**Record of Progress**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Action** | **Dur.** | **Reflective Commentary** |
| **25/08/16** | Project was announced, teacher went through the candidate instructions with us | 0h40 | Started thinking about what I might want to build, came to the conclusion that a 2D game would easily fulfill the project requirements and would also be something I am interested in creating. |
| **29/08/16** | Thinking about different 2D games that I could make | 0h30 | Have thought about a few games that I could attempt to make, all of them being clones. I feel like a clone would be a good route to go down as I will probably already be very familiar with the mechanics of it and how it will work/how it should feel. A few ideas going through my head, all being classics, are Snake, Pong and Minesweeper. Have also maybe thought of doing a clone of the more modern game ‘2048’. Will continue to think. |
| **06/09/16** | Started project proposal | 0h30 | Decided on creating a game inspired by the app ‘2048’, happy that it is feasible within the time frame I have and that it will fulfill all the necessary requirements. |
| **07/09/16** | Created outline project plan | 1h00 | Happy with the timings I have given each aspect of the project. Although I may go in an out of these timings I do feel it’ll help me be completely ready come submission time. |
| **14/09/16** | Completed project proposal | 0h30 | Whilst keeping to the basic premise of the the original ‘2048’, I decided to extend the progression of the game by adding multiple levels. If I’m being pushed for time later on I may have to cut this idea though. |
| **21/09/16** | Completed feasibility study | 1h00 | I feel all aspects of my project seem completely feasible however I know it’s unlikely by the time submission comes around for there not to have been anything that has gone wrong at all. |
| **22/09/16** | Created user survey and distributed it to my fellow students in class | 0h30 | Tried hard to think of questions that would help me make decisions, mainly involving the design and feel of the game. I feel the user interface of a simple game is extremely important. |
| **28/09/16** | Completed analysis of findings | 0h40 | Felt that the first three questions helped me decide key aspects of the interface and design however the last two were too ambiguous and vague to give me very helpful responses. Especially question 4 in which I did not specify that I meant time ‘per session’. |
| **06/10/16** | Teacher issued updated project specification and I combined my project files into one document | 1h00 | Decided it was more efficient and uniform to hold all documents in my project in one file. Feels nicer having everything all in one place |
| **13/10/16** | Created detailed project plan in the form of a Gantt chart | 1h30 | The visual representation of my schedule has shown me I need to get the requirements spec and test plan done quickly so I can get onto actually building my game. Aware that it could take a while to get everything working nicely. |
| **15/10/16** | Started requirements specification | 0h40 | I feel as though there will probably be a few changes to my scopes and boundaries down the line, some scope will probably become a boundary and although unlikely if I have enough time after implementing all the scope I need I may potentially try and add some boundaries for fun, sound for example sounds interesting, although unnecessary. |
| **16/10/16** | Completed requirements specification | 0h20 | Realised that a downside to writing my solution in python may be ease of access on getting it onto other machines, will need to look into this as thinking a good approach for testing would be to give my game to a lower year to try. So will possibly need to look at ways of converting .py files. into windows .exe. I’m sure there’s a way somehow. |
| **25/10/16** | Created preliminary test plan | 0h60 | Will need to revisit this at a later date when I think of more things that my game should be able to do. |
| **30/10/16** | Added to resources in the project plan | 0h20 | Added youtube and stack overflow to resources list as well as specifics of the software that I will be using |
| **02/11/16** | Started sketching my interface design | 0h20 | Already had a pretty good picture of what most of my interface would look like but it was good to finally get it on paper and think about the spacing of elements on the screen. |
| **05/11/16** | Completed sketch of my interface design | 0h20 | Happy with the design and flow of my interface, feel it keeps with my findings from my user surveys. Kept the design as clean and simplistic as I could, leaving the options behind the game screen is one example of this. |
| **07/11/16** | Amended and added to my test table (functional requirements) | 0h60 | Now that I have a better idea of what elements I will need to test in my user interface I could add those elements to my test table. Having started my interface design later than expected and still not having finished it I am now about a week behind schedule, this will eat into the time I have for actual implementation of my solution but I feel I made my outline and detailed plan with enough contingency time that this should not be a critical problem. |
| **09/11/16** | Amended and added to my test table (non functional requirements) | 0h30 | Will need to come up with a way of testing response time between key presses and the screen actually being updated. Already know that python has a clock module so I’m sure this will be possible to achieve after some research is done. |
| **12/11/16** | Started wireframing my interface design | 0h45 | Took me a while to find a suitable wireframing tool that I liked but finally decided to use ‘moqups.com’. Decided to rename the ‘Hi-Scores’ title on my leaderboard screen to ‘Leaderboard’. Also decided to change the ‘element’ tile that was going to appear in the header between the user’s name and score to a ‘level’ counter label that displays the level the user is currently on. Felt this information would be more useful to the user, especially if they leave their game for a while and then come back to it. The highest element will always be visible on the board anyway. |
| **14/11/16** | Completed wireframing my interface design | 0h30 | Added interactivity to my wireframe and realised that my navigation was not complete. The options menu screen did not have any way off of it. Decided to add a back button so that it was possible to return to the previous screen. This really showed me the importance of planning before actual implementation as if I have probably saved myself a good bit of time finding this now rather than further down the line when I am actually writing the code. |
| **16/11/16** | Annotated interface design wireframes | 0h40 | Turns out I needed the premium version of ‘moqups.com’ to have access to download the PNG image format of my wireframes so I just had to screenshot from the browser and crop the form out as best I could. |
| **17/11/16** | Small amendments to my project proposal, feasibility study and scope | 0h30 | Had previously decided that the username could only be 3 letters long, when creating my name input screen wireframes I decided there was actually no need for this cap therefore I will now accept a range between 3 and 8 characters. Added screen size, resolution and bit depth to my list of minimum requirements as I feel they are necessary to mention, if I think of any other minimum requirements in the future I will come back and amend this again. Also added to the schedule feasibility as I felt it was a bit too short. Amended the scope so that it now includes all game screens, in doing this however I have realised that I have not considered one of the screens I will need: the instructions screen (which also affects the main menu screen as an extra button will need to be added). Will work on this tomorrow. |
| **18/11/16** | Created and annotated wireframes for the ‘Instructions Screen’ | 0h30 | Feel like I have done sufficient UI design now to move onto my program design. I am just over 2 weeks behind schedule at this point but I feel this is manageable given my optimistic deadlines. |
| **25/11/16** | Sketched out class diagrams for the block and gameboard classes | 0h30 | Think I have a good idea now of what methods and attributes and classes both of my classes will need. It’s turned out that the block class will basically just be a node, containing only two bits of information, the element and the location of the block whereas the gameboard object will contain all of the mechanics that will then act on these block objects. |
| **30/11/16** | Started pseudocoding:   * Block constructor * Gameboard constructor | 0h20 | Just copied out what I had wrote in my class diagrams, didn’t think of any other methods or attributes that I will need. |
| **04/12/16** | Continued pseudocoding:   * Spawn method | 0h40 | Relatively simple method, made sure to comment as I went so that it doesn’t all pile up and leave me having to do it at the end. |
| **09/12/16** | Continued pseudocoding:   * MoveBlocksLeft method | 0h30 | This method has been harder to think about, leaving it for now and will come back with a fresh mind another day. Have the general idea in my head, need to iterate through the blocks and look through the empty cells until I find an edge or a block and then take appropriate action. Am sure when I come back to it I’ll be able to think of a solution. |
| **11/12/16** | Continued pseudocoding:   * MoveBlocksLeft method * MoveBlocksRight method * MoveBlocksUp method * MoveBlocksDown method * Combine method * DrawGameBoard method * DrawBlocks method | 1h10 | Confident I have worked out a way to do the movement mechanic now, copy and pasted the left movement method and made the few changes needed for each direction. Thought about how I will combine blocks, confident my method will work. Also started to think about how I will draw each screen to the display and pseudocoded the methods for drawing both the gameboard and the blocks within it. For the moment the colours of the blocks are just some random colours that I could think of rgb values for, will make these values more appealing down the line, not necessary whilst this deep in development. |
| **14/12/16** | Continued pseudocoding:   * DrawGameBoard method * DrawGameScreenHeader method * DrawMainMenuScreen method * DrawInputScreen method * DrawLeaderboardScreen method * DrawInstructionsScreen method * DrawOptionsScreen method | 2h00 | Wrote out all the drawing methods, relatively easy to think about, just referred to my user interface when deciding which elements I will need in each screen. These functions basically just set a list of the elements for their respective screen and return the properties of these elements so that they can be used by the ‘loop’ methods for event triggers such as mouse clicks. The ‘instructions’ screen is a bit too basic at the moment, will come back later and make a more comprehensive guide. |
| **18/12/16** | Continued pseudocoding:   * gameLoop method * nameInputLoop method * leaderboardLoop method * instructionsLoop method * optionsLoop | 1h00 | Thought of a way to switch between game screens that involves infinite loops withing infinite loops. Unsure this method will actually work when written out in code as I feel like python may have problem with so much function calls within function calls. |
| **20/12/16** | Wrote a short script to test out navigation method | 0h45 | Very simple script that just demonstrates that this method works. In this case just using string inputs through the console but eventually the navigation will obviously be triggered by events such as button clicks. Huge relief that this method works as it means I can move on knowing that that this aspect of the game will work!  This code can be found in the appendix. And a screenshot of it working can also be found in the appendix, screenshot number |
| **23/12/16** | Continued pseudocoding:   * saveGameState method * openGameSave method | 0h20 | Decided I will end up using some sort of binary file to store my game save objects. Will research and find a way to do this when I get to it as I have not done this in python before. |
| **25/12/16** | Wrote the block class:   * \_\_init\_\_() * \_\_str\_\_() | 0h20 | Felt good actually finally getting the implementation stage of the project, I decided to use the text editor ‘Sublime Text’ as I have used it in the past and it offers a great number of user created addons that can be really helpful. |
| **27/12/16** | Started writing the gameboard class:   * \_\_init\_\_() * \_\_str\_\_() * spawn() | 0h40 | Wrote a ‘special’ \_\_str\_\_ method that prints out the state to the console so that I can see how I am affecting it. So that I don’t have to run the program from a terminal every time I want to test it I installed a sublime text ad don called ‘REPL’ that opens a new tab and shows what would be printed to the console whenever you build the .py file. This will really help move the project along faster as I can quickly test what affect changes to my code have now, definitely glad I picked sublime text as It’s very fast and fluid to write in. |
| **30/11/16** | Continued writing gameboard class, attempted to add the movement mechanic :   * blocksLeft() | 1h20 | I have begun attempting to write the movement mechanic, today I have focused on just writing out what I had thought up in my pseudocode. Just stuck to testing one direction of movement until it works so that when it works I can just copy/paste and make the few changes that I will need to make each respective direction work as intended. At the moment I am running into a problem involving multiple layers not moving properly. |
| **02/01/17** | Continued writing gameboard class, managed to get the movement mechanic working:   * blocksLeft() * blocksRight() * blocksUp() * blocksDown() * leftSort() * rightSort() * upSort() * downSort() | 3h00 | Finally got the movement mechanic properly working, the tiles were being incorrectly blocked by each other due to them not being moved in the right order, this was fixed by defining 4 separate sort functions (one for each direction) so that the tiles were moved from the side that they were being moved to first. This fixed the problem.  There was another issue with this mechanic, this one being a bit more strange. When tiles were being combined, they were leaving ‘shadows’ to the side of the newly formed tile. I realised that this was because because I was not updating the game state correctly. The fix to this was to introduce a flag variable that flagged the fact that a combination had occurred. This fixed the problems. A before and after screenshot of one of the movement functions can be found in the appendix, screenshots numbered 2.0 and 2.1. And a closer look at the actual ‘combinedBlock’ logic that fixed the bug can be seen on screenshot 2.2.  Also had an issue where the block loop would skip over the block in the list after a block that had been combined. Resolved this by changing the looping structure. |
| **03/01/17** | Continued writing gameboard class:   * combine() | 0h30 | An issue arised where when blocks were combined, a ‘shadow’ of the moving block would be left on the gameboard. First tried to resolve this by giving each block an ID but ended up just needing a boolean variable to flag if blocks have been combined or not so a ‘shadow’ isn’t added to the game state if blocks have been combined. |
| **05/01/17** | Added ‘draw’ methods for the game and main menu screen to my ‘main’ file | 1h00 | Slightly disappointed that pygame doesn’t quite have inbuilt functions to create buttons, labels and input boxes. Looked up how to create these and then made my own functions to do these tasks so to prevent a lot of repetition. |
| **07/01/17** | Added the ‘loop’ methods for the game and main menu screen to my ‘main’ file | 1h30 | Turns out my method of switching screens works as intended, extremely happy about this as it means I will not have to go back to the drawing board in regards to this mechanic. |
| **10/01/17** | Added the save and load functions to my ‘main’ file | 0h45 | Researched how to do binary file handling in python and discovered pickle, this makes the process of saving a game state under a name and then opening it back up quite easy. Implemented the save function so that it runs every time the player makes a move, an ‘autosave’ feature. Therefore the ‘save’ button in my current design for the ‘options’ screen is now redundant, will refine my interface design next session. |
| **13/01/17** | Refined my interface design for the options screen | 0h30 | Removed the ‘save’ button from my ‘options’ screen design as I now have an ‘autosave’ feature as explained above. Replace the ‘save’ button with a ‘leaderboard’ button, which will allow players to check the leaderboard mid game without having to return to the main menu. |
| **15/01/17** | Added the ‘draw’ method for the ‘name input’ screen, consequently added the ‘drawLabel’ and ‘drawButton’ functions | 0h45 | The process of creating labels (text on the screen) and buttons was a relatively long process so I decided to generalise each process and create a ‘drawLabel’ and ‘drawButton’ function to avoid major code duplication throughout the project. |
| **16/01/17** | Added the ‘loop’ method for the ‘name input’ screen and the ‘drawInputBox’ procedure | 1h30 | Used the same loop for both the ‘new game’ and ‘load game’ screens as the only thing different between the two is the text that appears, this can easily checked upon entering the loop. The code for the input box took a while to get working but now it fully functions, only thing I might want to come back and add is some kind of indicator that you have clicked inside the input box. As at the moment it isn’t very clear and might leave some users confused. Could maybe work on filling the box grey when the user clicks on it so it is clear that they are then free to type. This isn’t essential however so I will leave it for now. |
| **18/01/17** | Added the ‘draw’ method for the leaderboard screen | 0h45 | Was going to take me a while to manually put in the coordinates for every line on the table so instead I crated a for loop that incremented the y coordinate and draws all the lines I need. Decided to show just the top 10 scores. |
| **20/01/17** | Added the ‘loop’ method for the leaderboard screen | 1h30 | Back button initially didn’t work as expected, was sending user back to the game screen they had just left. Rectified this by adding a flag that tells the loop if the user is coming from the game screen. If they are then it doesn’t just ‘return’ the function, it calls the ‘mainMenu’ loop instead. |
| **21/01/17** | Removed the white gaps between blocks | 0h45 | This was just an aesthetic fix, however I do think it was worthwhile as it makes the game more appealing to look at, therefore making it more engaging. You can see the white lines I am referring to in screenshot 1.2 in the appendix. The fix involved creating a new dictionary that matches each cell to a tuple value containing the width and height required for a tile occupying that cell, this differs between different cells due to the thickness of the grid lines that each cell touches. The fixed blocks can be seen in screenshot 1.3. The before and after of the fix itself can be seen in screenshots 3.0 and 3.1 respectively. |
| **27/01/17** | Added levels to the game | 1h00 | To test this mechanic out, I changed the code so that it would move onto the next level when you reach block 4. This is just temporary whilst I fully test the feature out. There’s a slight issue where if the user continues pressing keys whilst the ‘level complete’ text comes up then the moves are stored in the event queue and quickly execute after the new board appears. This might confuse some users however I have tried to resolve this by clearing the event queue. But to no avail, so might just have to let this remain. |
| **29/01/17** | Created colours.py | 1h00 | Took the colours from the atomas wikia, have commented the link in on this file, really like the colours used in this game and the colour codes are freely available so I see this as the best solution to the problem of finding a good colour scheme for the blocks. |
| **05/02/17** | * Separated drawing functions into a new file: ‘drawing.py’ * Separated game loop and related functions into a new file: ‘loops.py’ * Separated basic setup variables and other constant variables and structures into a new file: ‘setup.py’ | 1h30 | Took a while to get everything working. With the files now separated, lots of functions have to be imported between the files. It now works though, and as a result, my solution looks and feels a lot cleaner which I am pleased about even it doesn’t actually help the program functionally |
| **07/02/17** | Edited test plan | 1h00 | Created the final version of the test plan (1.3), ready to start actually formally testing the game now, feel like I’m getting closer and closer to the finish line, still got ways to go though |
| **09/02/17** | Carried out final testing | 2h00 | Majority of testing went without a single unexpected result until I found that entering a null string into the name input box triggered a typeError. Easy fix, screenshot is numbered 4.0. |
| **11/02/17** | Refactored ‘block’ to ‘tile’ | 0h30 | Felt it was bad having this inconsistency between the code and the documentation. Especially as it involved one of the key components of the game. |
| **15/02/17** | Collected final end-user testing from class mates and friends | 1h30 | Both wrote down comments that the testers mentioned while testing and collected answers from the questionnaire I made in my test plan. Happy with the varying feedback I got, plenty to talk about in the evaluation. |
| **21/02/17** | Started evaluation, completed the ‘development process’ section | 1h00 | Referred to requirement spec extensively for this part, basically just listed all my scopes and boundaries saying what I had(not) achieved. |
| **24/02/17** | Continued evaluation, completed the ‘solution’ section | 1h20 | Compared my end result to my requirements spec. Fortunately, nearly all of it matches meaning that I have done a good job sticking to the original plan, very pleased about this |
| **27/02/17** | Deadline changed | 0h20 | Edited project plan to show this (1.1) Really thankful for this extension as I believe it will give me just enough time to fully complete this thing. |
| **01/03/17** | Continued evaluation, completed the ‘performance’ section | 1h00 | Decided to focus this part of the evaluation on my time management and my implementation. Feel that I gave myself a fair review. Is good to reflect on myself and how I have done |
| **05/03/17** | Started compiling appendix, added old versions | 1h00 | Put all my old versions into this document, bit fiddly with formatting and such but got there in the end. |
| **09/03/17** | Added all screenshots into appendix | 1h00 | Took a while but compiled all the screenshots I had taken for final testing (and others that I had taken during development that may be referred to in other areas) into an appendix. |
| **10/03/17** | Cross referenced screenshots in final testing | 0h40 | Added references in this record of progress to some screenshots of various bug fixes. Feel this will give a better understanding of the bugs. |
| **11/03/17** | Converted .py ‘main’ file to .exe so it can run on windows | 0h40 |  |
| **13/03/17** | Final formatting | 0h50 | Feels good to package everything up and make it all look nice, can see all the hard work that I’ve put into this over the months. |
| **14/03/17** | Printed everything, sorted it into ring binder | 1h30 | Final comment: I’m not going to lie, it’s a massive relief to get this of my back, it’s been a good experience and has definitely taught me a lot however it did get stressful at times and it was beginning to become a bit of a burden. Anyway, all over now, happy with my final product. |