**Introduction**

This is your project booklet. You will complete this project booklet throughout the course of 6 weeks.

You will evidence your planning, your implementation, and your testing in this booklet. You will complete your extended writing Evaluation in this booklet. **It is therefore important that you have access to this booklet during your lessons. SAVE A COPY NOW.** You will call this booklet YourFirstName\_YourLastName\_Year8\_Project. Make sure you **do not save this file in your downloads file**, it needs to be saved in your user area.

You will notice that there are two colours of text through the project booklet. Blue text (like this text) should not be submitted, but contains helpful advice and guidance. You should keep a record of the blue text as long as you find it helpful, and delete it before submission. Black text (like the text in the table below) NEEDS to be completed and submitted. **DO NOT DELETE BLUE TEXT AUTOMATICALLY; YOU WILL NEED IT TO ASSIST YOUR WORK IN THE PROJECT.**

A model project booklet will contain only black text, with lots of evidence from your planning, your implementation, your testing and your extended writing. Multiple screenshots of your progress from each of your lessons are mandatory.

**Planning**

*Your project is a substantial task to be completed over several lessons. The aim of this planning stage is to encourage you to think through your project, and manage your time effectively.*

*This planning document will be filled in by you over the course of the first lesson after you have chosen your project. You cannot change project once you have filled in this planning document.*

*This is worth 20% of the total marks available in the project.*

|  |  |
| --- | --- |
| **Initial Plan**  **Without doing other research** what are your initial thoughts on how to tackle the problem?  Laying out your ideas in a mind map can be helpful here.  What are the key parts that will make your project successful? Identify 3 key areas of success.  What do you need to find out/research to make your project successful? Identify 3 areas to research. | Turn the snake by replacing segments with rotated versions.  To detect where a segment has to turn, we create an object called turning point which has attributes called location and direction. Once a segment reaches the location, it turns in the direction it needs to face.  Create 2 different levels by opening text files which contain the locations of all the blocks. The text files will contain lists of 0 and 1 as such:  111111  000000  111000 Where a 1 is a block and a 0 represents nothing.  000000 We shall divide the grid up into blocks of 30 px  001111 so every time we blit we just multiply 30 by the  position the block is in. We can compute this using 2 variables which we add to when we check each row and column (in a **for** loop).  In order to make the food disappear once we have eaten it, we will generate a random location which changes once we have eaten it. This MUST only run once we have eaten the food. Once we eat the food, we will create a new variable called is\_eaten that will be made true. It is detected to be true in the while loop and if so, the location changes, and is\_eaten is reset to false. |
| **Coding plan**  What code do you think you will need? Identify at least 5 functions you will require and their purposes.  Where will you find help for this code if you need it? Identify 3 useful sources of information.  *Your teacher does not count!* | Using module **pygame**, as well as my Utilities module from an earlier project that I am working on at home.  Classes: main, snake (handler for entire snake), segment (object for each part of the snake), board (for adding obstacles in levels), score (score/time displayer, average time between collecting, etc.)  Functions:  **def segment.\_\_move\_\_(self, direction**) -> shifts a segment 30 pixels in a specified direction.  **def snake.\_\_move\_\_(self, direction)** -> Moves the entire snake forward by 30n pixels.  **def segment.\_\_turn\_\_(self, direction)** -> switch the segmentfor the rotated version.  **def snake.\_\_display\_\_(self)** -> render the snake on the board. What we shall do is to analyse the properties of every segment and render accordingly.  **def segment.\_\_edge\_\_(self)** -> detect if the segment has hit the edge of the board.  Websites that are useful:  https://pygame.org - > for finding out how to implement all the features.  <https://docs.python.org/3/library/enum.html> -> on how to construct classes with enum that allows me to make custom variable names. (ie: in c++ this would be implemented using #define THING 1,2,3 etc.) This will be very useful in making the code readable.  Stackexchange -> best website for finding answers to SPECIFIC problems that other programmers have had. This means you can look up different problems you’ve had if you can’t solve them by simple debugging. |
| **Existing solutions**  How have other people/companies solved a similar problem?  What do you like about their solution?  How could it be improved?  How has your design been influenced as a result of their work?  Find at least 3 other solutions to similar challenges. A well-researched project will contain 5. | * One thing I did over the week (as is shown in the Implementation part) was to ask my dad about how he would construct turnings. He explained to me that he would use a class with attributes direction and position that when the snake reached, it would move in the direction specified, and showed me an example. This is technically a way of solving the problem that someone else used, although it was given personally. * Another thing that I researched was how to implement a highscoring system. I found this page: which was very informative, and helped me learn a new way of storing data.   Image result for snake game   * You can see in this image how the snake includes turned versions of segments when turning, different segments for head and body (although I am also adding segments for tail as well.)   In order to stop the game if the snake’s segments collided/the snake bumped into the wall I checked an example game which used a similar mechanic. (It can be downloaded here: [www.raywenderlich.com/24252/beginning-game-programming-for-teens-with-python](http://www.raywenderlich.com/24252/beginning-game-programming-for-teens-with-python)) |
| **Additional Resources**  What additional websites/guidance have you used during your planning phase?  *Failure to correctly identify your sources of information can result in a mark of 0 being awarded for your entire project due to plagiarism.* | pygame.org/docs -> used in working out how certain classes and objects that are built into pygame work.  Tried just searching for “snake game” on Images, to see how the boards were laid out and how the games could look.  As is mentioned several times in this booklet, I asked my dad for some help on turning the snake. He wrote a very basic example code and explained to me how he would create different “turning points” that when reached, would rotate each segment.  I did not have to teach myself Pygame for this – I already knew it, but here are some of the websites that I originally used to learn Pygame:  [www.raywenderlich.com/24252/beginning-game-programming-for-teens-with-python](http://www.raywenderlich.com/24252/beginning-game-programming-for-teens-with-python)    [www.inventwithpython.com](http://www.inventwithpython.com) – the second part of the first online book on this page ([Invent Your Own Computer Games with Python](http://inventwithpython.com/inventwithpython_3rd.pdf)) is about pygame. |
| **Improved plan**  Based on your research and information how has your plan improved? Discuss 3 ways in which you have adapted your initial ideas based on your research.  Re-draw your mind map with more detail. *You will be turning this mind map into a step-by-step plan.* |  |
| **Coding breakdown**  Code that is written well is often modular. Research what is meant by this.  Take part of the code you have identified in your mind map. Identify how this code could be designed in a modular way. *You will need to break down all of your coding into these modular tasks.* |  |
| **Peer discussion**  Find a partner who is also in the peer discussion phase.  Ask them to read your improved plan. They should provide up to 3 pieces of constructive feedback on how to improve your plan.  Record this here, and (if relevant) your ideas on how you can incorporate this feedback into your plan. |  |

**Implementation**

*Your project is a substantial task to be completed over several lessons. The aim of the implementation stage is to record your progress in each lesson.*

*The implementation stage will be filled in by you over the project weeks. You must evidence some work in here from each lesson, even if your work is debugging or research.*

*This is worth 20% of the total marks available in the project.*

|  |  |
| --- | --- |
| Lesson 1 | During this lesson I did not really have a very clear idea of how I was going to implement it. I decided which project I was going to do, what I was going to use to do it, and wrote a little bit of the startup code – such as the game loop.  I had success in starting up the code – a pygame loop is something that I have written many times, so it was not that difficult to do it quickly from memory.  What I could’ve done much better, I feel, is to start thinking a bit more about what I needed to add – not just planning-wise; maybe even adding some of the commands and classes as templates first, and then working on them.  For the next lesson I hope to get more done in terms of planning. |
| Lesson 2 | During this lesson, I worked on planning a bit. I had also worked on my code a little at home. I managed to get a slightly clearer structure of how I was going to solve the problem, but I was having trouble thinking of how to turn the snake. This was a very big part of the project. Since I have experience in many of the other stuff, such as representing things with datastructures, moving things around etc. which I thus did not need to focus on massively during my plan, a large part of my implementation is missing.  What I hope to get done next time is to have worked out how to create turnings in the snake.  (I actually ended up doing this at home after talking to my dad a bit about how it could be implemented. He created a basic template for a couple of the classes that could be involved which we then looked at together, went over and added things to.) |
| Lesson 3 | Over the week I completely restarted my plan (this plan, in fact) as I felt my original planning had not been good enough and focused enough. I worked on my code for moving the snake and created the class for the turning point. |
| Lesson 4 | During this lesson I worked on a little of everything. I tidied up some code, added some new code for filling the board, score variables etc. that I forgot in the last lesson. I also jotted down in this plan a few more ideas for how stuff might be done, such as making the food vanish once eaten.  What went well: I managed to implement quite a few of the small essentials, which saved me having to work out all of them and then putting them in later.  What could be improved: When running the game at home, I discovered that actually, there were many things I had forgotten to implement, such as remembering to start all the different segments in different positions. This |
| Lesson 5 |  |
| Lesson 6 |  |
| Lesson 7 |  |
| Lesson 8 |  |
| Lesson 9 |  |
| Lesson 10 |  |
| Lesson 11 |  |
| Lesson 12 |  |

**Testing**

*Your project is a substantial task to be completed over several lessons. The aim of the testing stage is to show your methodical approach to coding and your problem solving skills.*

*You will start to complete the testing stage when you have your first module of code successfully written. Your testing stage will involve playing your game until you discover an error or a logical fault with your game, repeating it, then fixing the error in your code and demonstrating that the error no longer exists.*

*Testing is not the same as debugging. Finding a typographic error in your code (spelling mistake) is not the same. An example of an error is your code crashing because someone clicked 3 buttons when you wanted them to click 4.*

*A successfully tested game will have discovered at least 5 errors and implemented solutions for them.*

*This is worth 20% of the total marks available in the project.*

|  |  |  |
| --- | --- | --- |
| **Error description** | **Proposed solution** | **Evidence of success** |
| **After I thought I had implemented the turning function, the snake did not register turns. (not quite sure if this is testing or debugging)** | **Go over code, think, exactly what is happening with the code I have written, check if I have missed anything. Use the debugger to see if the turning point gets created properly or not.** | **The turning point was created in the WHILE loop, so it was reset every time. Another error was that the points were all set to go in the same direction and some methods that were meant to contain single “=” contained “==”.** |
| **If you turn left or right and then turn downwards, nothing happens. (two movements in a row didn’t work.)** | **Read code carefully. Use debugger and step through. (In truth, I believe there can be NO proposed solution for MOST bugs in code. There is only one way – reading code carefully, debugging over and over, stepping through and watching object statuses.)** | **The turn detector needed to detect every single version of a segment, including turned versions, otherwise certain types would not register.** |
| **The turning piece was inserted one move too late, and then the turning piece was often facing the wrong way.** | **Look at the exact timing of the turns. Use the direction of the next segment in the snake when inserting the turning point, so we know where to put it so it is exactly between the two of them.** | **In the end, my solution was to create flipped versions of turning points, and use the methods described to set it correctly.**  **It turned out they WERE actually being inserted right (I slowed the game down) and the** |
| **If you turn whilst eating the food, the segment direction goes wrong and the snake’s tail/body often goes in the wrong direction and the snake loses parts.** |  | **The reason for this was because we had forgotten to add a case for turning the new types of segments such as the flipped versions of pivot points** |
| **Some segments of the snake that were not the head somehow managed to go off the board and this made many of the segments keep going upwards and vanish, so there was just a disembodied head and tail.** | **Set it so the snake is killed if any part of it goes off, and not just the head.** |  |
|  |  |  |
|  |  |  |

**Evaluation**

*Your project is a substantial task to be completed over several lessons. The aim of the evaluation stage is to encourage you to reflect on the progress you have made over your project and what skills you have developed.*

*You will evaluate your progress during the project as well as at the end; however most of the required evaluation is completed at the end of the project when your coding is completed.*

*Your evaluation is a piece of extended writing, no more than 2 sides of A4, where you critically assess the progress you have made as a computer scientist during your project as well as show your ability to clearly communicate the skills that you have learnt.*

*This is worth 20% of the total marks available in the project.*

**Your evaluation must contain:-**

**An introduction to your project where you explain what the objective was.**

**A detailed discussion of what went well for the whole project, including examples of successes.**

**A detailed discussion of what could have been improved for the whole project, including examples where you suggest improvements.**

**Evidence of the skills you have developed through the project.**

**Evidence of learning and progress through the project, e.g. learning a new coding language.**

**Suggestions of improvements to the project, at each of the stages.**

**(no more than a paragraph) A discussion of what grade you feel your work deserves with evidence of why you believe this.**