

APPENDIX A: SPRINT DOCUMENTATION TEMPLATE

1) Summary data	
Team number	5
Sprint technical lead(s)	Lin, Stuart, Duncan
Sprint start date	09/03/25
Sprint end date	29/03/25

2) Individual key contributions	
Team member	Key contribution(s)
Eric Shi	Testing, bug fixes
Lin Hein	UI developments
Owen Chen	Testing, bug fixes
Duncan Law	Testing, UI developments
Stuart Baker	Produced sprint document and notes

3) User stories / task cards
<p>We are hoping task 1 will help us identify any bugs and issues with the game, as well as give us an insight into how the game runs and the AIs ability. This will be the first large test we will run on the game without any human input, so the results will be very valuable. We are anticipating problems to arise.</p> <p>Task 2 is an extension of task 1, to get a better understanding of how the game will work, and if bugs that popped up in earlier runs were fixed, an anomaly, or still present. it will also allow us to test the game using different setups (hard mode, normal vs abridged, etc.).</p> <p>Task 3 will help us further identify any bugs and issues with the game, allowing us to ensure the user experience is smooth and accurate. It will allow us to locate any discrepancies in the games logic and see if what is displayed on screen is a true reflection of the current games' situation.</p> <p>For task 4, we will be running the whole game to make sure it works and functions correctly. This is touched upon in task 1 and 2, however, for we will use a variety of setups for this, having human vs human, human vs ai and ai vs ai to get a broad output of results. Any bugs and issues revealed will also be valuable and will help us to eliminate them.</p> <p>Task 5 is the postponed user story from the previous sprint as we didn't have enough time to complete it and it also wasn't a priority of that sprint. Our aim for this is to create our final UI version that we are happy with. We want to implement a few design features to make the board more understandable, providing the players with board information visually, instead of just when they land on a tile. We will introduce a few new images and features as well, such as a few game sounds.</p>

Our aim for task 6 is purely just to eliminate any bugs that we have identified. We are expecting multiple issues to arise from previous tasks, and this task will be dedicated to fixing them. We will have notes from the previous tasks outlining what the bugs are and possible solutions.

For task 7, we will be speaking to Mr Raffles and his representatives to gather their feedback and opinions on the game we have produced, and whether it meets their criteria. We will take their feedback and advice and try to apply any improvements/changes they have put forth.

Task 8s goal is to reduce redundant code and make the code base easier to read and maintain. We have been trying to write the code in a similar manner, but we wanted to first ensure we had met all instructions first before changing/removing unnecessary code. We will adjust accordingly as we test the code and fix bugs, never changing too much at once to prevent major issues.

Task 1: Run the game with multiple AI player agents playing against each other for an extended time period

Task 2: Repeat task 1 multiple times

Task 3: Create and execute unit tests

Task 4: Begin system testing

Task 5: Update UI to final design

Task 6: Fix any bugs identified

Task 7: User/client feedback

Task 8: Improve code maintainability and readability

4) Requirements analysis

Tasks 1 and 2 is based off a suggestion from one of Mr Raffles representatives, who recommended a good way to test the game, specifically the AI over multiple extended time periods would be by running a game of just AI player agents and letting them reach a conclusion naturally. Whilst running the game, we won't necessarily have to stay watching the game pan out for the full duration, but we will have the game log open which will allow us to see any game ending bugs or minor issues. We will still watch the game for a full run through or two just to see any errors on screen and to make sure it is functioning as we wish.

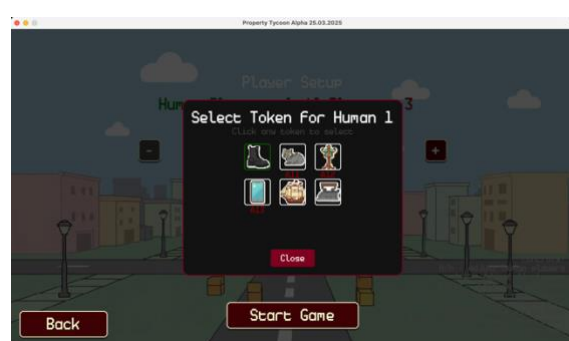
Tasks 3 and 4 are part of the deliverables in the coursework brief, so would be a mandatory, non-functional requirement. The purpose of including this is to check that our programming is all running as expected, updating variables when it's supposed to and displaying the correct information when needed. We will include multiple unit tests, hoping to cover the majority of our implementations, and the system testing will be run multiple times, using the results from task 1 and 2 and then expanding by running human vs human and human vs ai games. All aspects will be covered.

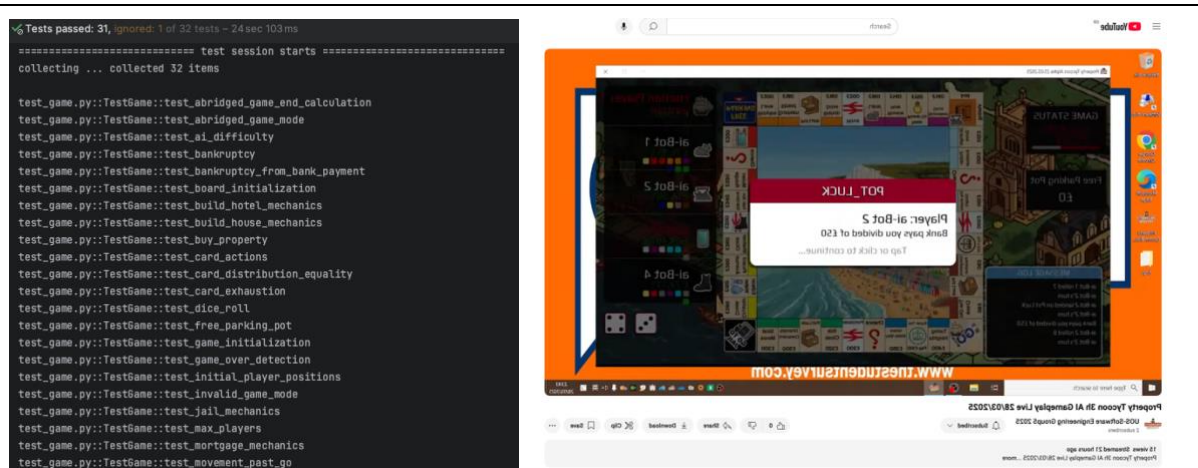
Task 5 addresses making the game ‘fun to play’ and have a ‘colourful and intuitive interface’ user requirement which we have focused on repeatedly in previous sprints. This should be the final major update to the design, excluding a few minor tweaks if we aren’t happy afterwards. Once this task is done, we will have finished the UI implementation for the game. Task 6 will hopefully fix all bugs we have discovered, and if not, it will hopefully complete the coursework deliverable that states – ‘It is acceptable for bugs and other issues to remain in the software as long as they are clearly identified, documented and that you have provided appropriate commentary to explain why they exist and what you would do about them’ – because we will have documentation explaining what bugs we weren’t able to fix/solve. Time permitting, we should fix most of the bugs.

Task 7 will enable us to adjust and tweak the game to Mr Raffles desire if he is not fully content with what we have produced. It should help us make the game even more ‘fun to play’ as it will tailor the game towards his vision he has. Mr Raffles and/or his representatives will be able to guide us in the correct direction and suggest changes that will help us enhance the final product.

Task 8 will be a continuous task throughout the entirety of the sprint, overlapping with other tasks and joining in with them, reducing redundant code as and when it is found and safe to do so. It will hopefully make the readability and maintainability easier for Mr Raffles and his team when they take over the operation after the project finishes. There is no specific user requirement for this, but we want to provide Watson Games with the best product possible, which this will help with.

5) Design





Six images have been included, four of them are highlighting the UI changes, and the final two are displaying our unit testing results and system testing respectively.

The first three images show our new UI for starting the game, having an updated and more arcade-style feel to it. The design is a lot cleaner than the previous version we had and fixes the issue where the text displayed was overlapping and difficult to read. The only tweak we may make is the text colour because sometimes it can be a little tricky to see against the background image as the colours are similar shades. However, this is a massive improvement to the previous UI we had and has been complimented by Mr Raffles and his representatives.

The fourth image is a screenshot taken whilst running the abridged game mode against three ai player agents. There are no new features on here compared to the last game version, however, each component has been enhanced and improved. The most noticeable difference, other than the background, is the board and board tiles. We have changed the boards background to a few local hotspots, the cliffs and Brighton pier, which provide the users with an enticing view. The board tiles have been improved drastically. Before, we only provided a few tiles with information, being the corner tiles, card tiles and station tiles. Now, we have laden all tiles with basic information, making the game more interactive for the user. Each tile states what they do, be it a card tile, corner tile or property tile, it will be shown. This makes it a lot easier for the players to know where they are and what the next best moves would be.

The fifth image is a screenshot when running all of our unit tests. Each test is executed and produces the expected results, bug-free.

The final image is the YouTube channel we created, specifically showing a video where we ran the game for three hours with just AI player agents playing. This nicely demonstrates the functioning game we have produced, as well as our system testing. The link to the channel is here: https://www.youtube.com/channel/UCfA_oAs-qoOGMWJ_UueYHng

6) Test plan and evidence of testing

```
def test_player_attributes(self):
    """Test if players have correct attributes after game initialization"""
    self.assertEqual(self.game.players[0].name, second: "Duncan")
    self.assertEqual(self.game.players[1].name, second: "ai-Owen")
    self.assertFalse(self.game.players[0].is_ai)
    self.assertTrue(self.game.players[1].is_ai)
    self.assertEqual(self.game.players[0].money, second: 1500)
    self.assertEqual(self.game.players[1].money, second: 1500)

def test_abridged_game_mode(self):
    """Test if abridged game mode is properly initialized"""
    abridged_game = Game(self.players, game_mode="abridged")
    self.assertEqual(abridged_game.game_mode, second: "abridged")

def test_timed_game(self):
    """Test if game with time limit is properly initialized"""
    timed_game = Game(self.players, game_mode="full", time_limit=1800)
    self.assertEqual(timed_game.time_limit, second: 1800)
    self.assertTrue(hasattr(timed_game, "start_time"))

def test_free_parking_pot(self):
    """Test free parking pot functionality"""
    self.assertEqual(self.game.free_parking_pot, second: 0)
    self.game.game_actions.add_to_free_parking(50)
    self.assertEqual(self.game.free_parking_pot, second: 50)

def test_ai_difficulty(self):
    """Test if AI difficulty is correctly set"""
    easy_game = Game(self.players, ai_difficulty="easy")
    hard_game = Game(self.players, ai_difficulty="hard")
    self.assertEqual(easy_game.ai_difficulty, second: "easy")
    self.assertEqual(hard_game.ai_difficulty, second: "hard")

def test_board_initialization(self):
    """Test if the board is properly initialized"""
    self.assertIsNotNone(self.game.board)
    self.assertEqual(len(self.game.board.spaces), second: 40)

def test_initial_player_positions(self):
    """Test that players start at position 1 (Go)"""
    self.assertEqual(self.game.players[0].position, second: 1)
    self.assertEqual(self.game.players[1].position, second: 1)
```

```
< Tests passed: 31, ignored: 1 of 32 tests - 24sec 103ms
collecting ... collected 32 items

test_game.py::TestGame::test_abridged_game_end_calculation
test_game.py::TestGame::test_abridged_game_mode
test_game.py::TestGame::test_ai_difficulty
test_game.py::TestGame::test_bankruptcy
test_game.py::TestGame::test_bankruptcy_from_bank_payment
test_game.py::TestGame::test_board_initialization
test_game.py::TestGame::test_build_hotel_mechanics
test_game.py::TestGame::test_build_house_mechanics
test_game.py::TestGame::test_buy_property
test_game.py::TestGame::test_card_actions
test_game.py::TestGame::test_card_distribution_equality
test_game.py::TestGame::test_card_exhaustion
test_game.py::TestGame::test_dice_roll
test_game.py::TestGame::test_free_parking_pot
test_game.py::TestGame::test_game_initialization
test_game.py::TestGame::test_game_over_detection
test_game.py::TestGame::test_initial_player_positions
test_game.py::TestGame::test_invalid_game_mode
test_game.py::TestGame::test_jail_mechanics
test_game.py::TestGame::test_max_players
test_game.py::TestGame::test_mortgage_mechanics
test_game.py::TestGame::test_movement_past_go
test_game.py::TestGame::test_negative_money_handling
test_game.py::TestGame::test_player_asset_calculation_with_mortgaged_properties
test_game.py::TestGame::test_player_attributes
test_game.py::TestGame::test_player_movement_after_roll
test_game.py::TestGame::test_property_group_completion
test_game.py::TestGame::test_property_ownership_restriction
test_game.py::TestGame::test_rent_payment
test_game.py::TestGame::test_station_rent_calculation
test_game.py::TestGame::test_timed_game
test_game.py::TestGame::test_utility_rent_calculation

===== 31 passed, 1 skipped, 1 warning in 24.67s =====
```

```
===== test session starts =====
collecting ... collected 1 item

test_game.py::TestGame::test_initial_player_positions

===== 1 passed, 1 warning in 1.16s =====
PASSED [100%] Attempting to load player image from: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/image/Playertoken (1).png
Successfully loaded player 1 image
Player Duncan initial position: 1
Initialized Easy AI controller for ai-Owen
Attempting to load player image from: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/image/Playertoken (2).png
Successfully loaded player 2 image
Player Owen initial position: 1
Game initialized in Full mode (no time limit)
Loading dice image: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/image/Dice/1.png
Loading dice image: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/image/Dice/2.png
Loading dice image: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/image/Dice/3.png
Loading dice image: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/image/Dice/4.png
Loading dice image: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/image/Dice/5.png
Loading dice image: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/image/Dice/6.png
Loading property data from: /Users/stuartbaker/Downloads/Software-Engineering-Project-Build-28.03.2025-212616/assets/gamedata/PropertyTycoonBoardData.xlsx
Successfully read Excel file. Found 49 rows
Processing position 1: Go
Processing position 2: The Old Creek
Processing position 3: Pot Luck
Processing position 4: Gangsters Paradise
Processing position 5: Income Tax
Processing position 6: Brighton Station
Processing position 7: The Angels Delight
```

Three screenshots have been included showcasing our unit testing script. We have included 32 unique unit tests, testing everything from the correct player attributes to buying a property correctly. All 32 test cases run, but one of them is ignored due to it containing the line 'self.skipTest()'. Some of the unit tests focus on edge cases, and they all performed as expected. We had mitigated any potential errors here during development by designed the code to prevent edge case issues happening in the first case.

No bugs were thrown up during unit testing, which is a very positive outcome. However, errors and bugs did occur during actual gameplay, both when we left the game to run for three hours with AI agents, as well as when we played the game as a team and with AI. A few of the issues we found were the bankrupt logic not working correctly and the property cards showing up one round late. We haven't fixed these issues yet, but we have added them to our bug log that we are keeping updated and working through.

We haven't included an image for our system testing, but we have created a YouTube channel to showcase our game running and to present some of our test games. This is the link to our channel: https://www.youtube.com/channel/UCfA_oAs-qoOGMWJ_UueYHng. We will continue to post a few more videos demonstrating the game, and it will provide us with a visual tracker of development.

7) Summary of sprint

We gave ourselves an extra week for this sprint as our aim was for this to be our final sprint in order to meet our proposed deadline to complete game development, at the end of week 9. So instead of the usual two weeks for the cycle, we gave ourselves three weeks. The tasks for this sprint were relatively similar to each other and had a lot of overlapping components, but they were also scheduled in to take a longer duration to complete. This was because they were final checks and enhancements to make the final product better. All of the tasks were completed, but to varying degrees of success, for instance, there are still a few bugs we are yet to eliminate, and certain areas of code could be changed to become a bit more maintainable and readable for future developers. However, these are only small and relatively insignificant in the grand scheme of the project, as we have produced a fully functioning and working game to present to Watson Games.

Our working prototype is now very close to being delivered and presented to Mr Raffles and his team. There are a few minor bugs and tweaks we still wish to address, but these are very small. We have a document that lists each bug to fix and our thoughts on the best way to tackle it, which we are now slowly working through.

We collectively feel this was a very strong sprint completion, as even though we only implemented one major change, being the UI, the results and knowledge we gained from this sprint were invaluable. We are extremely happy with the UI overhaul as this was one of our main objectives for this sprint because we didn't have time to address it in the previous sprint. The final design has come out much better than we had anticipated, and we are pleased that one of Mr Raffles representatives were impressed with the design. It is a much cleaner and clearer UI.

All of our unit and system testing worked very well. The unit testing threw up no bugs or issues, which is exactly what we were hoping for. We won't allow this to make us complacent as we know we may have missed some errors, so we will continue to test and keep an eye out for any bugs. The system testing proved a little more useful as it did present us some bugs to fix, albeit not a huge amount. We have produced some YouTube videos showcasing our system testing, which will provide us with a good base to look back on when we do change certain features and fix specific bugs.

What didn't work so well was changing the code to make it more readable and maintainable. We have managed to transform the code, so it has a lot more consistency with its formatting, but we still have a fair amount of redundant and duplicated code we need to handle. We also haven't added as many comments as we would like. We are conscious that we need to be very careful when changing the code because we don't want to create future problems from

poorly edited code, so we are checking multiple times everytime we change a section that it hasn't created new bugs and errors. Furthermore, we need to add more comments into the code as some parts aren't too easy to figure out, so additional comments would prove invaluable to any new developers looking into our task.

This has made us realise that for future projects we will need to ensure we write our code in a more efficient and readable manner, ensuring we avoid including redundant code and inconsistencies all over.

During seminar sessions, we have followed on from the previous sprint and continued to ask Mr Raffles and his representatives questions about our project and what we can improve. We are very happy that his team is pleased with our current product and had no real improvements or adjustments that they would want, asides from the few bug fixes. We had addressed their previous comments regarding overlapping text and kept an eye on any questions posed to the discussion forums.

To conclude, this sprint was very successful. We have achieved our goal of a week 9 deadline for the development of the game, and if asked to, we are confident that if we submitted the game in its current state, we would satisfy almost all the requirements. However, we had planned in two weeks of slack time before the final submission date, and we intend to use this time to finish some bug fixes from our bug document, finalise all of our deliverables, and check we are happy to submit.