suggested by our Assessment 1 feedback.

Link to updated Gantt-Chart with Critical Path:
https://teamcraigzombie.github.io/assets/downloads/UpdatedGanttChart-CriticalPath.pdf

Changes to Risk Assessment

For Assessment 2 very minimal changes were made to the risk assessment deliverable as we believe that our structure and method for managing risks was both efficient and worked well with our agile methodology. For example, our use of ownership to allocate responsibility for managing risks worked well in Assessment 1, as transpiring risks were handled appropriately and brought to the attention of the group by the owner of the risk in a timely manner. Due to this, many risks which transpired had less severe impacts on the assessment than expected. The simplistic layout of our risk assessment table with risk IDs and categories helped with the identification of risks and the succinct nature of our mitigation plans meant plans where easily understood and executed well by all team members.

Our mitigation plans also proved vital in reducing the severity of the impact of transpired risks and many worked well throughout the first assessment. Despite this, the team felt that clarification on the use of mitigation plans was needed in the risk assessment deliverable. Under the Mitigation heading we have explained how each mitigation plan should be used as a basic outline on how to alleviate the impacts of a risk. The mitigation plan should be adjusted according to the context of which the risk transpired, in order to minimise the severity of the impact. This was a frequent occurrence throughout Assessment 1. Risks which could be categorised under one of the risks in the risk table would transpire and the mitigation plan would be put into place. However, the context in which the risk transpired affected the effectiveness of the mitigation plan. For example, risk A4 states that issues will arise if team members fall ill close to the deadline. Our mitigation plan suggests a change to our project plan should be made and workloads for each member should be adjusted. This was often unnecessary as members who had fallen ill were still capable of completing their assigned work, but remotely, rather than travelling to work face-to-face. So, yes the mitigation plan suggestions helped to alleviate the impacts but slight adaptions were often needed. We believe this decision to adapt mitigation plans based on context is justified as it allows the effectiveness of mitigation plans to be maximised for each particular scenario.

A suggestion from our Assessment 1 feedback mentioned that we needed to include a method of updating the risk assessment table as the project progresses. As a result, we have included an additional section which describes our method of monitoring risks. This section gives examples of where updates to established risks are required, where additions to the risk table are needed and when risks can be removed from the table all together. As mentioned in the deliverable, our inclusion of owners considerably helped with ensuring all risks where being monitored and the necessary adjustment to the table made. This helps to justify our inclusion of an ownership column. Despite having owners of risks who are responsible for managing their allocated risks, we had not yet put in place a procedure for updating the table, as stated in our Assessment 1 feedback. A procedure was put in place where team members will attend a risk meeting every three weeks to discuss and update the risks assessment table. We agreed that three weeks was a suitable window between meetings, not only because it ensures that we can have at least two risks meetings per assessment, but also because it gives enough time for new risks to transpire and new observations made. Another justification for these meetings is that the table will be uploaded with all team member's present. This ensures that all members are aware of new changes to the table including discussed and agreed changes to mitigation plans. A final point is that the team have agreed to be flexible with the addition of extra risk meetings should severe and unanticipated risks transpire. This should hopefully reduce the time unanticipated risks impact our project.

Links to Updated Assessment 1 Deliverables

Updated Method: https://teamcraigzombie.github.io/assets/downloads/UpdatedMethod.pdf

Updated Risks: https://teamcraigzombie.github.io/assets/downloads/UpdatedRisk.pdf

Updated Requirements: https://teamcraigzombie.github.io/assets/downloads/UpdatedReq.pdf

References

[1] I. Sommerville. *Software Engineering*. Pearson, tenth ed., 2016

[2] IEEE, 830-1998. *IEEE Recommended Practice for Software Requirements Specifications*. Institute of Electrical and Electronics Engineers, 1998.