Centre: [redacted]

Candidate: [redacted]

H446 – “Vampire hunter”

Additional Commentary to Examiner

- The pen drive contains some additional evidence in the form of a video. Please take a look at this first so you can get a feel for the program.

- **Analysis:** Analysis starts with an outline of problem and then identifies the stakeholders. Here a named user from the target audience is given. How the problem can be solved by computation methods starts on p5, this is done well by taking each of the computational thinking strands in turn. Detailed research begins on p7. This takes the form of an interview which is first planned, then the transcript (p8) and then analysed (p9). Research continues on p9 by looking at existing similar games. This is support well with annotated screenshots. Features identified are then discussed on p16. Following this they have gone back for a more detailed second interview with their user, again this has been planned (p17), transcript (p18) and analysed (p20). The main features of the proposed solution are then detailed on p21. Hardware and software requirements begin on p23, these are sensible and justified. Detailed success criteria begin on p25. There are specific and measurable, each is justified along with a reference to show where the requirement came from in the preceding analysis. On p23 is an excellent table which looks at all the possible limitations of their proposed system, the limitations are described, then explained and then justified! These cover issues of time, hardware and software **9/10**

- **Design:** The candidate has presented a top down modular design on p30, this is accompanied by a good description of why he has done this and is followed by a break down of the diagram given more high level detail into what each area will include. Starting on p35 is a section on usability features support with annotated screenshots. An OO class diagram is present on p40, this is excellent and makes sense for their program. A very detail section on algorithms, all in the form of flow diagrams is presented on p40 onwards, and several of the more complex ones have been dry run tested using trace tables (see p40, p50, p56). A very good attempt to show how all the algorithms are intended to fit together to form a complete solution to their problem is found on p 59. A detailed table of key variables and data structures begins on p60, and each item has an explanation and justification. Test data to be used during development starts on p66 and this is followed by detailed acceptance testing they plan to use post development (p70 onwards). Valid, invalid and boundary input data is provided where appropriate. Several references to the candidates awareness of the importance of appropriate data validation can be found on p30, p31, p42, p60 and p66). **15/15**

- **Iterative development of a coded solution:** Iterative development begins on p79 and takes the form of telling the development story. There is clear evidence of successive prototypes being developed and the user is clearly involved at regular points for the purpose of review p96, 110, 124. There are excellent explanations and justifications backing up all decisions made as the solution is developed and these can be well seen at various points. Code is seen in decent code snippets throughout the development p83, 84, 86, 88, 90, 99, 106, 109, 115 etc . The code has also been provided as an appendix at the end. The code is well annotated both in the classes at the top and these annotations are meaningful and help to improve maintainability. It has been developed in an OO and modular way. All of the variables and classes are sensibly named, which all aids in the maintainability of the solution. Good evidence of validation can be found on p34, 35, 63, 69. However there isn’t really much discussion on why validation is being used or if it is appropriate to this solution, for that reasons I have dropped the top mark here. **14/15**

- **Testing to inform development:** There is excellent evidence of both implicit and explicit testing throughout the development process. Implicit testing can be seen whenever they hit a bug in developed which are all well recorded, along with their subsequent investigations and actions taken p90, 101, 107, 148. Explicit testing is excellent, test data identified in design is used and results recorded and justified and actions taken p89, 92, 93, 100, 101, 102, 107, 108, 109, 113, 115, 116 etc. **10/10**

- **Testing to inform evaluation:** Detailed end user / systems testing starts on p159. Detailed tests are seen for every section along with referenced video evidence on the USB pen. Testing is robust and doesn’t just test valid situations, there are plenty of borderline and invalid tests to check if the system would fail (Examples on p159, 160, 161, 163, 164

etc). Usability testing occurs on p166-170 in the form of an interview and also many usability features are tested in the previous test tables. **5/5**

- **Evaluation of solution:** The main bulk of the evaluation of the solutions tarts on p171 with evidence of how well the solution matched the original success criteria. Each requirement from the start is taken in turn and a justification for it given, the candidate then discusses if each has been met and if not why not. This is detailed and goes on for 5 pages! Starting on p177 the candidate looks at maintenance and limitations. Each feature which was not implemented is taken and a justification of how it could be included in the future provided. Every comment is well throughout our, criterial and analytical and at all times comments made are fully justified. **15/15**

68/70