Aim of gathering user game play data:

1. Provision of data during gameplay to the user so that they can gauge performance.
2. Analysis of password practices adopted by users of varying levels of security awareness.
3. Analysis of password practices in users provided with the tools to perform better.

Associated Considerations: -

1. Current level of security awareness.
2. Users’ abilities in following/understanding advice given.
3. The tools required to capture the data/information.
4. What data/information will be collected?
5. How will the collected data/information be processed?
6. How will the collected data/information be analysed?

Preliminary Steps: -

1. Evaluation of current level of security awareness
   1. This could come before or after game session. This depends on what we believe would provide more useful data. For this we must consider:
      1. Will evaluating after the game influence a user's approach in answering evaluation questions?
      2. Will evaluation before the game make it more likely that we gather useful data? This may especially be the case if we are giving choice in terms of optional evaluation. (For that and other reasons we should look to make evaluations quick and simple.) Having in game data and a user evaluation would serve more purpose than mere in game data.
      3. Considering the game will be providing the user with useful hints and tips on how to improve during gameplay in the form of techniques on setting better passwords, it may be likely that the information provided serves as a reminder to knowledge once held. This could show an increased level of security awareness on an evaluation placed after the game. We must consider whether or not this would be good or bad for our data.
      4. Evaluating after the game would give opportunity to focus on elements within the game just played. However, it may be better if we separate concerns and not worry about how the user felt the game went as we can gather the actual data from the gameplay itself. However, we may end up with outliers in the data where the user got bored with the game and stopped trying.
   2. As mentioned in point a) ii. above, the form of the evaluation would be best kept simple and easy to complete e.g. by using drop down menus/tick boxes/radio buttons.
      1. Simplified approach will be fast to complete and therefore less likely to put the user off.
      2. Gathering data in such a manner will provide us with an easier job of analysing what is eventually collected. That is not to say that no free format text should be allowed. This could provide useful data too, but not data that we could plan to use prior to the launch of the game/evaluation (but that depends entirely on the questions we pose). If such data is to be collected, profiling will need to take place in order to try and categorize recurring themes so that useful data may be extracted. (If the set up for data extraction involves insertions to tables in a database, adjustments may need to be made to allow for potentially useful evaluation information after initial user testing stage).
   3. Contents of the questions
      1. Questions could gauge how security aware the subject believes themselves to be. (Probably best suited to evaluation occurring before game play)
      2. Other questions could focus on factual information, such as: "How many accounts do you frequent that require a password to access them?" ... "Of those accounts, how many unique passwords do you use?"
      3. As well as these, we could include questions relating to password security issues/facts in order to gauge the actual level of awareness held. (Care must be taken here though as some users may just select an option in an attempt to guess the correct answer. An option of "Don't know" should always be provided for such questions, where the said option has no affect on our analysis of the subject) Data gathered here could be suggestive of correlations but cannot guarantee them.
2. Username or Username & Password to access the game
   1. This would allow for further analysis to be conducted on users who had logged in and played multiple times
      1. This would then bypass any need to complete the evaluation associated with the game if already collected.
      2. Use of a password & username may be required, as we wouldn't want users data being corrupted by others' pretending to be them.
   2. Recognising previous players should potentially change the flow/appearance of the game
      1. There may be no need to deliver the instructions on how to play the game or the initial tips/hints for setting passwords. An optional skip may be required, as we could potentially gather further data in relation to those who decide to read the instructions multiple times.
      2. Providing an optional "skip instructions" implies the need to store NULL values in relevant tables associated with session. We may also need a table column of bit type to represent that they had previously read the instructions and opted out this time around so that the user session data may be excluded from certain analysis pertaining to the ability to set strong passwords based on initial hints/tips received.
      3. Data relevant to the user and previous sessions could be read from the DB and displayed to help them in understanding their historic performance.
   3. Implications
      1. Providing ability to log in means we will need to provide storage of these items.
      2. Storage of sensitive details will then force us to safeguard account details within our DB through encryption.
3. Database to hold data
   1. Initial considerations
      1. Where will this be stored? Will it be on a server that also hosts the game?
      2. What will the tables/relations look like? This will of course depend on the data we decide to collect.
      3. What tools will we need to use? (PostgreSQL; MSSQL Server; MongoDB ....) These will depend entirely on host capabilities: OS, permission to install further applications etc ...
      4. What jobs will be associated with the load/cleanse/process of data? Again, this depends entirely on the data we receive and how best we split it up. Primarily cleansing may only be concerned with free format data received from the user evaluation stage, or with data that fails to conform to a format that we desire. SQL scripts or SSIS type packages may be required for cleanse.
   2. Collecting of game & evaluation data
      1. Variables will be required to hold important information. Arrays, for example, in storage of multiple like items such as passwords, password scores etc... String variable may be required for sessionID, username etc...
      2. Functions in the game will need to be developed to integrate with the DB we choose. These functions will be responsible for writing to the DB and depending on decisions regarding username logins, reading from DB may also be necessary.
      3. Consideration of how these functions operate and when they are called in terms of impact on gameplay will be required. We do not want to be writing to a DB during the game update functions as this may hinder performance. Ideally a bulk write of session data would be required for efficiency's sake. In the case where we also allow reading of previous data from DB, the implication is such that we would at the very minimum be required to write session specific data to the DB following the end of the session itself. This will prevent the possibility of incorrect stats being reported to the user on starting a new game where writes to DB haven't taken place prior to new session start.
      4. Provided the user has a stable Internet connection when playing, data collection will not be an issue if writing at the end of the game. However, if the user was to lose connection, we face potential data loss. In these instances, we may deem it necessary to have considered caching locally, to the host, session files containing user data that is session & IP specific so that we can let the user continue where they left off. This would enhance user experience whilst safeguarding potential data loss.

Data that could be gathered during game play: -

1. SessionID to gather stats based on particular game session.
2. Passwords that the user sets (if they opt in to let us use them that is, but we will need to clear this with ETHICS).
3. If we can't gather their passwords, perhaps hash them to see what that provides in terms of data. (However, hashing could be arbitrary and not necessarily provide anything of use).
4. User name as mentioned above.
5. Was game successfully completed or did they lose? After how many attempts did they win or give up?
6. Closest encounter with an enemy - Perhaps not entirely useful in terms of our analysis aims but could be good to feed back to the user.
7. Tools used; how many times.
8. Memorability - what passwords did they successfully remember?
9. Doors revisited & visited; Passwords set per door.
10. Hints and tips reading time (how long before the user closed hints/tips) - This can only be used to determine potential effects on user performance as users may not read the information even though it appears on screen.
11. Passwords set total.
12. Passwords re-used.
13. Similar passwords used.
14. Entropy of passwords; Lowest and highest (job in DB could use windowing functions to determine these in order to insert in session information).
15. Our score for the passwords in relation to the algorithm to determine alien potential to break through a door.
16. Total length of the session (use date stamps to mark the start and end of sessions).
17. Overall game session score to put them on a leader board.
18. Current level of security awareness needs captured in DB where possible. This will enable us to categorise people into groups: Such as those with a high understanding compared to those with low and then subdivided again into those in the respective groups who seemed to take advice onboard compared to those who did not. Then we can analyse performance on those bases as well. For this to work, we will need to devise a way to assess levels of awareness, then categorise based on the evaluation results. Separate scoring categories could be used, determined by the line of questions asked i.e. did they score high on factual questions vs. their opinion vs. current practices.

How do we analyse the data afterwards: -

1. Insert data/information into relevant staging tables in DB.
2. Run jobs to extract data to the relevant core tables and clean records as necessary.
3. Provide views within the DB that enable users to select appropriate sets of information (refer to point 18 above. The views may only be concerned with results of users across certain awareness levels. Multiple views for each level could be provided.).