Tauresium Test Plan

This document will detail the different tests made on the Tauresium program after the first working build was completed. All examinations in this document fall under one of three types, acceptance, edge case and erroneous, which are defined as the following:

* Acceptance – Input that is to be expected from an average gameplay session.
* Edge Case - Input that lies on the boundary of what is expected to be a reasonable parameter.
* Erroneous – Input that is specifically meant to be unacceptable and handled with an appropriate error response.

Using these three types of testing, this document will review each major feature of Tauresium to ensure the submitted release of the project will not contain any major errors. For the purposes of this log, the following assumptions will be made:

* The user using a modern browser, ideally chromium based or similar. Any issues that arise from using an outdated or unsupported browser are likely to be client-side only and the responsibility of the user to maintain.
* The user does not modify the source code in any way except to make intentional erroneous inputs or HTTP requests. This means any issues particularly concerning presentation that occur when the original document is modified are not to be examined.
* The user does not have any form of administrative privileges or access to the backend database – this discludes tests that would not be possible in a regular game session – such as the forced deleting of an account when a session is active.
* All inputs are assumed to be sequential, this means tests that concern the timing of inputs will not be considered, such as if two players attempt to take the same location at once. This assumption is applied due to the lack of resources available to conduct said tests.

# Test 1 – Page accessibility (Logged Out, Acceptance)

*Description*:In this test, I attempted to access the pages naturally accessible by logged out users, these include 5 pages – The index, the new world creation menu, the country creation menu, the API page, and the tutorial page.

*Result*: All pages listed were successfully accessed using their buttons provided. Pages such as the API page did not include the information not applicable to a logged-out user, notably the APIKEY. This feature works as intended.

Successful

# Test 2 – Page accessibility (Logged Out, Edge Case)

*Description*: This test concerned the accessing of pages not normally provided to users through direct means such as menu buttons – but should still nonetheless be accessible by logged out users. This includes: The session success page, the login failed page and the error page.

*Result*: Both the login failed page and general error page were accessible through URL entry, as expected. Once accessed, the session success page caused a redirect to the error page, due to no parameters supplied in the URL – an expected result – attempting to provide this page with correct parameters as would be usually applied by the website allowed for the correct loading of the page. This feature works as intended.

Successful

# Test 3 – Page accessibility (Logged Out, Erroneous)

*Description*: This test covers the access of session specific pages – notably the Main page, provinces page, session stats page and two event pages. These pages should cause a redirect to the error page when accessed, as they are only meant to be usable by logged in players.

*Result*: All pages successfully redirected to the error page when accessed, a process which occurs through the backend, meaning the page resources should not have been loaded before the redirect occurred.

Successful

# Test 4 – Page accessibility (General, Erroneous)

*Description*: This test will cover the accessing of files that are not meant to be accessed via the user but exist within the scope of the website and therefore can be loaded. This includes files like the top bars which are embedded within pages when appropriate, as well as scripts normally used by the website.

For the purposes of this test, any errors are only of concern if they cause changes to be made to the database or allow access of unintended resources. As these pages are not meant to be normally viewed on their own, any client side issues that occur and do not persist between pages are the responsibility of the user.

*Result*: When attempting to use scripts, two incidents of note occurred. Firstly, when the login verify script was entered, the page attempted a redirect to an error page within the scope of the Scripts file, which did not exist. Secondly, when the PageUpdate script was accessed, a PHP error occurred when the script attempted to include a file that did not exist – this is a notable issue but one that has no impact on anything but the client accessing the page, and therefore the fixing of this issue is low priority.

When loading the PageElements files, JavaScript issues occurred and resources such as the stylesheet were not correctly loaded, but again this is to be expected when these files are accessed in this way. Similar issues also occurred when loading PageElements files.

In conclusion these issues are not dangerous but fixing these issues should be considered if possible, but this is not a high priority task, due to the nature of how these issues can be accessed.

Low Priority/Minor Issues

# Test 5 – Login (Acceptance)

*Description*: Entered a valid login already stored in the database into the top bar login page as well as the login page accessed when an invalid login is provided.

*Result*: Both pages were able to successfully load the player into their game and store their session variables as expected.

Successful

# Test 6 – Login (Edge Case)

*Description*: Attempted to log into accounts using incorrect case in logins, which is expected to not permit a user to login.

*Result*: All attempts failed to login, entered logins must be in the same case they were originally created in.

Successful

# Test 7 – Login (Erroneous)

*Description*: Attempted to login with no parameters supplied, only a username or password supplied, an invalid username with a valid password and vice versa supplied, and input of a large number of characters, as well as input of non-Latin characters.

*Result*: All logins failed as expected, when a large number of characters were entered the response was slower than other responses, but the login still failed.

Successful

# Test 8 – Create a world (Acceptance)

*Description*:Attempted to create a world with the following parameters:

* World name: Geb
* Map Type: Earth
* Game Speed: Normal

*Result:* The world was successfully generated and the world code “**HHZLQT1FLNLS5MT9**” was provided.

Successful

# Test 9 – Create a world (Edge Case)

*Description*: Attempted to create a world with a duplicate name, Geb, and a game speed of slow, using the map type earth. This should not cause issues, as the unique identifier for the world is the world code, not the name, and the game is set to support all four speeds.

*Result*: The world was generated as expected, providing the world code “**ARQP89DDUA68HF71**”. Checks of the database of this and the previous test showed that both worlds existed in memory.

Successful

# Test 10 – Create a world (Erroneous)

*Description*: Attempted to create a world using numeric characters and with no inputs provided for other properties

*Result*: The page redirected to the world creation screen with three error messages describing the issues that had occurred – the non-alphabetical world name and the lack of inputs for other fields.

Successful

# Test 11 – Create a world (Erroneous 2)

*Description*: Used the developer console to remove restrictions on word length on the client side, as well as allow the selection of invalid properties. The submission was as follows:

* World Name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
* Map Type: TBA
* Game Speed: e

*Result*: Three errors occurred as expected, one mentioned the invalid name of the world, as it exceeded the character limit, and the other two errors noted the invalid map type and game speed. The latter two errors may only occur if the user modifies the code in this way, as there is a separate message for missing inputs and there are no invalid inputs that can be selected on the page. As with the last test, neither world appeared in the database.

Successful

# Test 12 – Create a country (Acceptance)

*Description*: Created a new country in the world “**HHZLQT1FLNLS5MT9**”, using the following parameters:

* Country Name: England
* Password: 1234
* World Code: HHZLQT1FLNLS5MT9
* Government Type: Democracy
* Colour: Red

*Result*: The country was successfully generated and added to the world. Upon generation, “England” was granted the province “**Europe\_Croatia**” and all starter buildings were constructed. Logging into this account was successful and showed the aforementioned information.

Successful

# Test 15 – Create a country (Edge Case)

*Description*: Attempted to create a new country using the maximum number of characters possible as the nation name, as well as a single space as the password. All other fields were valid.

*Result*: The inputs provided caused the program to redirect to the error page, rather than redirect to the same page with error messages appended. This only occurs when all basic checks have passed but the input provided has caused another error. After retesting this, the error seems to have come from the password. This issue may require fixing, but it should be noted that no modifications were made to the database before the error occurred, meaning the only consequence of this problem was the redirect to the error page rather than to the create a country screen.

Low Priority/Minor Issues

# Test 14 – Create a country (Erroneous)

*Description*: Attempted to create a country without any parameters supplied

*Result*: The user was redirected back to the create a country page with error messages supplied that told them that the information was missing.

Successful

# Test 15 – Create a country (Erroneous 2)

*Description*: Entered invalid characters into the username, password, and world code boxes, with valid government and colour inputs.

*Result:* input of username and world code issues was detected, but invalid passwords were permitted. Like in the former edge case, the country was not appended to the database but did cause a redirect to the general error page, rather than the intended account creation page. This means the password input is liable to errors, which are likely occurring when the password is hashed. This issue may need resolving.

Low Priority/Minor Issues

# Test 16 – Create a country (Erroneous 3)

*Description*: Entered a name in use by a player as the nation name, a world code of a full world, and a colour in use by a player in said world. These inputs should cause issues, as the nation name is intended to be a unique identifier, worlds should not permit new users after reaching their capacity, and colours must be unique in a world.

*Result*: Nation name and world name were rejected, and the error was reported appropriately. Due to failing the world name check, the colour code was not checked, further testing showed that entry of a new country with a duplicate colour was not permitted by the system in a non-full world.

Successful

# Test 17 – Create a country (Erroneous 4)

*Description*: Used developer console to allow inputs that exceed the nation name size limit and the world code size limit, as well as permitting the entry of invalid government types and colours.

*Result*: All inputs were reported as invalid when the player was redirected to the country creation page, including the country colour (as it checks if the colour is valid before checking if it is used within the world code provided). No new data was added to the database as a result of this, meaning the country creation did not succeed, as expected.

Successful

# Test 18 – Page accessibility (Logged In, Acceptance)

*Description*: Players with an active session should be permitted to access certain pages, including all pages accessible to logged out players as well as the main pages, province page and event pages. This test was conducted using both the buttons available on the website, as well as URL inputs.

*Result*: All pages were accessible by the logged in player. It should be noted that URL inputs redirected to the error page when no parameters were supplied for the provinces and event results pages, this result is expected.

Successful

# Test 19 – Page accessibility (Logged In, Edge Case)

*Description*: Logged in players may be able to access pages originally intended for logged out players – there should be no issues that arise from these pages being accessed, however it is worth testing to be sure.

*Result*: When accessed using a logged in account, no issues arose from loading the pages, including the “login failed” page, which is not usually accessible under normal circumstances, but does not present any problems. Attempting to use the login failed page to login to a second account while previously logged in does not seem to cause any issues, as the session variables used to store the players state are all overwritten when the login script is accessed, meaning the session is treated the same as if it was logged out and back in.

Successful

# Test 20 – Page accessibility (Logged In, Erroneous)

*Description*: While there are no pages that a logged in player cannot access, this does not account for events that may occur when a page is loaded but the player is logged out through external means (For the case of this test, by opening a second tab and logging out). This test will attempt to access certain high-risk scripts or pages while the session is unloaded in a separate tab, the subjects being: Events, annexation, and building.

*Result*: When attempting to answer an event while the session is removed, the event is not completed, and the user is redirected to an error page. Similar events occur when an annexation is attempted, both for the cases of peaceful annexation (culture and economic) as well as military annexation. Interestingly, the same is not true for buildings. When a building request is made while the session does not exist, the building is constructed before the player is redirected to an error page, with the appropriate balance still being deducted from the player and the building bonuses applied – it even prevents building from occurring when the player does not have the required influence. This is likely because the session variable API key is being loaded when the provinces page is being accessed – so the AJAX call is using the stored value for the API key, which allows it to conduct the request while the session does not exist server side any longer. This issue is minor at best, as it does not get in the way of players interacting with the game, nor does it allow for players to exploit the game in any way, therefore it is likely this issue will not need to be fixed.

Low Priority/Minor Issues

# Test 21 – Event generation

*Description*: This test has no type as it is a feature outside the control of the user. Event stacks are generated based on world speed and use the last online time to calculate the event stacks that should be added, storing a decimal value equal to the amount of time passed divided by the event speed – this means if 45 minutes pass for a 30-minute event speed game, the events should be incremented by 1.5 event stacks. These stacks are limited to 5 at any time and cannot be less than 0. To test this, I reset the event stacks to 4 and let a minute pass on a 30-minute event timer, which should result in an increase of 0.03333 event stacks.

*Result*: The event stacks were set to 4.03278, which, an increase of 0.03278. This is almost exactly 0.03333, with the error being only 0.00055 stacks missing due to the decimal limit of 5 decimal places – which is a small enough margin of error to not be impactful to a user’s experience. This feature seems to work as intended with minimal issues, and even works while the server is offline – as the database stores the last online time of a player and compares that value against its own time.

Successful

# Test 22 – Event loading (Acceptance)

*Description*: For this test I will attempt to load a new event from memory using an event stack, if this functions correctly it should remove an event stack and load a random event into the game, with three relevant options.

*Result*: The game decreased the players event stacks by 1 and loaded the event ‘Deicide’ with its three relevant options. Leaving this page and opening the event page again loaded the same event without decreasing any stacks, demonstrating that once an event is loaded into memory the player can open it again at any time.

It should be noted that with event loading, there is instances where the developer console will receive a failed GET request. This is an intended feature, the webserver first sends a GET request to the server to see if an event exists, with a failed request signifying a new event needs to be loaded by using a PUT request. The response code for this operation is an error response code, and thus some consoles will report it as an error, despite it being an intended feature.

Successful

# Test 23 – Event loading (Acceptance 2)

*Description*: For this test I attempted to load an event from the button that appears on the event completed screen (which only is visible if the player has 1 or more event stacks left), to ensure that this button also properly functions.

*Result*: A new event ‘Munch Squad’ was loaded for the player, leaving this page and re-entering it kept the event as with the first event load test. Attempting this test again but without pressing the ‘next event’ button allowed the player to load a new event by pressing the event button on the top bar, as is expected.

Successful

# Test 24 – Event loading (Edge Case)

*Description*: This test concerns the case in which a player attempts to load an event when they have exactly one event stack, the bare minimum amount that a player can have before an event can be loaded.

*Result*: The event ‘Deicide’ was successfully loaded despite the player having just barely enough stacks to load the event – demonstrating that the event loading system does not have any issues when dealing with exact values.

Successful

# Test 25 – Event loading (Edge Case 2)

*Description*: This test examined the case in which a player exits the event page after using their last event stack to load an event – giving them 0/5 event stacks. If the website functions as intended, the player should still be able to access the event page to resume their previously loaded event.

*Result*: When exiting the page, the top bar event button was still available despite the lack of event stacks. When pressed, the player was allowed to resume their event as normal, after answering the event button was removed as the player now had no event stacks remaining nor any event loaded.

Successful

# Test 26 – Event loading (Erroneous)

*Description*: For this test, I used up all the event stacks loaded by the player before attempting to load a new event either through any buttons available or the load event URL.

*Result*: Once the final event was completed, the button allowing a player to use their next event did not appear, and neither did the top bar event button. Manually entering the load event URL caused a redirect to the error page and did not load an event for the player nor use any event stacks, an expected effect when a player attempts to cheat the system by loading an event without having the necessary events.

Successful

# Test 27 – Event answering (Acceptance)

*Description*: For this test I loaded the event ‘A New Form of Warfare’ with an account set to 1 in all focus point generation categories as well as 0 influence for said categories. I then selected the answer ‘Modernising the nation is certainly important, but the costs outweigh the benefits’, which is an option set to grant the player +0 culture generation, +0.03 economic generation and -0.03 military generation. Additionally, as the event is worth 60 influence, the player is expected to have the following statistics on selection of the aforementioned option:

* Culture Generation: 1
* Economic Generation: 1.03
* Military Generation: 0.97
* Culture Influence: 60
* Economic Influence: 61
* Military Influence: 58

*Result*: When answered, the player received the exact bonuses specified by the description, exactly as expected. Additionally, it should be noted that since the influence are stored as integers, new influence is always rounded down to the nearest integer, hence why economic influence (which would normally receive 61.8 influence) only received 61 points.

Successful

# Test 28 – Event answering (Acceptance 2)

*Description*: This test took place immediately after the last and concerns the second event loaded after a player has received their rewards, and if the rewards received are added accurately. The next loaded event was ‘Creature Comforts’ and the selected option was ‘Cull the population to acceptable levels’, which is worth -0.01 culture focus, +0.03 economic focus and +0 military focus, with the event itself being worth 25 influence. Adding these stats to the previously attained values, the player should have the following:

* Culture Generation: 0.99
* Economic Generation: 1.06
* Military Generation: 0.97
* Culture Influence: 84
* Economic Influence: 87
* Military Influence: 82

*Result*: All values change as expected, demonstrating that the bonuses provided by events do change over time, and previous changes do impact new point gains – if the original 1.03 economic generation were applied, the player would only have received 25 economic points from this event, leaving their total at 86 points, hence proving the new additions to the influence count are influenced by the new generation levels.

Successful

# Test 29 – Event answering (Erroneous)

*Description*: For this test, I loaded the event ‘Rosebud’ and edited the first options tag in the developer console to rename it as ‘Option 4’, then I attempted to press the button to see what would occur as a result of this.

*Result*: Pressing the button with an invalid name redirected the player to the error page as intended. When resuming the game, I attempted to load the event again to see if the event had been answered regardless, but the event was still loaded into memory, and no influence or generation changes had been applied, leading me to believe the issue was caught before any errors could occur.

Successful

# Test 30 – Culture annexation (Acceptance)

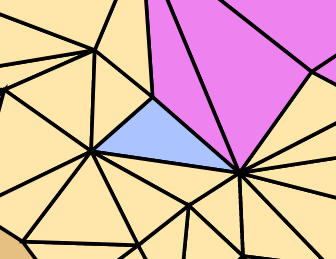
*Description*: Attempted to annex a location adjacent to a province owned by a player using culture influence. The location used for this was ‘**China\_SouthTibet**’, which had an adjacency with the player in question. This location was stated to cost 57 culture influence, and the player had a balance of 84 influence, thus meaning they should have 27 influence upon annexation.

Figure 1 China\_SouthTibet

*Result*: The location was successfully annexed by the player, adding the location to their nation. After annexation occurred, the players influence balance was set to 27, demonstrating that the correct amount of influence had been deducted.

Successful

# Test 31 – Culture annexation (Edge Case)

*Description*: Attempted to annex the location ‘**China\_WestTibet**’, a location worth 75 influence points, while the player attempting the annexation had exactly 75 cultural influence. This should allow the annexation to occur while leaving the player with a balance of 0 culture influence.

*Result*: The location was successfully annexed, and the cultural balance of the player hit 0 as expected.

Successful

# Test 32 – Culture annexation (Erroneous)

*Description*: Sent a cultural annexation request to the province ‘**China\_EastXinjiang**’ while having an insufficient balance to claim the province. The player should not be able to take this land, as they do not have sufficient influence to do so.

*Result*: On accessing the provinces page for the region, the button to attempt culture annexation was greyed out, with click events turned off for the button. I used the developer console to reenable the click events and attempt to send the annexation HTTP request, this resulted in a redirect to the error page as the script had detected the issue, and on accessing the map page, the land was not taken nor had there been any changes to the culture influence balance.

Successful

# Test 33 – Culture annexation (Erroneous 2)

Figure 2 SEAsia\_BurmaNorth

*Description*: For this test I attempted to annex the location ‘**SEAsia\_BurmaNorth**’ while having 999 cultural influence, more than enough to take any province on the map, but without any adjacencies to the location. This should pass the check for costs, but not the adjacency check, and therefore the annexation request should be denied.

*Result*: Much like the last test, as the location was not accessible, the button for the province was greyed out, preventing the player from sending an annexation request without changing the tag properties in the developer console. When this was reenabled, I sent an annexation request which resulted in a redirect to the error page, as the web server had noted that the location was not adjacent to any province.

Successful

# Test 34 – Culture annexation (Erroneous 3)

Figure 3 Occupied China\_EastGuangdong

*Description*: In this test I granted the location ‘**China\_EastGuangdong**’ to another player within the session, and attempted a cultural annexation of the location, this should not be possible, as only military annexation is permitted to take other players owned locations.

*Result*: As usual, the button for cultural annexation was not available due to the province being owned, but by editing the HTML in the developer console I was able to send a request to take the location, which once again ended in a redirect to the error page. It should be noted that the annexation button used is only forming a HTTP request to be sent to the API, hence the results for these requests would be the same when done through the API.

Successful

# Test 35 – Economic annexation (Acceptance)

*Description*: This test involved the use of economic points to annex locations in a local region, for this I used the previously mentioned location ‘**SEAsia\_BurmaNorth**’ which, while not adjacent to a province, exists in a coastal region the player has access to. This should mean economic influence can be expended to take the location, as the player has enough economic influence to do so.

*Result*: When accessing the province page, the economic button was available, and could be pressed to send an annexation request for 96 economic points, out of the players 100 economic points. When pressed, the player was granted the location and retained 4 economic influence as expected.

Successful

# Test 36 – Economic annexation (Acceptance 2)

*Description*: For this test, I granted a player a number of coastal locations in the Indian ocean region, allowing them to become dominant in the region. This meant the player should now be able to economically annex locations in the adjacent region of the East African coast (‘**Africa\_Somalia**’)**.**

*Result*: The button to take the location was available as expected, and annexing the location proved successful. Additionally, as expected, the cost of the location was increased by the regions ‘far away’ penalty, which adds an additional cost to sending economic annexation requests to far away locations. Finally, after annexing this location, the player now notably had 1 out of 4 coastal provinces needed for dominance in the East Africa region.

Successful

# Test 37 – Economic annexation (Edge Case)

*Description*: For this test, I attempted to take a local coastal region while having exactly the required amount of economic influence, if successful, the player should be able to take the location while retaining only 0 economic influence. For this test I used the location ‘**Madagascar\_Boeny**’, which has a cost of 76 economic influence.

*Result*: The player was able to take the location and was left with 0 economic influence as expected, similar to the results of the edge case cultural annexation.

Successful

# Test 38 – Economic annexation (Erroneous)

*Description*: In this test I attempted to send an annexation request to a coastal region while having insufficient economic influence to do so, much like the related cultural annexation tests, I used the developer console to allow me to create this request and the requests for all future erroneous economic annexation requests.

*Result*: Attempting to annex the location resulted in a redirect to the error page as expected, and the location was not taken nor was any economic influence deducted from the player’s balance.

Successful

# Test 39 – Economic annexation (Erroneous 2)

*Description*: This test attempted to annex an inland location using economic points, which should not be possible. To ensure this test is not biased by coastal region or point cost, the location chosen, ‘**Africa\_Zimbabwe**’ was a location within the players coastal region, and said players economic balance was set to 999, higher than any province should ever cost.

*Result*: As expected the annexation request led to an error page response and did not result in any changes to the map or point balance, demonstrating that players may not annex non-coastal locations using economic influence

Successful

# Test 40 – Economic annexation (Erroneous 3)

*Description*: This test concerned the sending of economic annexation requests to far away locations when the player does not have dominance in an adjacent coastal region. For this test I granted the subject player coastal land in the west Africa node, granting them 1 of 5 locations needed to travel to the adjacent South America East or Mexico node. This should not allow them to use economic influence to travel to this location.

*Result*: The player was unable to annex the location, as before, changing properties in the developer console to submit the request caused the error page to be displayed and no changes to be made, preventing the player from circumventing the game rules to their own advantage.

Successful

# Test 41 – Economic annexation (Erroneous 4)

*Description*: In this test I attempted to annex an enemy occupied location using economic points which, like cultural annexation, should not be possible – only military influence can be used to take other players provinces. This test used the previously mentioned location of ‘**China\_EastGuangdong**’, as was tested for cultural occupied annexation.

*Result*: As with similar tests, the player could not take the province, with attempts through console causing a redirect to the error page. This should mean that all possible avenues for exploits with economic annexation are covered.

Successful

# Test 42 – Military annexation (Acceptance)

*Description*: This test will concern the use of military annexation to annex an adjacent, unoccupied location. As military annexation can be done both through fulfilling the cultural annexation requirements or the economic annexation requirements, this test should demonstrate that an adjacent province can be taken using military points.

*Result*: Taking the location ‘**China\_EastXinjiang**’ using military points was permitted, due to the adjacency of the location to existing borders. This came at an additional +20% cost due to the lack of occupation in the region, and -5% due to adjacency. This meant the total cost came to 99 military influence, which the player was able to use to annex the location, and which was subsequently removed from their balance.

Successful

# Test 43 – Military annexation (Acceptance 2)

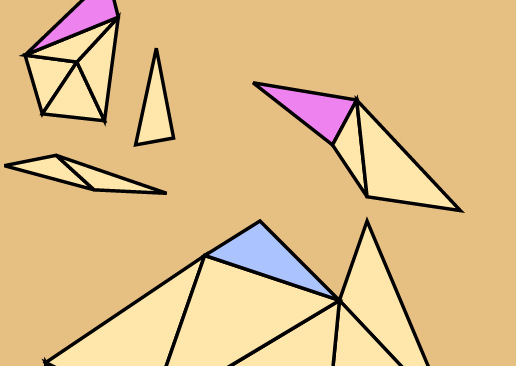
*Description*: As previously mentioned, military annexation is expected to work either through adjacency or sea routes, hence this test will ensure military points can be used to annex locations within a local region (a region that has at least one coastal city within the area). For this test I used the location ‘**Australia\_NorthernAustraliaNorth**’, which is present within the south east Asian region, of which the subject player has a coastal region within.

Figure 4 Australia\_NorthernAustraliaNorth

*Result*: The annexation request was successful, costing 117 military power for the player and granting them the location in exchange for that influence cost.

Successful

# Test 44 – Military annexation (Acceptance 3)

*Description*: This test will determine if the player can use military power to annex far away regions - locations which are adjacent to a location a player has dominance in, but which said player does not have any coastal provinces within. To this I granted the player 5 provinces in the West Africa region, which grants them coastal dominance in said region, and allows them to use their influence to annex the South East America coastal region.

*Result*: The player was able to annex a location within the South East America coastal region, costing an enormous 226 military power to do so. This is because using military power to take an province on a far away location costs an additional +100% military influence.

Successful

# Test 45 – Military annexation (Acceptance 4)

*Description*: In this test, the player will attempt to annex an adjacent occupied location using military power – this should be permitted, as players can take occupied locations using military influence. To do this I will be using the ‘**China\_EastGuangdong**’ previously used to test if a player could annex an occupied location using cultural influence or economic influence.

*Result*: The player was able to finally annex the location, taking it from its previous occupant after expending the necessary points to do so. This demonstrates military annexation Is possible for adjacent provinces.

Successful

# Test 46 – Military annexation (Acceptance 5)

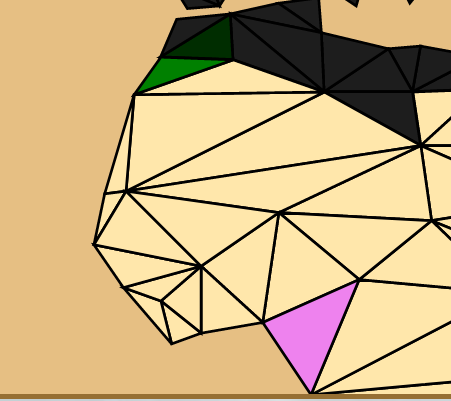
*Description*: This test concerns occupied military annexation within local regions, specifically within the West African region. To do this, I attempted to annex the location ‘**Africa\_WestSaharaState**’ which is present within the region, under the occupation of green.

Figure 5 Africa\_WestSaharaState

*Result*: The province was able to be annexed as expected, granting the player the land, and deducting the appropriate military influence, with a 20% bonus cost for not having adjacencies.

Successful

# Test 47 – Military annexation (Acceptance 6)

*Description*: This test will verify that the player is able to use military influence to annex occupied locations in far away regions, for the purposes of this test, I gave the green nation the previously occupied location in south America, making the location once again considered a ‘far away’ region for the pink country. This test should allow pink to take the location with an added cost.

*Result*: The pink country was able to annex the aforementioned location successfully, as usual with the correct province granted and the appropriate cost removed from the players military balance, with the previous mentioned +100% cost for far away regions being applied again.

Successful

# Test 48 – Military annexation (Edge Case)

*Description*: Like the other annexation edge case tests this test will examine if the player is able to annex a province while having the exact number of points required for said action. For this test, I granted a nation all but one province in the Indian subcontinent and used the last province (owned by another nation) as a point to attempt to invade from.

*Result*: Interestingly, this seems to be the sole annexation case in which the edge has an issue. When attempting to annex a province worth 84 military influence, the button to attempt annexation was unavailable, meaning at the very least the front end is not correctly handling exact values for military annexation.

Additionally, when I granted the player one more military point, they were able to annex the location and were left with 1 military point left, meaning the actual cost deducted is correctly displayed. However, when the front end is circumvented using the developer console to reenable annexation buttons, the annexation request still fails, redirecting to the error page as if the request were invalid.

Gameplay Issue

# Test 49 – Military annexation (Erroneous)

*Description*: This test will determine if military influence can be used to acquire provinces when the player has insufficient points. As with other tests of this type, this will be done by editing the HTML to allow the clicking of the annexation button, to send the request – this same effect can be done by manually submitting a POST request.

*Result*: Attempting to militarily annex a location with insufficient points will cause a redirect to the error page as expected, and will not permit the annexation to occur, nor will points be deducted.

Successful

# Test 50 – Military annexation (Erroneous 2)

*Description*: This test will concern the use of military influence to attempt to annex far away coastal regions without first having sufficient coastal power in an adjacent region. To do this, I will attempt a military annexation of the province ‘**SouthAmerica\_NorthChile**’ which is not in an adjacent coastal region to any provinces owned by the subject nation. This annexation attempt should fail.

*Result*: Once again using developer console to reenable pointer events, sending the annexation request to take a province in a region a player has no access towards causes a redirect to the error page, with no provinces being taken nor any points lost. Additionally, this test was repeated on both an owned and unowned province to ensure there are no differences, both of which had the same result.

Successful

# Test 51 – Military annexation (Erroneous 3)

*Description*: In this test, I will use the API directly to attempt an annexation request on a province owned by the player making the request. This is not possible in normal circumstances on the website, as the annexation page is replaced with the building page when it detects it is owned by the connecting player.

*Result*: The server responded with the default response for an invalid annexation request, indicating it detected the player was attempting to annex their own province. While the error message could be changed to reflect this, since it is not is possible in normal gameplay, it is not a priority to add this. However, since the annexation request was not successful, no provinces changed nor did any point balances.

Successful

# Test 52 – Military capacity (Acceptance)

*Description*: This test will attempt to exceed the military influence capacity of a player using events, to ensure the player is not able to get a balance over their limit. To do this their military influence will be set to one below the maximum and then an event will be completed, this player should then only reach their capacity.

*Result*: The event attempted to give the player an additional 46 influence points but only added enough to hit the capacity, hence meaning the capacity is enforced by the web server. However, the event screen does still say it added the original amount of military influence, rather than the amount adjusted for the capacity – while this is a small issue it could be worthwhile to modify this to reflect the real number of points added.

Successful

# Test 53 – Province page loading (Erroneous)

*Description*: The province page requires the name of a province to be passed to load information, this is set as a variable within the URL, and therefore the name can be easily modified by a player. This test will be used to examine the cases in which the province URL is modified to include no parameters as well as invalid province names.

*Result*: Both tests resulted in a redirect to the error page, as is expected. However, it is interesting to note that the redirect using no parameters was instant, whereas the false parameter test redirect was not. This is because the GET request for the URL parameters is done through PHP, whereas the search for the province name is handled using AJAX, hence the request for a redirect is conducted shortly after the loading of the page – this does not seem to cause any issues.

Successful

# Test 54 – Main map visibility

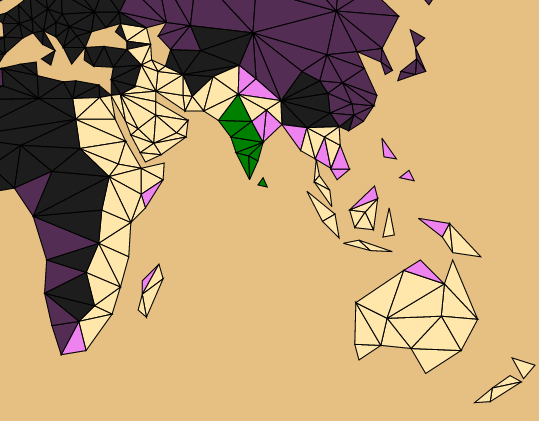
*Description*: This test concerns province visibility and texturing on the main page – all owned provinces should be highlighted in their respective national colour, and provinces that a nation is not adjacent to, nor has any coastal adjacency or coastal provinces within, should be greyed out to signify the lack of importance to the player. This test will determine if the view system is correctly implemented by limiting a certain players provinces to the India region and examining the view available to said player – if the functionality works as intended, this player should gain visibility of the east African region, the south east Asian region and any adjacent provinces.

Figure 6 The visibility provided to green.

*Result*: Once restricted to india, the green player was only able to view regions relevant to them, including the east african, indian and south east asian regions. Additionally, as can be seen at the top of the nation on the supplied figure, the player was able to view some adjacent locations that would not be normally visible if visibiltiy was restricted to solely coastal regions.

Successful

# Test 55 – Main map province costs

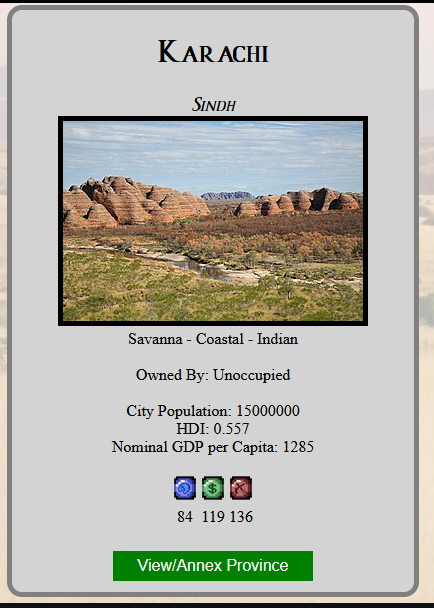
*Description*: When selecting a province on the map, an AJAX request is made to permit the player to read the province cost before entering the province page, this request should return accurate results, which include any modifiers that would be applied, such as military modifiers etc. This test will ensure the values stated on the short province view screen are accurate.

Figure 7 The shorthand province view window

*Result*: The values of a selected province are identical to that provided within the province view page, and even are not displayed when the province selected cannot be taken via that method. It should be noted that this system has a flaw in that any province not annexable by the player displays the message “inaccessible”, even for their own province, potentially confusing newer players.

Successful

# Test 56 – Construction (Acceptance)

*Description*: Building a province in a location owned by a player allows players to increase certain statistics and is highly beneficial to the late game operation of a session. Player should be able to construct buildings in their owned locations provided they have sufficient points and have previously constructed prerequisite buildings. This test will determine if the player is able to complete the action of constructing infrastructure in their own province as intended.

*Result*: The province first selected for this task was the location ‘**India\_Rajasthan**’, which is a cultural province, and hence has access to both culture and military buildings. On testing, the player was successfully able to construct buildings of both types, provided they had the valid points required. This test was then repeated on the location ‘**India\_SriLanka**’, a military province with access to both military and economic buildings. Tests here once again permitted the construction of buildings in this location as expected.

Successful

# Test 57 – Construction (Edge Case)

*Description*: This test will be used to determine if constructions can be completed when the player has the exact number of required points for a construction to be made, as the aforementioned similar issue with military has brought into question if this problem exists elsewhere within the game.

*Result*: All tests on each point type returned successful, with the player able to construct the buildings when they had exactly the correct number of points, and the post-deduction point balance for all three categories being 0 influence.

Successful

# Test 58 – Construction (Erroneous 1)

*Description*: If a player has an insufficient balance of influence to construct a building, they should not be able to complete the construction, nor should any influence be deducted. This test will attempt construction of buildings when the player has an invalid amount of points available.

*Result*: Any construction attempts made with insufficient influence balance caused the page to be reloaded with an error message telling the player their point balance is insufficient to complete the transaction – after this, no points are deducted. This test was repeated with all three-point types to ensure the result remained the same, and all tests resulted in the player being unable to construct the buildings.

Successful

# Test 59 – Construction (Erroneous 2)

*Description*: The API for this website grants players the ability to submit construction requests manually, whereas through the website the HTTP requests are constructed automatically. This means while the result for the point balance will be the same for both, there is the possibility to create illegal requests through the API – for this test, I will attempt to construct a building using the API that is of an invalid type – as in a cultural province, you may not construct economic buildings, this test will attempt to send an economic build request to a culture province using the API. To do this, the following request (with type POST) was sent:

***http://127.0.0.1/Tauresium/TaurAPI/Building/India\_Rajasthan/4M2UU5PQEVFLF14N/E***

*Result*: The request returned an error code 400 with the appended message “Invalid building”, specifying that the supplied building type (E for economic) was not a possible building type within the location.

Successful

# Test 60 – Construction (Erroneous 3)

*Description*: In previous builds of the program, an unseen error existed in which the player was able to construct buildings that did not exist by sending an API build request to a province with the supplied influence type having all possible buildings completed, this did not cause any bad effects to the program, as the invalid buildings had no properties or cost, but was still fixed to avoid any other related issues occurring. This test will ensure the measures provided were sufficient to prevent this error, to do this I will max out the buildings available in a location, before sending a request to construct a new building using the following request:

***http://127.0.0.1/Tauresium/TaurAPI/Building/India\_Rajasthan/4M2UU5PQEVFLF14N/C***

This request will attempt to construct a building in the province that should not exist.

*Result*: The server responded again with a “400 Invalid building” error, demonstrating that building could not be constructed, as is expected, additionally, checking the database ensured that the stored building parameters were not invalid, and had stayed at its correct ID – C4. This test was then repeated for all three influence types, all of which had the same result.

Successful

# Test 61 – Construction (Erroneous 4)

*Description*: Due to building POST requests being user specific, an API key is required, signifying the player who is attempting the request. In this test, I will attempt to create a build request using the API key of another player to a province that they do not own. This should not be possible – only the building owner should be able to submit build requests, and therefore sending the API key of another player should not allow the request to pass. To do this I used this POST request:

[***http://127.0.0.1/Tauresium/TaurAPI/Building/India\_MadhyaPradesh/YG5YDTTTHM819CAK/C***](http://127.0.0.1/Tauresium/TaurAPI/Building/India_MadhyaPradesh/YG5YDTTTHM819CAK/C)

The provided province is a province owned by green, whereas the API key provided is owned by the pink nation, this request should therefore not succeed.

*Result*: The API responded with “400 You are not the owner of this location” – demonstrating that the API is able to determine that the player with the provided API key should not be permitted to perform this action. In addition, the building was not constructed and neither players influence balances were modified in any form – showing that the construction request was not completed.

Successful

# Test 62 – Military capacity bonus

*Description*: When a player constructs a building, they will receive bonuses to certain statistics, one of which is their overall military capacity. This test will examine the before and after military capacity of a player when they create a new building providing this statistic. This will be done by constructing a “**Military Base**” (**M1**) on a location, which provides +10 military capacity. Before starting this test, the subject player had a military capacity of 405.

*Result*: Upon construction, the players military capacity was incremented to the value of 415, demonstrating that the capacity increase is properly implemented for the player.

Successful

# Test 63 – Military capacity bonus 2

*Description*: When a province is taken, all constructions in the location should be taken with it – hence, when a player loses a province, the provinces initial military capacity bonus and the bonus from any buildings should be lost. This test will see one player take a province with the aforementioned “**Military Base**” (**M1**) construction, to observe the effects on the military capacity of both players after this action. The province itself provides an initial bonus of 15, which is then incremented by 10 due to its constructions, giving it a total military capacity bonus of 25 – as such, the annexing player (with their current military capacity being 1835) should receive a new total of 1860 military capacity, whereas the original owner (with their capacity of 415) should have their capacity lowered to 390. Additionally, the original owner’s military influence was set to 415, to see if there are any issues that occur when a player’s military capacity is reduced past their current influence.

*Result*: Upon annexation, the annexing players capacity was set to the expected 1860, and the capacity of the original owner was lowered to 390, both properly representing the transfer of 25 military power. Additionally, the original owners current military influence was then lowered to 390, demonstrating that the lowering of military capacity will deduct military influence if required. It should be noted however, that this effect will only take effect the next time a request is made, or the account is active on the website – therefore if a GET request is made to the original country, it may show their unadjusted military influence, but this should automatically be changed whenever any request is made, and therefore cannot be abused in any way.

Low Priority/Minor Issues

# Test 64 – Build cost modifier

*Description*: Another statistic that can be improved by buildings are the costs of future buildings to the location, players may wish to prioritise buildings that lower future building costs to avoid wasting influence if possible. This test will ensure the building cost reduction is applied both:

* After the building is constructed
* At the correct amount
* With all other build cost modifiers applied additionally to create the total reduced cost.

To do this, I will first construct the building “**Cathedral**” (**C1**) which will cost 50 cultural influence to build, I will then construct the “**Military Base**” (**M1**) which, while originally costing 50 military influence, should now cost 45 military influence due to the 10% cost reduction of the previous building. After this, I will construct the “**Train Station**”(**C2**), followed by the “**University**” (**C3**), the latter of which provides a -25% build cost, then I will construct the “**Airfield**” (**M2**) which, while originally costing 100 military influence, should use the reduced cost of 65 military influence due to the 35% cost reduction.

*Result*: The first construction cost the expected amount, 50 culture influence, signalling that the build cost modifier is not appended until after the construction is complete, after this I attempted to create the military base as previously mentioned, which had the reduced cost of 45, which was correctly deduced from the balance with the adjusted price. After this, I then built the other two culture buildings granting me -35% build cost, and then used my remaining military influence to buy the airfield, now worth the expected 65 military influence, which was correctly deducted at this correct price. It seems the build cost modification works entirely as intended.

Successful

# Test 65 – Local defensive strength

*Description*: The last available construction modifier, local defensive strength, refers to the additional military penalty cost for attempting to annex the location – this statistic is strong, and most buildings that grant this effect do so in large amounts – in some instances, I have observed province costs growing to roughly 400 military influence to take, due to the player constructing all the possible defensive buildings in that location. This test will determine if the defensive strength modifier is correctly applied, by noting the before and after military cost of a province when the local defensive strength is incremented. The province used for this, while having no defensive strength modifiers, cost a total of 98 military points, and while the exact order at which modifiers are applied might stop the result from being an exact double of this (as the building used, “**Airfield**” (**M2**) should provide +100% cost), any significant increase in cost will be counted as a success.

*Result*: After constructing the +100% defensive strength building, the originally 98 military cost provinces incremented to a cost of 166, which is a roughly 70% cost increase, as previously mentioned, the exact order in which cost modifiers apply will stop any changes from being exactly 100%, especially as the original cost already had military modifiers applied (due to adjacency and other factors) which would be not changed by the new construction modifier, therefore this will be counted as a successful test.

Successful

# Test 66 – Local defensive strength 2

*Description*: This test will determine if the defensive strength bonuses are added together, this test will use the previous test province and construct additional defensive strength buildings to examine any changes to cost, this test will be subject to the same rules as the former, with any significant increases in cost being counted as a success. To do this, the building “**Cathedral**” (**C1**) Will be constructed, which applies a +20% defensive strength modifier.

*Result*: After construction, the cost to take the location increased to 180 military influence, which is a roughly 83% increase from its original cost of 98, and an 8% increase over its former value of 166. This is an expected and successful result, and interestingly it should be noted that the percentage increase reported in the province screen is an accurate representation of the real modifiers applied to the military cost, with the original cost of the province in question, “**India\_MadhyaPradesh**”, being worth a base of 68 military influence – with the province screen reporting a 165% increase in cost, which when calculated equals the expected 180 military influence (when rounded down).

Successful

# Test 67 – Session stats page

*Description*: This simple test is just used to ensure the session details screen correctly loads player details and will be tested by changing the number of occupied provinces of a player to ensure the values correctly update as expected. The session stats page stores some statistics of each player in the game, with more details available using the API. To do this test, I used a player with 79 provinces, 1835 military capacity and 3 of 4 provinces needed for dominance in the East African coastal region, I then annexed a new coastal location in the East African region, which should increment the provinces number to 80, the military capacity to 1850, and set the player to dominant in the East African Coast region.

*Result*: All values were changed to their expected value, and the same occurred when testing the taking of a province via military annexation, verifying that the statistics on the page are valid.

Successful