**Idle Planet – Analysis**

**- George Kerr -**

1. **Name: Idle Planet**
2. **Overview**

The aim of my project is to create an incremental, idle/tycoon style game, where the user is able to develop different planets in a galaxy, clicking them to earn an in-game currency which can be spent to progress them further. The main stat of each planet will be population, increasing as the player progresses and influencing money production. Idle and tycoon games are games where the player progresses through simple, repeated tasks (like clicking something) or automated processes that run without the user’s input and can be upgraded to increase their generation of a currency. Despite their simplicity, these types of game have a lot of potential for customisation through various features like upgrades, different worlds/scenes and prestige mechanics. The problem with some idle games is that they all involve the user following the same set of upgrades, which leads to the same tycoon at the end with little customisation. When I play idle games, I often notice many opportunities there are for more features, and more ways to expand the game, so I am making this game to ensure this won’t be a problem. The core functionality of this game will be just like other idle/tycoon games, but with much more customisation, allowing the user to add and customise their own planets, creating a more unique and satisfying experience. This game will have a clean, simple user interface similar to other idle games, but with a space theme to reflect the theme of the game. The planet will be in the centre, allowing the player to click, while upgrades, minigames and other options will be on either side. This draws the main focus to the planet, which is the focus of the game. I will research space and galaxy colour pallets to create a vibrant, themed user interface.

For a successful game, I will have to assess the potential stakeholders of this game, including their age and what type of gamer they are. This will allow me to get an idea of the user requirements for the players of this game, so I can better tailor the experience. I will ask the stakeholders questions about certain game mechanics that I have in mind, and whether they would enjoy playing with them. I will also take inspiration from idle games like cookie clicker, which has a similar concept. I will take some features from it, like certain upgrades and the prestige system, whilst customising the game to a space theme with my own features.

1. **Stakeholders:**

The stakeholders for this game are mainly casual gamers, who enjoy relaxed, slow paced games with some challenging features. This is because my game is not very stressful or over-competitive, like first person shooters, so it will attract gamers that enjoy a calm gaming session where they can create something to feel a sense of achievement. The age range for my stakeholders is between 15 and 20, as the sci-fi, space theme is appealing to a young age group. Furthermore, incremental games like this can be played in short sessions, where players can get on for just 15 minutes and make meaningful progress. This is perfect for students who may have limited time due to homework and tests, as they can use this game to take short, relaxing breaks. I will include my stakeholders in my analysis for the user requirements, by distributing a Microsoft Forms survey that will get their opinion on core features, gameplay style and visual design. This will allow me to tailor the game to the specific needs of the stakeholders, making it more personal and engaging.

1. **Other similar products:**

* **Cookie clicker:**
* This game allows the player to click a cookie, which grants them cookies (the in-game currency) which they can use to purchase automated clickers, like grandmas or farms that click the cookie for them, and upgrades that help them progress, like increase cookies per second. As the player progresses, they earn achievements which gains them milk that unlocks unique upgrades. The player can prestige, which resets their progress but gives them a permanent upgrade that helps them earn cookies quicker than before.

|  |  |
| --- | --- |
| **Features I will include in my game** | **Features I won’t inlcude in my game** |
| 1. The automated clickers/workers that the player can purchase and upgrade to idly earn cookies. This is a key feature of an idle game, the player needs some form of automated resource generation, which they can purchase and upgrade to make more powerful. I will adapt this to fit the space theme of the game, for example a space station. | 1. Sugar lumps – These are the secondary, in-game currency in cookie clicker, and they are unlocked after the player has baked a certain amount of cookies. They take between 18 and 24 hours to mature, and can be used to make additional purchases that cookies cannot be used for, like levelling up buildlings (the automated clickers). I like the concept of an additional currency, however I think these don’t give the player enough of a sense of achievement, as they are just collected over time, meaning the player must wait instead of beigng active and earning them themeselves. This makes the game less engaging and active, so I won’t implement this type of additional currency in my game (I will still have one, but it will be gained through active tasks). |
| 2. The factory upgrades, like increased total cookie production or decreased costs. This makes progression in the game more interesting, as the player has more ways to increase their progression. I may adapt these to give the player a choice between multiple upgrades, causing them to progress differently based on their individual choices, which further will further add to the uniqueness of each user’s tycoon. | 2. Heralds – These are people who have donated to the game’s Patreon highest tier, which then gives everyone who plays (whether they have donated or not) a boost. For every one person who donates to the Patreon tier, everyone gets a +1% cookies per second increase. This feature is a nice way to recognise the donations of the game’s supporters, however I won’t be having a Patreon where people donate, so this feature won’t be needed in my game. |
| 3. The prestige mechanic, where the user rests their progress to earn a permanent upgrade to make progressing easier. However, the prestige is quite complicated in cookie clicker, with heavenly chips, an upgrade tree, and angel and demon upgrades. I will likely make it a simple progress reset with a permanent currency multiplier. | 3. A stock market – This is a feature of the bank building in the game, where the player has options of different items that they can buy and sell to make a profit. I think this feature will be too complicated to implement into my game, and is just not necessary. I don’t have much knowledge on how stock markets work, and I cannot assume that the players of my game will either, therefore it may create unnecessary confusion. |

* **Idle world:**
* This game starts with the user clicking a small button to earn energy, which then allows them to develop and colonise a planet, using the energy to purchase upgrades like water and land which generate more energy. The player can also purchase a second world (world 2.0) and mars, to colonise. There is an additional currency, diamonds, which is earned through completing missions, which can be spent on upgrades like 2x energy for 4 minutes.

**Features I will take into my project and why:**

|  |  |
| --- | --- |
| **Features I will include in my game** | **Features I won’t inlcude in my game** |
| 1. The diamond currency, which is earned by completing various challenges. These diamonds can be spent on upgrades which help the player progress further. I like the idea of completing additionally challenges for an additional currency, as it gives the player more to do, making the game more interesting and exciting. These challenges will be the minigames that I mentioned in the overview, which will grant the user this additional currency, which I will change to make my game unique to Idle world. | 1. The random, free rewards in return for watching an advert. Every so often, something like a satellite will pas in front of the planet, and if the player clicks on it, they will get a message offering them a reward (like 2x energy for 2 minutes) if they watch an advert. I like the idea of these free rewards, however having to watch and advert every time can become frustrating, and the player might find them quite annoying. Therefore, I won’t be implementing these type of free rewards in my game. |
| 2. The level mechanic, where the player’s level increases as they progress in the game. I like this feature as it is an easy way to show that the player is progressing in the game. The player can then receive unique rewards once they level up, giving them a sense of achievement and reward. | 2. The upgrades that are purchased by watching adverts. Another way the the game tries to get the player to watch adverts, is through purchasing upgrades. In the upgrades tab, only the first 2 upgrades can be purchased by the in-game currency, whilst the other 10 all require the player to watch an advert to purchase. I think this makes the game boring, as the player is progressing by just watching short videos, rather than spending the currency that they worked to get. Therefore, I am not going to implement this in my game. |
| 3. The offline earnings, where the player still earns energy even when they are not on the game. There is a maximum amount of time that the player can earn offline for, which can be upgraded. There is also a value that determines the offline earnings capacity, which can also be upgraded. I think this is an essential feature in my game, since it is one that can be played in short sessions, meaning it is important that the player can make some progress while they are not on the game, otherwise it would take a lot of short sessions to make significant progress. | 3. Manual level-up. This is a small feature, it is simply where the player has to press “Level up” when they have reached the next level. The game then shows them the reward they earned for levelling up. However, I don’t think this should be a manual process, I think that the level-ups should occur automatically. I think there should still be a pop-up to show the player that they have levelled up, but I don’t see why they should have to press a button to level up, as they are being rewarded for getting to the next level so shouldn’t have to manually press a button. |

* **Adventure capitalist:**
* The player has their own lemonade stand, which they click to earn money. The more they click the stand, the more money it earns them. They can then purchase more stands which earn them even more money. The player can also hire managers, which idly earn money for them. Upgrades are also available which further increase the player’s progress, for example 3x profit for a certain stand. Once the player reaches a certain amount of money, they can go to a new world with different upgrades and stands

|  |  |
| --- | --- |
| **Features I will include in my game** | **Features I won’t include in my game** |
| 1. The multi-purchase feature for stands. This allows the player to purchase more than 1 of a stand at a time, if they have enough money. For any stand, they can either purchase 1, 10, 100 or the max possible amount with the money they have at once. This is also a feature in cookie clicker (except without an option to purchase the maximum amount possible), and it is quite a simple feature that makes purchasing much easier when the player starts to earn a large amount of money and wants to spend it all without clicking on a certain building/stand load of times. | 1. Angel investors – These are an interesting concept, they are attracted as the player earns money, but they can only earn them when they prestige and restart. However, I don’t think these are needed, as it may overcomplicate the prestige process. I will just make some sort of prestige multiplier that increases throughtout the player’s progression, and the more these are when the player prestiges, the better multiplier they will get. I also think that it’s not clear in this game how much money the player needs to start earning these angel investors, so the player will forget about them for a while. |
| 2. The game stats, called “Swag & Stats” in the game. Thisa shows the stats of the player’s tycoon, like their session earnings and lifetime earnings. I will implmenet and adapt this feature into my game, as I believe it is another great way to allow the player to view the progress they have made, to give them that sense of achievement. This is also a feature in cookie clicker. | 2. Mega tickets – These are a third currency, on top of the main cash, and gold. It must be exchanged, taking a huge amount of earth dollars just to get 1. These tickets permanently increase investments on any planet by an extremely large amount (777%), however I don’t think it is needed as there are already 2 currencies, so it might get confusing. Furthermore, the player’s investements could still be increased but just with a purchasable upgrade, rather than a whole extra currency. |

1. **Requirements**

* **Survey:**
* To aid my choice of user requirements, I made a survey with questions about desired gameplay features and importance of aspects like the visual design. I distributed this survey to my friends and members of my college. Here were the results:

**A screenshot of a computer

AI-generated content may be incorrect.Question 1:**

* I made this question to ensure that the answers were coming from users of a specific age range, of 15-20 which is what I decided was the target age range for this game. This confirms that all my responses are from the desired stakeholders, as they were all between 15 and 20. If there were some results from outside the desired age range, and they still showed interest (rated it highly in question

**Question 2:**

**A screenshot of a computer

AI-generated content may be incorrect.**

* This question asked the participants whether they liked idle/incremenhtal games like Cookie Clicker and Adventure Capitalist, which is useful to find out whether the participants enjot this genre of game. 71% of participants said they liked incremental games, confirming a strong interest, meaning I can focus on the core mechanics of these types of games without needing to make it appealing to a wider audience. However, since there were still some participants (29%) who said they didn’t like idle/incremental games, I may have to add some features for a wider audience, that are more engaging rather than simply idle.

**Question 3:**

A graph with colorful bars

AI-generated content may be incorrect.

* This question was designed to get more of an idea on the specific user requirements. I asked the participants to select all the features they thought the game should have, with an option to select all of them. These are all the core features that I have in mind for the game, however I wanted to find out if any needed to be eliminated due to lack of interst, or if any needed to be prioritised due to high interest. The problem with this question is that there were multiple participants who selected all the features, and also selected the “All of the above”, for example:

A screenshot of a computer

AI-generated content may be incorrect.

This creates the problem where some choices appear to be chosen more than they actually have been, if a participant selects it as well as the all of the above option. To solve this problem, I need to clean the data, by removing any individual options that have been selected if the participant has also selected the all of the above option. To do this, I will use the Excel spreadsheet that Microsoft Forms stores all the responses to the survey in. I will use the copy the column for question 3 into a new sheet:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.I will then look at each row, and look for rows that conatin “All of the above” as well as individual options. Where this is the case, I will remove all of the individual options, leaving the “All of the above” option there. The responses are now clean and look like this:

From this, I can calculate how create a table for the correct results for each option:

|  |  |
| --- | --- |
| **Option** | Count |
| Workers that earn money for you | 5 |
| Achievements that apply multipliers | 2 |
| Prestige | 4 |
| Additional challeneges like minigames | 4 |
| An additional currency to purchase powerful effects | 3 |
| Multiple planets in the galaxy that the player can purchase and colonsie | 4 |
| Population gambling | 3 |
| Progress saving | 2 |
| Option to add and customise your own worlds with unique stats | 2 |
| All of the above | 9 |

I can then put this table into excel:

A screenshot of a computer

AI-generated content may be incorrect.

And use a Clusted Bar chart to create a bar chart for the results, similar to the one on Microsoft Forms, but with the accurate, clean results:

A graph with blue and black text

AI-generated content may be incorrect.

This tells me that all of these should be essential features of the game, since all the options were chosen by at least 11 participants, meaning all these features are generally desired.

**Question 4:**

A screenshot of a graph

AI-generated content may be incorrect.

* This question asked the users why they would play my game, allowing me to understand the reasons and motivations for playing it. This then allows me to more accurately tailor the game’s features and visual design to it’s players. The participants were able to select a maximum of 3 choices, and all of them apart from 2 used all 3 of their choices (the other 2 only used 2 of them). The results show that the most common reason that the participants would play this game is to relax and to kill time(as these options combined received 43% of choices), which tells me that I need to create a more calm, cozy visual design, with a focus on relaxing, idle features rather than fast-paced competitive mechanics(I can still include challenging minigames, but potentially make these more of an optional feature that the player doesn’t have to do). I also need to ensure there is always something for the player to do, so they don’t get bored. The results also show that personal achievement and the love of high scores and competition would play a big part in the motivation to play this game, as both of thse options combined received 38% of choices. This tells me that there also neeeds to be a focus on the score that the player is achieveing and what they have earned as a result of it. I need to make the player feel rewarded by getting a high score, potentially giving them unique rewards for reaching a certain score. Additionaly, 18% of participants said that a reason they would play this game is because they enjoy customising and creating things, meaning there must be a focus on the planet they are generating. I must make it clear that they are developing a planet, by visually displaying it’s expansion as they play. This also confirms that the feature where the player can add and customise their own planet with unique stats would be a desired feature for many, as it allows for more creation and customisation.

A diagram with a circle and a circle with a circle with a circle with a circle with a circle with a circle with a circle with a circle with a circle with a circle with a circle with

AI-generated content may be incorrect.**Question 5:**

* This question asked the participants how long they would typically play this type of game in a single session, further allowing me to tailor my game’s features to its players, as I know how long they need to be playing before they see significant progress. There were no responses of less than 15 minutes, meaning the game doesn’t need to be too fast paced and the player shouldn’t be seeing much progress before 15 minutes of playing. Most of the particpiants (64%) said that they would play for between 15 and 30 minutes in a session, and the rest (36%) said they would play for 30+ minutes. This informs me that the game should start to feel rewarding after 15 minutes, where the player can see the progress they have made, ensuring they stay engaged. This can help me design some of the features, like the speed of the idle resource generation.

**Question 6:**

A close-up of a pie chart

AI-generated content may be incorrect.

* This question was designed to find out how idle/active the game should be. A more idle game will involve simple tasks that the player completes (like clicking) to then be able to gain a lot of progress without any input. Whereas, a more active game will invole lots more tasks that the user can complete, like challenges and minigames to help them progress further in the game. Since the majority (65%) of participants said they would like a mix of idle and active gameply, this informs me that the game needs to include lots of interactive features that the user can engage with, whilst also allowing them to gain significant progress without any input.

**Question 7:**

A close-up of a graph

AI-generated content may be incorrect.

* This question asked about the importance of the visual design in the type of game that I’m making, allowing me to decide the priority of how the game looks. Since 86% of participants said the visual design is either somewhat important or extremely important, the UI of this game will be a high priority. This question allows me to ensure I focus on an aesthetic, visually appealing design whilst also focussing on the features and game mechanics.

A close-up of a graph

AI-generated content may be incorrect.**Question 8:**

* Following the question that asked the importance of the visual design, this question asked the participants which visual style they prefer. From the 4 options, there was a split between the most popular. Both Sci-fi and Minialist + Clean received 36% of choices, informing me that the best style for the game would be a combination of both. The sci-fi style will fit well with the space theme of the game, which I can ensure is still clean, aesthetic and modern.

**Question 9:**

A screenshot of a computer

AI-generated content may be incorrect.

* This question asked the participants if they had any suggestions for additional features for the game, which was designed to see if there were any features that a lot of people were asking for, that I had missed, however there were no responses to this so I am sure that the game isn’t missing any must-have features.

**Question 10:**

**A screenshot of a survey

AI-generated content may be incorrect.**

* This question was a quick, rating question where I asked the particpants to rate (on a scale of 1-10) the idea of this game. This was just to check that the particpiants were actually interested in my idea and whether they thought it would make a fun game. Fortunately, I received an average rating of 8.71, and no one rated it below 7, which confirms that this game has potentially to be enjoyable and engaging.
* **User Requirements + Essential Features:**

|  |  |
| --- | --- |
| **Requirement** | **Reason** |
| 1. The player should be able to click a planet, to increase it’s population which earns them money. | This is the core mechanic of the game, as it’s an incremental game so requires a simple action that the user can complete to progress. The population allows them to earn money, which is essential for progress and upgrades. |
| 2. There should be a clean, sci-fi/space-themed GUI with space-themed animations. The GUI should be separated into the main section, with the planet and popuation and money displayed clearly, and then individual sections like challeneges and upgrades. | In the survey, 50% of participants said that the visual design in this kind of game is somewhat important, and 36% said that it is extremely important, therefore the there must be a clean GUI and s to ensure there is an aesthetic visual design. Furthermore, in response to the question that asked the participants what kind of visual style they prefer, there was quite a strong interest in all the styles. However, the most popular styles were Sci-fi and Minimalist/clean, therefore the visual design needs to have a sci-fi aspect (which fits well with the space-theme), and also be clean and sleek. |
| 3. A shop where the player can purchase upgrades, like cash multipliers and worker upgrades. | This is an essential feature of the game, since it is going to have upgrades that need to be purchaseable. A shop is a perfect way to organise all of the upgrades into one area, so it is clear where they all are and easy to access. |
| 4. A multi-purchase feature, where the player can purchase multiple of one upgrade/automated clicker in a single purchase, if they have enough money. | This is a very useful feature, as it allows the user to purcahse a large number of something in one click, without having to click it loads of times (especially if they have a lot of money, which will take a while). It is also extremely easy to implement, so is an essential feature. I am also taking this feature from Adventure Capitalist and Cookie Clicker, as I have benefitted from it in both of them, making purchasing much more convenient. |
| 5. Workers, that the player can purchase, that click the planet for them without their input, increasing the planets population which then increases the players earnings. These workers need to be upgradable, allowing their generation to be increased. | This feature was the most highly requested in the survey (question 3), with every participant sayign they would like this feature in the game, thus it should definitely be implemented. Just to add, in the survery this feature was labelled as “Workers that earn money for you”, however the workers will actually be clicking the planet so earning population for the player, however this then increases their earnings so essentially workers do earn money for the player indirectly. This is a minor error in my survery and I don’t think the results would be any different if the feature was labelled “Workers that earn population for you”, since the key concept is the actual workers themselves”. Furthermore, workers are a core mechanic in incremental games, something that idly generates resources and can be upgraded. |
| 6. A prestige mechanic, that allows the player to reset their progress (keeping some progress like achievements), so they can restart the game again but with multipliers that make progression easier than before. | This feature was highly requested, with 93% survey particpants chosing it as a feature that should be in the game, meaning it should definitely be implemented. Prestiging is also a core mechanic in many games, including incremental games, as it keeps the game fresh, by allowing the player to restart but with progress multipliers. |
| 7. Multiple planets that the player can purchase and colonise. | This was another popular feature, also with 93% of survey participants selecting it, making it an essential feature. This is also essential to keep the game fresh, as it may get boring just colonising a single planet, whereas a galaxy of different planets with different upgrades and effects will make the game much more interesting and engaging. |
| 8. Additonal challenges, like space-themed minigames that the player can complete for a reward. These challenges will earn the player the additional currency that I will implement, that will allow them to purchase unique, limited effects/multipliers. | This was highly requested, with 13 survey participants choosing it as a feature that should be in the game. Furthermore, in response to question 6, 64% of particpants said they would like a mix of idle and active gameplay in the game, meaning I need to have features like this to ensure the game has active mechanics to allow the player iteract with it more rather than just watching idle progress. |
| 9. An additional currency that the user will earn from the challenges/minigames. This currency will allow the player to purchase unique, limited effects/multipliers, for example double money for 2 minutes. | This was a popular feature amongst the participants of the survey, with 12/14 (86% of) survey participants choosing it in question 3, meaning it is in high demand so should definitely be added to the game. Furthermore, the challenges/minigames should have a unique reward to give the player a better sense of achievement, this reward will be this unique, additional currency. It also just makes the game more engaging, with another currency that the player needs to earn. |
| 10. A population gambling feature, where the player can gamble a specific amount of their population, after which they have a chance of either losing it all, or gaining more than they gambled. | This feature was popular amongst the survey participants, as 86% of them selected it as a feature they would like in the game, meaning it should definitely be implemented. Furthermore, it increases player engagement and activity within the game, making it more enaging for them. Again, 64% of paticipants (in response to question 6) said that they would like a mix of idle and active gameplay and 14% said they would like more actige gameplay, thus population gambling ensures there is an active side of the game, allowing the player to better influence the game themselves, rather than them just progressing idly. |
| 11. Achievements that the player reaches after time playing, for example “Clicked 500 times”. These achievements will grant xp, which goes towards the player’s level. | This was quite a highly requested feature, with 11/14 (79% of) survey participants saying they woulld ike the game to have it, meaning it is a desired feature amongst the stakeholders. Furhtermore, this is a common feature amongst most games, not just incremental games, as it gives the player a sense of achievement, as their progress in the game is being recognised and rewarded. In fact, the (joint) third highest reason for why particpants said they would play this game (in response to question 4), with 7 participants choosing it, was for personal achievement. This confirms that the game needs to recognise the progress of players, and reward them with achievements that then give them even more progress, to give them a sense of achievement. |
| 12. An option for the player to add and customise their own planets, with unique stats. This will be available to the player once they have reached and fully colonised all of the game’s planets. The player will be able to give the planet any stats they want, giving them free control of how they progress on it. This will be more of a free roam/sandbox mode that the player unlocks at the end of the game | This feature was also quite a highly requested one, with 79% of participants choosing it as a feature they would like in the game, so I must make sure it is included. Furthermore, in response to question 4, about the motivations for playing this game, 7 participants said that they would play this game because they enjoy customising and creating things. Therefore, this feature is important to increase the customisability of the game, giving the player the option to create their own unique world. |
| 13. Progess saving, where the player’s progress will be saved in the local storage of their browser, allowing them to return to their save whenever they re-open their browser. Furthermore, the player will be able to export their save, which gives them a save code that they need to keep somewhere (like in their notepad), which they can then import into the game on a different device/browser to return to their save. | This was another feature that was requested quite highly, by 79% of participants of the survey, so it should definitely be implemented in the game. Furthermore, progress saving is a key mechanic of many games, as it allows the user to come back to the game at a different time and still have their progress, that they worked hard for. Additionally, 64% of survey particpants said that they would play this kind of game for 15-30 minutes in one session, therefore many players will have to exit the game after this time, and it is important that the game saves their progress so they haven’t wasted their time playing. Players might also switch devices, or browsers, and want to keep their progress, which is why it’s important that they can export their save and then transfer it. |
| 14. A level mechanic, where the player earns xp through upgrades, purchases and general progression. This xp will increase their level, and once they reach the next level they will get a reward and a congratulations screen will be displayed to them. | This is another important feature that shows the user their progression in the game, thus giving them a sense of achievement and making them more likely to continue playing to get an even higher level, to earn more rewards. This is a feature thatis inspired from Idle World, but with the removal of it's manual level-up mechanic and replaced with automatic levelling up. |
| 15. Offline earnings, where the player’s workers carry on clicking while they are offline, giving them money. When the player returns, their population and money will have increased. The amount that their population can increase by and how long it can increase for while they are offline will be 2 different stats that the player can upgrade with money. | Since 64% of survey particpants said that they would play this kind of game for 15-30 minutes in one session, it is very important to ensure that they make progress while they are offline. This is because that they will likely not make significant progress in such short sessions, so allowing them to make some progress while offline will ensure the game is engaging and doesn’t take too much constant effort to progress in. Furthermore, offline earning is a core feature in many games, especially idle/incremental games |

* **Limitations:**
* These are desirable features that I would have wanted to implement, however I won’t or am unable to for specific reasons

|  |  |
| --- | --- |
| **Limitation** | **Reason** |
| Complex Animations | Due to the number of features and mechanics in my game, I will be limiting myself to simple animations. This is because I don’t have much knowledge on web-based animations and I want to focus on how the UI actually looks and the core features. Focussing too much on unique, detailed animations for all the game’s features will take up too much development time, therefore I will ony be implementing basic ones. These will still fit the sci-fi, space-theme, however. |
| Multiple Languages | I am not going to develop any options to change the text language in my game, so it will all be in English. This is because the main language for all games is English, it is the most widely spoken language so I can assume that the majority of my stakeholders will be English speakers. I also don’t need ot make this game accessible to everyone, but instead I want to focus on making it as good as possible for those who can understand it. |
| All planets in galaxy | Since there are 8 planets in the galaxy, I am limiting myself to only developing 4, as it would take too long to develop all 8, with unique stats and designs. Instead, I will just implement Earth, Mars, Jupiter and Saturn, so I can focus on making high quality planets within the time I have. Furthermore, the player will have the option to add their own planets with custom stats, so if they want to add other planets, like Venus, then they can. |

* **Hardware + Software Requirements:**

**Hardware:**

* + This is a game so any personal computer or laptop will work.
  + Mobile phones will be able to access it, however the format may make it unplayable as I won’t be developing the game to work for phones.
  + The device must have an internet connection, as this is a web-based game.
  + The device may need speakers to hear game sound effects and music, however this is not essential.
  + The device requires a mouse, for navigating the GUI and clicking buttons to play.
  + The device will require a keyboard for certain minigames and text input, like when customising new planets.
  + The game is web-based, so won’t take up any space in memory. The local storage will take up space in memory, but this will be a very small amount.

**Software:**

* + A browser is required to play this game, as it is a web-based game.
  + There are no additional software requirements, it is just the browser, which all computers/laptops come with, pre-installed.

1. **Computational methods:**

* **Why this is a computable task:**

1. The problem can be **decomposed,** which is where it is broken down into smaller, more manageable sub-problems which make development easier to understand, reducing the overall time it takes to solve the problem. The reason this problem can be decomposed, is because there are various sections of the game that can be individually identified, and that all integrate to form the overall solution, for example:

* Planet development – The program needs to track the users clicks, increase population, and generate a certain amount of money per second
* Upgrades – The program needs to manage and display upgrade cost and effects, handle the purchasing logic (removing the item price from the player’s money) and the increase of price each time the upgrade is purchased
* Minigames – The program needs to handle the minigames, which are a separate game mechanic from the main game, with different rewards based on wins/loses, or how well the player did in a minigame.

I will use decomposition to split up this problem into sub-problems that I can individually design, develop and test. This reduces the complexity of the problem, avoids overlapping of tasks, allows me to track development much more easily, making it easier and quicker to develop and test, giving me more time to fix bugs and improve the game. At this stage, I am only identifying how the gameplay logic will be decomposed, into sections like planet development and upgrade handling. However, in the design section, I will explain how each of these sections will be further decomposed into Javascript subroutines, with flowcharts and pseudocode where appropriate.

1. The problem can be **abstracted,** meaning unnecessary details from it can be removed to make the problem simpler to solve. This is mainly because it is a space-themed game, meaning there are a number of features I could include, like stars, meteors, satellites with unique physics and animations. This would allow for an extremely engaging and immersive experience, however would significantly increase the development time, and I would likely not have enough time to complete the project. However, with abstraction, I can remove these unnecessary elements from the problem, meaning I only need to develop what is needed in the requirements of the solution. For example, when developing the Earth planet, I don’t need to design every city and landmark on the Earth, as this is not needed to play the game and would take way too much time. Instead, I can just use/design an image of the Earth, which the player can click, ensuring I have enough time to develop the core features of and the gameplay logic. Abstraction makes the game run smoothly, reducing unnecessary resource usage, and reduces the time it takes to develop the program, due to it’s reduced complexity.
2. This problem contains tasks that can run **concurrently,** which is where they overlap. For example, once the player is generating idle income from their planet, they can do other tasks like a minigame, which will overlap with the idle income from their planet, meaning the player can play minigames while still earning money from their planet. This makes the gameplay feel more smooth and interactive, as the player can play minigames, browse the store, or look at their achivements, whilst earning money in the background, thus improving their overall experience. Additionally, concurrency makes more efficient use of the processor, idle (waiting) time by overlapping processes.

1. **Success Crieteria:**

1. The player should be able to click a planet, to increase it’s population

2. The player’s money should increase as the population of the planet increases

3. The player should be able to purchase workers, with money, which click the planet for them to increase it’s population. These workers should be upgradeable, through cash purchase

4. The player should be able to access a shop, with upgrades like worker and cash upgrades that they can purchase with cash. The shop should have a multi-purchase feature, where the player can buy multiple of one upgrade in one click, if they have enough money.

5. The player should be able to prestige once they reach a certain level, allowing them to reset their progress to restart with increased multipliers that make progression easier

6. The player should be able to purchase different planets that they can switch to and colonise.

7. The player should be able to play minigames, where they can earn an additional currency, which should allow othem to purchase unique, limited multipliers.

8. The player should be able to gamble a chosen amount of their population, where they have a change of losing it all or getting back more than they gambled.

9. The player should be able to earn achivements as they progress in the game, for example by clicking the planet a certain amount of times. The player should earn xp each achievement.

10. The player should be able to add their own planets with unique stats, which they can colonise like the default planets. This should only be available once the player has fully colonised all of the default planets.

11. The player’s progress should be saved to the local storage of their browser, and they should be given the opportunity to export their save which should give them a code that they can import on other devices to return to their save

12. The player should have a level, which increases as they gain xp. Xp can be earned through upgrades, minigame wins, successful gambles etc. The player should get a congratulations screen when they level up, and get a reward.

13. The game should update while the player is offline, so that when they return their money and population increases. The amount it increases by and the maximum time it can increase for while they are offline should both be stats that the player can upgrade to earn more offline.

14. The game should have a clean, sci-fi and space-themed GUI with space-themed animations. There should be a main, central section for the planet, population and money stats, with sections on either side for features like the shop and minigames

**Idle Planet – Design**

**- George Kerr -**

1. **Structure Diagram:**

**A group of white rectangular objects

AI-generated content may be incorrect.**

**2. Development Plan:**

**1. GUI Setup:**

**- Description:**

* In this stage, I need to create the HTML and CSS layout for the game, with the central planet and stats and side panels for other features.
* I need to setup the GUI, with all the sections of the game including their buttons, input forms and text displays
* I need to setup a central section, which displays the planet that the player will be clicking, with text underneath to show population and money
* Under this (still in the central section), I need to create the worker section, where the player can buy new workers, upgrade all workers, and see how quickly workers are clicking the planet
* I need to create the left section of the game, which will be dedicated to more active features of the game:
  + I will have an achievements section in the top left of the page, which will display the achievements to complete. For each achievement, the game will display how much xp the player has got out of how much they need to complete the achievement. I will also make a progress bar for each achievement, that visually displays the progress the player has made for each achievement
  + I will make a minigame section underneath, which will have 3 buttons for each 3 minigames that the player can choose from
  + I will make a prestige section, which will have text to display the current prestige multiplier, and the button that the player can press to prestige
  + I will make a population gambling section, with an input form to allow the player to enter the amount of population they would like to gamble. It will also have a win return display that will display the population that the player will earn if they win. I will also make a display that tells the player they have either won or lost.
* I need to create a the right section of the game, which will be dedicated to purchasable items:
  + - I will have an effects section in the top right, which will display all the limited effects that the player can purchase, by clicking, with their gems (the secondary currency), along with the price of each. The player’s gems will be displayed in the top left of the effects section.
    - I will also have a shop section, directly to the right of the planet, which will display all the upgrades which the player can purchase, by clicking, along with the price of each and the number of each upgrade the player currently has. In the top right of the shop section, I will make a multi-purchase button, which allows the player to purchase multiple of the same upgrade at once if they have enough money. The player can click this button, switching it between x1, x10, x100, and MAX (which means the maximum number of upgrades that the player can purchase with the amount of money they have). Under the price of each upgrade, I will make some text that corresponds with the multi-purchase setting selected, and if MAX is selected then this text will display the number of each upgrade that the player can possible purchase with the money they currently have.
    - I will also have a planet section, underneath the shop, which will display the next planets for the player to colonise. I will put a lock symbol next to each planet, apart from the very next one which I will put text next to that displays the level needed to unlock the planet.

**- Why this stage first:**

* Creating the GUI gives the foundation and base layout of the game, allowing me to add and program features into it later. Without creating the GUI first, it would be very hard to test features, as I wouldn’t get a visual representation of how they are working.
* It helps show me where all the features will go, preventing me from needing to constantly redesign the GUI later when I add sections

**- Test plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Is the title section set up correctly |  | Open the game and look at the top middle of the page | There should be a container which displays the game title and the name of the current planet. The save and load icon should also be in the top left of this container |  |
| **2** | Is the central section set up correctly. |  | Open the game and look at the centre of the page | There should be a container, where the planet is displayed at the top, with the money and population display underneath |  |
| **3** | Is the left section set up correctly |  | Open the game and look at the left of the page | There should be a container in the top right which displays the achievements.  Underneath, there should be another container which has three containers within it (the minigame, prestige and population gambling containers) |  |
| **4** | Is the right section set up correctly. |  | Open the game and look at the right of the page | There should be a container in the top right which displays the effects (along with the players gems in the top left of this container). Underneath, there should be a container with two containers within it (the shop container and the planets container) |  |

**2. Planet Clicking and Population Growth:**

**- Description:**

* Here, I need to implement the planet clicking feature, which allows the player to click the planet which increases the planets population
* Each time the player clicks the planet, the population will increase by 1
* I need make the population automatically increase money
* I will start by making the money increase by 0.01 every second for every 1 population.

**- Why this stage here:**

* This is the core mechanic of the game and without it, nothing else would work as the player needs a way of earning the currency
* Since all the other features depend on this feature, getting this feature done at the start will allow me to test all other features

**- Test plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Does the population increase correctly when you click the planet |  | Click the planet and see if the population number goes up | The population number increases by 1 (Before any upgrades) every time you click the planet |  |
| **2** | Does the money increase as the population increases |  | Click the planet to increase the population, and see if the money starts increasing | The money should start automatically increasing by 0.01 every second once the population increases above 0. The amount that the money increases every second should increase as the population increases. |  |

**3. Workers:**

**- Description:**

* In the worker section inside the main section (underneath the planet ) I need to add the option to buy workers that automatically generate population
* I will start by making any workers increase the population by 1 every 5 seconds.
* I will make a display in the worker section, to display the total amount of population that all the workers are generating per second.

**- Why this stage here:**

* Once the manual population growth has been developed, the next stage should be the idle income,
* In order for the workers to be able to automatically increase population, I need to first implement the population and money system, which is why this is after stage 2
* Idle growth is also a core feature of this game since it’s an idle game, so should be developed early on.

**- Test plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Is the player prevented from purchasing workers when they don’t have enough money |  | Press the buy worker button without any money | Nothing should happen; the total workers number should not increase at all |  |
| **2** | Is the player able to purchase a worker |  | Press the buy worker button with enough money and see if the total workers number increases | The total workers number should increase by 1 |  |
| **3** | Does the worker automatically increase the population |  | Buy a worker and check whether the population number is increasing automatically | The population number should increase by 1 every 5 seconds (Before any upgrades) |  |

**4. Shop:**

**- Description:**

* I need to create the shop, which has all the upgrades available for the player to purchase including their price
* I will have different categories of upgrades:
  + - Money and population upgrades
    - Worker upgrades
    - Minigame upgrades
    - Gambling upgrades
    - Achivement upgrades
    - Offline earnings upgrades
* I need program the purchasing system and ensure purchased upgrades actually apply effect
* I need to make sure the costs of upgrades and workers increase as the player buys more of them

**- Why this stage here:**

* Now that the player can earn money, and there are workers that automatically increase their population, they need a shop to buy these workers and spend their money on.
* The player will not be able to progress much without a shop to buy upgrades from

**- Test plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Is the player prevented from buying upgrades when they don’t have enough money |  | Attempt to buy an upgrade by clicking one in the shop with no money | Nothing should happen, upgrade number for that specific upgrade should not increase |  |
| **2** | Is the player able to purchase an upgrade |  | Click an upgrade in the shop with enough money to buy it to see if the upgrade number increases | The upgrade number for the upgrade clicked should increase by 1. |  |
| **3** | Does buying upgrades correctly reduce the player’s money |  | Click an upgrade in the shop with enough money to buy it and observe the money display. | The money should decrease by the price of the upggrade |  |
| **4** | Do the shop upgrades correctly apply their effects |  | Buy any upgrade in the shop | The upgrade should do what it says in the name of the upgrade, for example double money should double the money per second |  |

**5. Save + Load System:**

**- Description:**

* I need to implement the storage of player progress in local storage, I will do this by taking all data that links to the player’s progression and things they have unlocked (which will be stored in an object), and store it in local storage, which are:
* Money
* Population
* Workers purchased
* Upgrades purchased
* Gems
* Level
* Achievements
* Planets unlocked
* I need to also add an export feature for manual saving and loading. I will do this by taking all the progression data (which will be stored in an object), converting it to a json string, using a simple checksum algorithm on it, and returning an export code for the user to copy which contains the original json string and the checksum
* For the import feature, I will use an algorithm that recalculates the checksum using the json string in the export code, compares it to the checksum, and if it matches will import the progression data. If the json string in the export code does not match the checksum, then the game will not allow the player to import the code, telling them that it is invalid.

**- Why this stage here:**

* Now that the player can make meaningful progress, they need a way to save it
* Also, developing this feature will make testing much easier as I won’t have to keep restarting the game to test features that occur later on the game. I can make different saves for different progress levels and test them easily.

**- Test plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Does the player data like money, population and workers save to local storage correctly |  | After getting some money and at least 1 worker, close the browser, re-open it and go onto the game. | The money, population, and worker count should be the exact same as before the game was closed |  |
| **2** | Does the game generate an uninterpretable save code that the player can copy to their clipboard |  | Open the save and load menu and press the export button. Then press the copy to clipboard button. | The game should generate and show a code that cannot be interpreted (appears to be random characters). After pressing the copy to clipboard button, the code should be copied to the clipboard |  |
| **3** | Does the export code correctly import a game save |  | Export a game save with some progress made. Open the game in a different browser/device, then open the save and load  menu and import the save code. | The game should return to exactly how it was on the other browser/device, with all the data (money, population, workers etc.) the same |  |

**6. Prestige:**

**- Description:**

* I need to develop the option for players to reset their progress, and give them a population multiplier that makes progress easier than before
* The prestige multiplier will depend on the player’s population, so the higher their population, the higher the population multiplier will get when they prestige.
* I will use a formula to make the prestige multiplier increase exponentially as the population increases.
* I need to also make prestiging require a certain population, so the player can’t prestige as much as they want, whenever they want. The population that will be required to prestige will increase with each prestige.
* I will also make the game keep some of the players progress, so the reset is not too hard. The player will keep their:
  + - Gems
    - Achievements
    - Level
    - Prestige upgrades

**- Why this stage here:**

* Now that I have implemented the core mechanics and the saving feature, I need to start adding features that extend the life of the game. Therefore, I will implement prestiging here so I can start to make the game into a long term game.

**- Test plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Does the prestige multiplier increase as the population increases |  | Click the planet repeatedly to increase population and check if the population multiplier increases | The population multiplier gradually increases as the population increases |  |
| **2** | Is the player prevented from prestiging until they reach a certain population |  | Attempt to prestige with no population to see if it is allowed | Nothing should happen; the player should not be allowed to prestige |  |
| **3** | Does the game reset the player’s progress after prestiging |  | Prestige with enough money to see if the game resets the player’s money, population, workers and upgrades (not including prestige upgrades) | The game should reset the player’s money, population, worker count and upgrade count to 0, with all upgrade and worker effects removed. |  |
| **4** | Does the player keep their achievements, gems, level and prestige upgrades after they prestige |  | Prestige with enough money to see if the game keeps the player’s achievements, gems, level and prestige upgrades | The player’s achievements and their completion, gems, level and prestige upgrades should remain unchanged |  |

**7. Multiple Planets**

**- Description:**

* I need to add the ability to purchase new planets, allowing the player to colonise them in the same way as the previous planet
* I need to make planets require a certain level to colonise. For the next planet to colonise, I will show the level required to colonise (which will be 50 for the first planet after Earth), and for planets after that I will show a lock symbol until the next planet is unlocked.
* When a player switches to a new planet, the game will look the exact same, except the planet image will be that of the planet they are on, the name of the planet in the title will update to the current planet name, upgrade prices will increase, but minigames will grant higher rewards and effects will be stronger.

**- Why this stage here:**

* Now that I have built the core mechanics, I can develop other planets and just replicate these core mechanics on them, since the mechanics will be the same on all planets, but just will slightly different upgrades and progress.

**- Test plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Is the player prevented from unlocking the next planet if they don’t have the required level |  | Attempt to unlock the next planet with a lower level than the required level, to see if the game allows it | Nothing should happen, the game should not allow the unlocking of the next planet |  |
| **2** | Does the game correctly switch to the next planet when the player selects it with a high enough level |  | Attempt to unlock the next planet with a higher level than the required one, to see if the game switches to it and correctly updates the title | The game should switch to the next planet. The game should look the same, except the planet in the title should switch to that of the new one. |  |
| **3** | Does the game correctly increase upgrade prices, minigame rewards, and effect strength/time after switching to a new planet |  | Take note of the upgrade prices, minigame rewards, and effect strength. Then switch to the next planet and compare them with those of the new planet | The upgrade prices, minigame rewards, and effect strength should have all increased |  |

**8. Minigames:**

**- Description:**

* I need to add simple minigames which the player can play
* I need to create a screen for minigames, which will appear over the main game (as an overlay), allowing the player to play the minigame whilst the main game runs in the background
* I need to create a win and loss screen, which will be displayed to the player depending on whether they win or lose the minigame.
* The win screen will also display the gems and xp that the player earned from the minigame.
* The 3 minigames that I will create are:
  + - **Speed clicker**
      * The player has 10 seconds to click a planet as many times as possible
      * A timer is displayed to show the time running out
      * A counter is displayed to show how many times the player has clicked the planet
      * At the end, the player’s reward is calculated based on the clicks they got.
    - **Memory Game:**
      * A grid of 8 cards are displayed, face down and the player flips 2 of them at a time, trying to match pairs
      * There is a time limit for the game, and if the player matches all the cards withing this time limit, then they win
    - **Planet guesser:**
      * The player is shown all 8 planets in the solar system
      * A random planet is chosen by the computer, and the player has 2 tries to guess it
      * If the player guesses the planet correctly, they win, but if not then they lose
  + I need to also implement minigame cooldowns, so the player can’t just play minigames whenever they want and get loads of rewards. The cooldown will be a timer, the length of which will depend on the difficulty I make all the minigames:
    - Planet guesser (easiest) – 20 minute cooldown
    - Speed clicker (medium) – 10 minute cooldown
    - Memory game (hardest) – 5 minute cooldown

**- Why this stage here:**

* Minigames are an optional side feature that the player does not have to play to make progress
* Now that I have the base game sorted, I can add the extra features like minigames that user has the option to play

**- Test plan:**

**9. Gambling mechanic**

**- Description:**

* I need to make the gambling feature, where players can gamble their population with a chance of losing it or multiplying it
* I will make a display that shows the player, before they actually gamble, how much population they will get back if they win.
* I will make a different sound play, depending on whether the player wins or loses.
* I will make some way to tell them whether they have won the gamble or lost it.
* I need to then make the population actually update when the player wins or loses

**- Why this stage here:**

* This is another smaller, optional feature that I can add once the base game is working, as it requires the population mechanic to be working.

**- Test plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Does the game correctly show how much population the player will win once they enter their gamble |  | Enter an amount of population to gamble in the gambling section and look at the return display | The return display should show a number that is bigger than the population entered. |  |
| **2** | Does the player lose the majority of gambles |  | Gamble 20 times and count how many times were lost | The majority, if not all gambles should be lost |  |
| **3** | Does the player receive the correct amount of population when they win a gamble |  | Gamble a certain amount of population , remembering the population before the gamble, until a win | The population should increase by the amount displayed by the “win return” display. |  |
| **4** | Does the player lose the correct amount of population when they lose a gamble |  | Gamble a certain amount of population, remembering the population before the gamble, until loss. | The population should decrease by the amount that was gambled |  |

**10. Achievements**

**- Description:**

* I need to make achievements that the player reaches once they make certain progress
* I will make each achievement progress bar show the player how much progress they have made in it
* I need to implement the rewards for achievements, like multipliers and gems, which will get better as the player completes more achievements
* When the player completes an achievement, I will make a simple animation, where the achievement progress bar will light up and a sound will play.

**- Why this stage here:**

* Now that the player can make progress through all the features I have added, I now need to make achievements so that progress can be tracked
* Once I add achievements, I can test them straight away by using the earlier features I developed to check the progress tracking

**- Test plan:**

**11. Level system:**

**- Description:**

* + I need to implement a level system, where the player has a level that increases as they earn xp.
  + I will make the player will start at level one, needing 1000 xp to get to the next level.
  + I need to make the amount of xp that the player needs to get to the next level will increase exponentially, making it harder and harder to level up each time
  + The rewards for levelling up will be a certain amount of xp, a percentage of the players population, and a few gems.
  + The rewards for levelling up will increase each time the player levels up.

**- Why this stage here:**

* + At this point, I would have created the xp system where the player can earn xp, as well as an achievement system where the player completes achievements to get more xp. Therefore, it makes sense to add the level system at this stage so the xp the player earns will go towards their level
  + It will also be easy to implement this once I have the xp set up, as I just need to increase a progress bar based on how much xp there is, and create level up screens that show when the player levels up

**- Test plan:**

**12. Offline progression:**

**- Description:**

* I need to implement the mechanic that tracks when the player leaves the game
* I need to calculate the earnings of the player whilst they are offline, by calculating the time difference between when they last left the game, and when they re-opened it.
* Since there will be a max offline earnings upgrade, I need to make the population and money increase, only up until it reaches the maximum value of this upgrade.
* Also, since there will be a max offline earnings time upgrade, I need to make the population and money increase

**- Why this stage here:**

* Since the main game loops is finished at this stage, I can add the offline earnings
* Now that the player can fully play the game, they need to be able to make progress when they are off the game

**- Test plan:**

**13. Custom Planets**

**- Description:**

* I need to develop the planet customisation feature, where the player can add their own planets with unique stats
* I need to make an overlay screen that will appear over the main game, which allows the player to customise their own, unique planet
* The screen will have dropdown menus with different headings, grouping them into the different parts of the planet that hey customise. These dropdown menus will be:
* **Planet Basics**
* **Clicking & Workers**
* **Upgrades & Prestige**
* **Gambling & Minigames**
* **Offline Income & Gems**
* Once the player has chosen all the options, they can create their planet and will be taken back to the main screen, where the main game loop will run with their custom settings

**- Why this stage here:**

* This is an end game feature, which only unlocks once all planets have been colonised, so it should be the final feature
* Without this, the player can still play the majority of the game, just without an extra sandbox feature at the end of the game, meaning it would not be a big problem if I ran out of time for this feature

**- Test plan:**

**14. Animations**

**- Description:**

* I need to add space themed animations and effects, for actions like planet clicking, upgrade purchasing, achievement completion, and gambling.

**- Why this stage here:**

* This is not an actual functionality of the game, it is just to make it look better so should be left to the end once all the essential features are complete.
* If I ran out of time for this feature, it would not be a big problem since the core game would have been developed, it would just lack some extra aesthetic features that do not add any functionality to it.

**- Test plan:**

**3. GUI Design:**

**1. Main Game:**

**A screenshot of a video game

AI-generated content may be incorrect.**

**Title (Top middle):**

This is the title section, displaying the name of the game and the current planet that the player is on. The title is in big text, in the top center, making it clear what game is being played and what planet you are on

**Achievements (Top left):**

This is the achievements section, which displays the achievements that the user can complete, and how much progress they have made so far in completing them. Since this is not (directly) an active gameplay feature or a purchase feature, I have put it in the top left, as it is not something that the player will be constantly checking so can be separate from the gameplay and purchase features. The user can easily check there achievements from time to time by looking to the top left, but it does not get in the way of the main game loop. When the player completes an achievement, there will be a simple animation on the progress bar, it will light up and display text that says "COMPLETE: ", followed by the reward that the player got from that achievement. The progress bar will then go back to normal, showing the next achievement to complete

**Level (Under title):**

This is the level section, which displays the player's level and the amount of xp they are into the current level. This is right above the planet, in the top center of the screen, making it easily visible and stand out, which gives the player a sense of achievement as they see their progress increasing at the top of the screen.

**Minigames (Under achievements):**

This is the minigames section, where the player can select any of the 3 minigames to play, Each button is contained within its own box inside this section, making them easy to differentiate and select the desired one. Game 1, Game 2, and Game 3 are just placeholders and will be replaced with the actual name of the minigame once developed

**Prestige (Under minigames):**

This is prestige section, where the user can prestige and reset all their progress to earn a population multiplier. The population multiplier that they will receive after their prestige is displayed in this section, and underneath there is a button to prestige. The text of the button is in red to make it stand out, creating a sense of warning to ensure the player knows that they will lose their progress

**Population Gambling (Under prestige):**

This is the population gambling section, where the user can enter an amount of their population in the form, and gamble it with a change of multiplying or losing it. Once the user has entered the amount of population they want to gamble, they can see how much they will get back if they win, allowing them to see if a gamble is worth it to them. To gamble, the player will click the gamble button, and will be presented with either "YOU WON" or "YOU LOST" depending on whether they won or lost.  The text colour for the win message is green, as green is associated with winning in this context, and the text colour for the loss message is green as red is associated with losing in this context

**Left section as a whole:**

The entire left section, apart from achievements which are separated and put in the top left, is dedicated to active gameplay features. There are minigames, which the player actively plays, the prestige feature, which the player actively choses to use, and the population gambling feature, which is also actively engaged in by the player. Grouping these active features together on the left of the screen, makes them all easy to access and navigate to. When the player wants to use the more active side of the game, they can navigate to the left of the screen and have it all there for them to see and engage with.

**Main section (Centre):**

This is the main section, where the clickable planet is displayed, which will be a clip art image of either Earth, Mercury, Venus, or Mars depending on what planet the player is on. This section will also contain the planet's population, the player's money, and the money that they are currently earning per second. I have put the population and money right under the planet, which makes it satisfying for the player since they can see how much they're going up as they are clicking. The main section is the biggest section of the game, right in the center, drawing attention to it and making it clear that it is the main part of the game. The planet is the biggest element, as it the core feature of the game loop.

**Workers (Bottom centre):**

This is the worker section, where the player can purchase workers that automatically increase the population of the planet. Although this is a purchase feature, I have not grouped it with the other purchase features on the right section of the game since workers are a special type of purchase that directly links to the planet, so I have put it under the planet, within the main section, keeping it grouped together with the planet. There is a button to purchase a worker, which displays its current price. Under this, there will be a text display that shows how much population all the workers are generating per second, so the player can decide whether they should buy more or they need to upgrade their workers.

**Effects (Top right):**

This is the effect section, which displays all the effects (and their price) that the player can purchase with their gems (the secondary currency). Their gems are displayed right next to the available effects, making it easy to see how much gems they have when looking at effects to buy.

**Shop (Under effects):**

This is the shop section, which displays the upgrades (and their price) that the player can purchase, as well as the number of each upgrade that they have purchased underneath the name in the bottom left. Since there will be lots of upgrade here, There will be a scroll wheel to scroll through them, allowing them all to fit and not be too small or squashed. In the top left, there is the multi-purchase button, allowing the player to purchase multiple of the same upgrade if they have enough money. Clicking the button once will change it to x10, twice to x100 and three times to MAX (Meaning the maximum amount of upgrades that the player can purchase with the amount of money they have, accounting for the price scaling). In the bottom right of each upgrade, there will be a number that shows how much of each upgrade that the the player will be purchasing (which corresponds with the multi-purchase button in the top left of the shop). If this number, and the upgrade name, are in green text then the player has enough money to purchase the upgrade, but if it is in red then they don't. The red and green colours make it obvious to the player whether they can afford an upgrade, due to the connotations of green with money and access and those of red with restriction, meaning they don't constantly have to check their money to see if they can afford upgrades. If the player selects MAX as the multi-purchase setting, the text in the bottom right of the upgrades will display the number of upgrades that the player has enough money for.

**Planets (Under shop):**

This is the planets section, where the player can purchase the next planet to colonise, and create their own custom planet if they have colonised all the planets. The level that they need to unlock the planet is displayed next to the upgrade, ensuring they know what they need to achieve to unlock and giving them a goal to hit, and the text colour for this will be red when their level is too low, and green when it is high enough. This makes it clear and obvious to the player whether they can unlock the planet, due to the connotations of green with access and those of red with restriction

**Right section as a whole:**

The whole right section of the game is dedicated to purchasable items, where the player can buy effects, upgrades and planets. Grouping these features together makes purchases easier, as the player can navigate to the right section of the game to make all their purchases in one go.

**2. Level Up Screen:**

**A screenshot of a game

AI-generated content may be incorrect.**

This is the level up screen, which appears over the main page when the player levels up. It tells them what level they reached, along with how much xp, population and gems they earned. The emojis make it more exciting and lively, which increases the sense of reward from levelling up. When the player levels up, this screen will appear over the main game, with the game still running in the background.

**3. Minigame screens:**

- Minigame screen

**A screenshot of a computer game

AI-generated content may be incorrect.**

This is the minigame screen, where the player will be able to play the minigame they selected. The minigame screen will appear over the main game, with the game running in the background. I have divided this screen into 3 sections. The top section simply displays the name of the current minigame. The middle section displays the actual game being played, and this screen is the biggest out of the 3 as it is the main focus and the player needs to be able to see what they are doing clearly. At the bottom, there is a game stats section, which displays the stats of the game the player is playing, allowing the player to easily look at their points whilst still playing, so they know how well they are doing

- Minigame win screen:

A screenshot of a video game

AI-generated content may be incorrect.

This is the minigame win screen, which will display if the player wins the game. It shows them how many gems and xp they earned. The screen has celebration emojis in the title, to make it look fun and rewarding, along with emojis for the rewards to make them feel more special.

- Minigame loss screen:

A screenshot of a video game

AI-generated content may be incorrect.

This is the minigame loss screen, which will display if the player loses a game. It shows that they earned no gems or xp. The screen has death symbol emojis for the title and cross symbols for the rewards, which makes it clear that this is a loss and they have not earned any rewards.

**Win and loss screens:**

These 2 screens are only used for minigames where the player can either win or lose. If they win, they are displayed the win screen and get the set rewards, whereas if they lose they are displayed the loss screen and get no rewards. For minigames that give a reward that is based of how much score the player got, with no way to lose, the following screen will be used:

- Minigame score screen:

A screenshot of a video game

AI-generated content may be incorrect.

This is the minigame score screen, which displays how much the player scored and the rewards they earned from achieving that score. This is used for minigames where the player doesn't either win or lose, but instead gets a score which determines how much reward they get.

**4. Custom Planet Screen:**

**A screenshot of a computer

AI-generated content may be incorrect.**

This is the custom planet screen, where the player is given a range of options to customise their own, unique planet. I have split this section into a title, and a series of dropdown boxes with headings. Each box has a title relating to a different part of the planet customisation. When expanded, the dropdown boxes will give the user different options about how to customise that specific part of the planet, allowing them to enter values and tick checkboxes. Once the player has finished configuring the planet, they can press the Create Planet button at the button to create their custom planet. Grouping these options into boxes with appropriate headings makes the customisation process much easier for the user, ensuring they don't get lost in all the various settings they can change. The dropdown boxes allow multiple different options to get compressed into a single box, which can be opened and closed. I have also used emojis for the title and the box titles to make this screen more aesthetically pleasing and fun, which makes the experience more exciting.

**Colour design:**

A screenshot of a computer game

AI-generated content may be incorrect.

**Idle Planet – Development**

Stage 1 – GUI:

**Features:**

Title section:

* + Container in the top centre of the page
  + Large tag in the centre of the container, which reads “🌍Idle Planet - Earth”.
  + Small save icon in the top left of the container

Level display:

* + Container underneath the title, shorter and thinner than the title div
  + Text in the centre of the container that reads “Level 1 – 0/1000”.

Central section:

* + Container in the centre of the page, starting just under the level display.
  + Round planet image in the top centre of the container
  + Text underneath the planet reading “Population”
  + More text underneath that reading “Money”
  + Smaller text underneath that reading “Money/s”

Worker section:

* + Container at the bottom of the central section
  + Text at the top of the container reading “👷Workers”
  + Button underneath with the text “Buy Worker: Price”
  + Small text underneath the button reading “Total workers: x”
  + Text underneath that reading “Total worker gen: x population/s”

Left section:

* + **Achievements container:**
    - Container in the top left of the page
    - Text at the top of the container reading “📊Achievements”
    - 3 containers underneath the text (short and wide), each with text inside reading “Achievement 1 – 0/100”
  + Container undearneath the achievements container, with 3 shorter “sub-containers” inside
  + **Minigame container:**
    - 1st sub-container
    - Text at the top reading “🕹️Minigames”
    - 3 buttons, side by side, with text inside reading “Game 1 Play”, “Game 2 Play” and “Game 3 Play”
  + **Prestige container:**
    - 2nd sub-container
    - Text at the top reading “🔁Prestige”
    - Small text underneath reading “Population Multiplier: x1.0”
    - Button underneath with text inside reading “Prestige – x population”
  + **Population Gambling container:**
    - 3rd sub-container
    - Text at the top oreading “🎲Population Gambling”
    - Small input form underneath reading “Enter amount”
    - Text to the right of the input form reading “Win return: ”
    - Solid line underneath that spans across the width of the container
    - Larger text underneath reading “YOU WON/YOU LOST”
    - Button underneath with text inside reading “Gamble”

Right section:

* + **Effects container**
    - Container in the top right of the page
    - Text at the top of the container reading “🪄Effects”
    - 3 buttons, one underneath the other, with text inside reading “Effect”
    - 3 buttons, adjacent to each of the effect buttons, with text inside reading “Price”
    - Container outside of the effects container, in the top left, but still connected to it. Text inside the container reading “Gems: “
  + Container undearneath the effects container, with 2 shorter “sub-containers” inside
  + **Shop container:**
    - 1st sub-container:
    - Small, square container in the top left of the shop container, with text inside reading “x1”
    - Text at the top of the container reading “🛍️Shop”
    - 3 buttons, 1 under the other.
    - Text inside each button reading “Upgrade: Price”
    - Small text in the bottom left of each button reading “0”
    - Small text in the bottom right of each button reading “x1”
    - Scroll bar on the right on the shop section
  + **Planets container:**
    - 2nd sub-container:
    - Text at the top of the container reading “🚀Planets:”
    - 4 buttons, 1 under the other
    - Text in the 4 buttons : “Mercury”, “Venus”, “Mars”, “Custom planet” respectively
    - Button adjacent to the first button with text inside reading “Lvl 50”
    - Buttons adjacent to the other 3 buttons with a lock image inside

Theme + Styling:

* + Space themed colour palette according to GUI design
  + South gradient background from # 220B36 to # 4F2B35
  + # 1C1B29 backgrond for all containers
  + 60% opacity for central container
  + # 3F9FFF text colour for title
  + #7BCFFF title text colour for left containers
  + #C77DFF title text colour for right containers
  + # 3F0FFF title text colour for worker container
  + Red text for Prestige button, Gambling loss message, Worker buy button, Shop upgrades, and Planets the next planet to unlock
  + Grey text for planets (including custom planet) after the next one to unlock
  + # 3F9FFF border for title section
  + # 3F9FFF border for level section
  + # 1D3169 border for central section
  + # 7BCFFF border for achievements container
  + White border for achievement bars
  + # 0B7BDD border minigames, prestige and population gambling containers, including the outer container that they all sit in.
  + # 3F0FFF border for worker section
  + # 9205FF border for effects section
  + White border for each effect and its price
  + # 9205FF border for shop and planets container, including the outer container they sit in
  + All other text white

**Note - In the GUI design, I have an east Gradient from #7BCFFF to #C77DFF for level progress bar, however, I will not implement this in this stage as I am going to add the level progress bar when I actually implement the level system, since is will not go up until the level system is implemented**

**I also have an east gradient from #7BCFFF to #449BFF for achievement progress bar but I will not implement this until I implement the achievement system, for the same reason**

**Development**

**A screenshot of a computer

AI-generated content may be incorrect.Title + Level Section:**

A screen shot of a computer program

AI-generated content may be incorrect.

I have created a div for the top center of the page, which contains 2 divs, one for the title and one for the level section. The reason I have put these 2 sections in the same div, is because I am using a grid and these 2 divs are on different rows, meaning there would be a big gap between them which I don’t want. By putting them in the same div, I can set the grid-row of the title div to 1 / span 2, which makes it span over 2 rows, so the 2 divs are essentially in the same row together which will bring the level section up closer to the title section. For the title text, I have used a h1 with a span inside that reads the name of the planet. This is so that I can change the text of just the span when the planet needs to switch. For the save icon, I have used an image from Free Icons Png ([Save Icon PNG Transparent Background, Free Download #5404 - FreeIconsPNG](https://www.freeiconspng.com/img/5404)). I have put this image inside a button, with the background and border removed so only the icon is visible, but it is still a clickable button. I have set the position of the title div to relative, and the save button to absolute, so I can independantly move the button to the top left of the top centre section without affecting the position of the title.

This is what this looks like on the page:

A close-up of a card

AI-generated content may be incorrect.

**Central + Worker Section:**

A blue and green text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

I have created a div for the central section of the page. I have set the flex-direction of the div to column, so that the elements stack vertically on top of each other, and I have set the justify-content to flex-start so that the elements start at the top of the container (with the planet at the very top). Inside the div I have a button for the clickable planet, with an image inside from Pin Clipart ([Earth Transprent Png Free - Earth Drawing Png Clipart (#1635836) - PinClipart](https://www.pinclipart.com/pindetail/ohxTih_earth-transprent-png-free-earth-drawing-png-clipart/)) for the Earth. I removed the background and border from the button, so only the planet image is visible. I have also created a div inside of the center div, for the worker section. Inside this, I have the title for the section, and also a div for all the elements inside the worker section, including the worker button , the total workers display and the total worker generation display. The reason I have this nested div, is so that I can arrange the elements to stack on top of each other, and control the sizing and gap between them without affecting the title of the div.

This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

**Left Section:**

**Achievements**

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

I have created a div for the achievements section, with 3 divs inside for each achievements. I’ve aligned all the elements in the center and given each achievement a border and the correc text. The achievements section is still in the left section, but it is not inside the “left” div I have created because it is in the top right page, and not embedded inside another container like the next 3 sections. This is what this looks like on the page

A screenshot of a white background

AI-generated content may be incorrect.

**Minigames:**

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

I have created a div for the left section of the game which holds the minigames, prestige, and population gambling sections. Inside this left div, I have created a div for the minigames section, with the title of the section, followed by another div for all the minigame buttons. The reason I have put all the minigame buttons in their own div, is so that I can arrange them side by side, separate from the title. This is what this looks like on the page:

A screenshot of a video game

AI-generated content may be incorrect.

**Prestige:**

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

I have created a div for the prestige section (Also inside the left div), with the text for the title and the population multiplier display. I then have a button for the prestige which contains text showing the required population. This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

**Population Gambling:**

A computer screen shot of text

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

I have created a div for the population gambling section (Also inside the left div), with text for the title at the top. I then have a div for the gamble input, which holds the input form for the gamble and the gamgle button. The reason I have these in a separate div is because I can arrange just these two elements side by side. I then have text underneath this div for the Win Return display. This is outside the gamble input div because it is on the line below. After this, I have a separator div, which I have just used to make a line to divide the section according to my GUI design. The separator is just a very short div with a solid border, which makes it look like a line. Finally, I have the win/loss text at the bottom of the section. This will not be visible when I set up the population gambling, and it will only appear when the player wins or loses (YOU WON will appear if they win and YOU LOST will appear if they lose). This design is slightly different to that of my GUI design, because I thought it made more sense to have the gamble button right next to, on the same line as, the input form, rather than all the way at the bottom of the section. I then thought it would be better to have the win return text display underneath the input form, on its own line, so there is more space do display the number which may be large and take up a lot of horizontal space.

**Right Section:**

**Effects:**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

I have made a div for the effects section. The effects section is still in the right section of the page, but it’s not in the “right” div I have created because it is in the top right of the page and not embedded inside another container like the next 2 sections. Inside the effects div, I havce a div for the gems, as this needs to be a container in the top left of the effects section. I then have title text for the section. After this, I have an effects list div, which holds an effects row div for each effect. The effects row div contains a div for the effect, and then a div for the effect price display. The reason I have separate divs for each row of effects, is so that I can group the effect and price together and then arrange them side by side, then stack each one on top of each other. I do this by setting the flex direction to column in the effects list div, so that the effects and their price will be arranged side by side. Therefore, each effect row div will have the effect and then the price next to it, and then the next div will be underneath the previous one. For the gems div, I have set the position to absolute so that it can stay in the top left of the section without affecting the rest of the content. Otherwise, it would push the rest of the elements in the section down. This is what this looks like on the page:

A screenshot of a video game

AI-generated content may be incorrect.

**Shop:**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

I have created a div for the right side of the page, which holds the shop and planets section. Inside this div, I have created a div for the shop, which starts with another div for the multi-purchase button. I have used the same css code for this button as for the gems display in the effects section (except with a margin-buttom of 215px to push it to the top of the section). This css ensures it can be positioned in the top left of the section without affecting the rest of the elements. After this, I have the title text, followed by another div for the shop list. Similar to the effects section, I then have a shop row div for each upgrade, purchase number and its price, allowing me to put the upgrade and its price side by side. Inside the button for each upgrade, I have a span for the purchase number reads x1, which corresponds to the number in the multi-purchase container. I have set the justify content of each shop item to space-between, which pushes that span to the very right hand side of the button. After each shop row div, I have a span that reads x0, which will increase each time the player purchases the upgrade above. I have also set the overflow-y of the shop list to auto, which creates a scrollo bar for the upgrades, so that when I add more the player will be able to scroll through them all, without them taking up loads of space on the page. This is what this looks like on the page:

A screenshot of a shop

AI-generated content may be incorrect.

**A computer screen shot of text

AI-generated content may be incorrect.Planets:**

**A screenshot of a computer

AI-generated content may be incorrect.Entire page:**

This is what the page looks like as a whole:

* **Cleaning up the CSS:**

I have noticed that many my HTML elements have the same 5 lines of CSS code:

A screenshot of a computer program

AI-generated content may be incorrect.

This is because for all of these elements, I set a border with the same border radius, add padding and a margin of 10px, and center the text. Because of this, I can set these properties for all the elements at once. The classes of the elements that all have these properties set are:

* + left
  + right
  + achievements
  + center
  + effects
  + gems
  + level
  + minigames
  + planets
  + population-gambling
  + prestige
  + shop
  + title
  + planet
  + workers

Therefore, I will delete the css that individually sets the properties for each of these classes, and do it all at once:

A screen shot of a computer program

AI-generated content may be incorrect.

This cleans up the css code a bit, removing duplication of code

A screenshot of a computer program

AI-generated content may be incorrect.

I have created a div for the planets section (also in the right div), which has the title of the section at the top. Like with the previous 2 sections, I have a div for the planets list, and then a planet row div for each row. Inside this, I have a div for the planet, which displays the planet name, and then a div for the planet unlock button. The first planet unlock button reads “Lvl 50”, and then eahc one after has a lock image inside, which I got from Free Icons Png ([Black Lock Icon PNG Transparent Background, Free Download #29059 - FreeIconsPNG](https://www.freeiconspng.com/img/29059)). This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

* **Theme + Styling:**

A screenshot of a computer program

AI-generated content may be incorrect.

I added a linear gradient to the body, from top to bottom which creates a south gradient. I used the colours from the design. This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

The gradient has been correctly applied, however since my game doesn’t fully fit in the visible viewpoint of the page and you need to scroll down to see it all, the gradient resets at the bottom of the page. To fix this, instead of hard coding the height of the body to 908px, I should just set the height of the body to 100%, so that the gradient will fill the entire body rather than just filling the visible viewpoint and then resetting after that:

A screenshot of a computer

AI-generated content may be incorrect.

This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

Now, the gradient applies to the bottom of the page (scrolled down all the way), rather than just the visible viewpoint.

A screen shot of a computer program

AI-generated content may be incorrect.

I then changed the background of all the containers according to the colour from the design, which looks like this on the page:

A screenshot of a computer

AI-generated content may be incorrect.

I then applied all the title text colours according to the design:

A screenshot of a computer code

AI-generated content may be incorrect.

This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

**A screen shot of a computer program

AI-generated content may be incorrect.**

I then changed the text colour of the prestige button, gamble loss message, shop upgrades, worker button, and planets to red. Originally, I had a h3 tag for the gamble outcome messages, reading “YOU WON / YOU LOST”, but at this point I changed it to an h3 tag with 2 spans inside, one that reads “YOU WON” and the other that reads “YOU LOST”, so I could apply different colours to them. I then gave the win message span the class “gamble-win” and the loss message span the class “gamble-loss” (which I apply the red colour to in the screenshot above):



I then made the text of the win message to green using the span id:

Also, in order to make the text of the next planet to unlock red, and the rest of the planets grey, I gave the h3 and unlock text for the Mercury planet the class “next-planet”. I then applied the red text to this and then the grey text to the rest of the planets using the “planet-row” id. When the player unlocks a planet, I will use the JavaScript to change the give the id “next-planet” to the h3 and unlock text of the next planet to unlock:

A screen shot of a computer

AI-generated content may be incorrect.

This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

I am now going to set the correct borders for all the sections. I have already assigned borders to all the containers at this point, but they are just plain black borders and I did this so that I could get an idea of how the GUI is looking before applying the styling. Now I have set the correct colours for all the borders in the css:

A screen shot of a computer program

AI-generated content may be incorrect.

This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

I then added the line “color: white;” in the body of the css, to set all text to white. Since this is at the start of the css, every other time the colour of text is set it will overwrite this. This allows me to set the text colour of specific elements, and any elements I don’t set the colour of will be set to the default of white.

This is what this looks like on the page:

A screenshot of a computer

AI-generated content may be incorrect.

**Testing:**

This is my test plan for stage 1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **Description** | **Test type** | **Test data** | **Expected Result** | **Actual result** |
| **1** | Is the title section set up correctly |  | Open the game and look at the top middle of the page | There should be a container which displays the game title and the name of the current planet. The save icon should also be in the top left of this container | **There is a container that reads “Idle Planet – Earth”, and there is a save icon in the top left. Passed ✅**  ***See test evidence 1*** |
| **2** | Is the level section set up correctly |  | Open the game and look at the top middle of the page, below the title section | There should be a container that displays the current level and xp (out of the xp needed for the next level) | **There is a container underneath the title container that reads “Level 1 – 0/1000”. Passed ✅**  ***See test evidence 2*** |
| **3** | Is the central section set up correctly. |  | Open the game and look at the centre of the page | There should be a container, where the planet is displayed at the top, with the population, money and money/s text display underneath | **There is a container with the planet image at the top, and then a text display for the population, money and money/s. Passed ✅**  ***See test evidence 3*** |
| **4** | Is the worker section set up correctly |  | Open the game and look at the bottom of the central section | Underneath the population and money display, there should be a container for the workers, with a buy worker button and 2 text displays underneath (One for total workers and one for total worker gen) | **At the bottom of the central section, there is a container for the workers, which has a buy worker button, a text display that reads “Total workers: x” and anohter display that reads “Total worker gen: x population/s”. Passed ✅**  ***See test evidence 4*** |
| **5** | Is the left section set up correctly |  | Open the game and look at the left of the page | There should be a container in the top right which displays the achievements.  Underneath, there should be another container which has three containers within it (the minigame, prestige and population gambling containers) | **There is a container in the top left for the achievements. Underneath, there is a container which has 3 containers inside, one for minigames, one for prestige, and one for population gambling. Passed ✅**  ***See test evidence 5*** |
| **6** | Is the right section set up correctly. |  | Open the game and look at the right of the page | There should be a container in the top right which displays the effects (along with the players gems in the top left of this container). Underneath, there should be a container with two containers within it (the shop container and the planets container) | **There is a container in the top right for the effects, with the player’s gems displayed in the top left. Underneath this, there is a container with a container for the shop and the planets inside. Passed ✅**  ***See test evidence 6*** |

**Test evidence:**

**Test evidence 5: There is a left section, with an achievements section in the top left, and then a container which has 3 containers within, one for minigames, one for prestige, and one for population gambling**

**Test evidence 3: In the centre of the page, there is a container with the planet image at the top, with a population, money and money/s display underneath**

**Test evidence 6: There is a right section, with an effects section in the top right, and then a container underneath with 2 containers within, one for the shop and one for planets**

A screenshot of a computer

AI-generated content may be incorrect.

**Test evidence 4: At the bottom of the central section, there is a woker section with a buy worker button, anda total worker and total worker gen display underneath**

**Test evidence 1: At the top middle of the page, there is a title section with the text “Idle Planet – Earth” and a save icon in the top left**

**Test evidence 2: Underneath the title there is a level section, which reads “Level 1 – 0/1000**

# Stage 2 – Planet Clicking and Population Growth

1. **Features:**

* Variables for the money and population stores in a game data dictionary
* Clicking the planet should increase population by 1
* Money should increase by 0.01 every second for every 1 population
* **Data Dictionary:**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data type** | **Purpose** |
| game\_data | Dictionary | To store necessary data about the game, like the player’s money and population. |
| game\_data.population | Key |  |
| game\_data.money | Key |  |
| game\_data.money\_per\_second | Key |  |

* **Pseudocode:**

Game data:

dict game\_data {

population = 0

money = 0

money\_per\_second = 0

}

Planet click:

planet-button Eventlistener(click) {

population += 1

money\_per\_second = population \* 0.01

HTML.population.text = population

HTML.money-per-second = money\_per\_second

}

Money increase:

WHILE true

money += money\_per\_second

HTML.money.text = money

WAIT 1

ENDWHILE

1. **Development:**

A black screen with white text

AI-generated content may be incorrect.

At the bottom of my HTML, I linked my JavaScript file using the script tag

A screen shot of a computer code

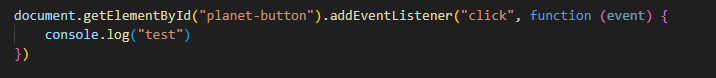
AI-generated content may be incorrect.

In the JavaScript, I created a game\_data dictionary, which holds population, money and money per second (all of which I set to 0). As I develop, I will add more variables to this dictionary for the data the game needs. The reason I have chosen a dictionary data structure is because it groups all the game variables together, so that when I come to implementing the save feature, all the variables are in one place and I can save them all at once. If I were to use individual variables for all the data, it would make the saving process more complicated as I would have to access each variables individually.

A screen shot of a computer

AI-generated content may be incorrect.

In the HTML, I gave the planet button the id “planet-button”, so I can access it in the JavaScript.



In the JavaScript, I added an event listener to the planet button, with the event “click” (which is the event when the button is clicked). In the even handler, I added a console.log statement which logs “test” to the console so I can test to see if it is working.

A screenshot of a computer

AI-generated content may be incorrect.

“test” was logged to the console every time I clicked the planet, so the code is working.

A black screen with white text

AI-generated content may be incorrect.

I then added 3 spans inside each text display under the planet with the ids “population”, “money” and “money-per-second” respectively, so I can access each display in the JavaScript, allowing me to update the player’s population, money and money per second on the page. I have put 0 in each span, as the values all start off at 0.

A screen shot of a computer code

AI-generated content may be incorrect.

I then added an event listener onto the planet button, using its id, with code that handles when the button is pressed. In the event handler, I increase the population by 1 because clicking the planet increases the population by 1. I also increase the money per second by the population times 0.01, as the player earns 0.01 money per population per second. I then update the population and money per second displays in the HTML using the id of the population and money per second span

A computer screen with text

AI-generated content may be incorrect.

I then created a loop that runs every second (1000 milliseconds) which increase the player’s money by the money per second variable, and then updates the money display in the HTML using the id of the money span. I could have just increased the money by population \* 0.01, and not used a money\_per\_second variable, however I did this because I may want some upgrades to increase the amount of money that the player earns per second, in which case I can just modify the money\_per\_second variable to make the player’s money increase by more, making it much easier and less complicated.

I tested this out by running the game and clicking the planet:

A screenshot of a computer

AI-generated content may be incorrect.

All displays are correctly increasing, when I click the planet the population increases by 1, the money/s then increases by 0.01 and the money then increases by 0.01 every second. However, there is an error with he money display as there are many 0s added onto the end of the value which should not be the case (This was also happening with the money/s display but not in the specific image). I realised that this was because of the way JavaScript handles floats, so I needed to round the money and money/s values to just 2 decimal places:





However, this did not fix the issue and stopped the money and money/s increasing at all:

A screenshot of a computer

AI-generated content may be incorrect.

I then realised that this was because when I times the population by 0.01 and then round it to 2 decimal places, this just gets 0. Therefore, instead of rounding I thought of multiplying them by 100 and rounding the result and then dividing them by 100 to get rid of the decimal places. This should work because it will multiply the values, moving the first 2 decimal places (the ones needed) to the right of the decimal point, and then remove the rest of the decimal places by rounding. Then dividing the number by 100 should return it to the correct value with just 2 decimal places





I tested this by running the code:

A screenshot of a computer

AI-generated content may be incorrect.

This completely fixed the issue, now the money and money/s displays don’t display any more than 2 decimal places. I added the comment “//Removes extra, unwanted decimal places” to the end of each of these lines to explain this code as it may not be clear.