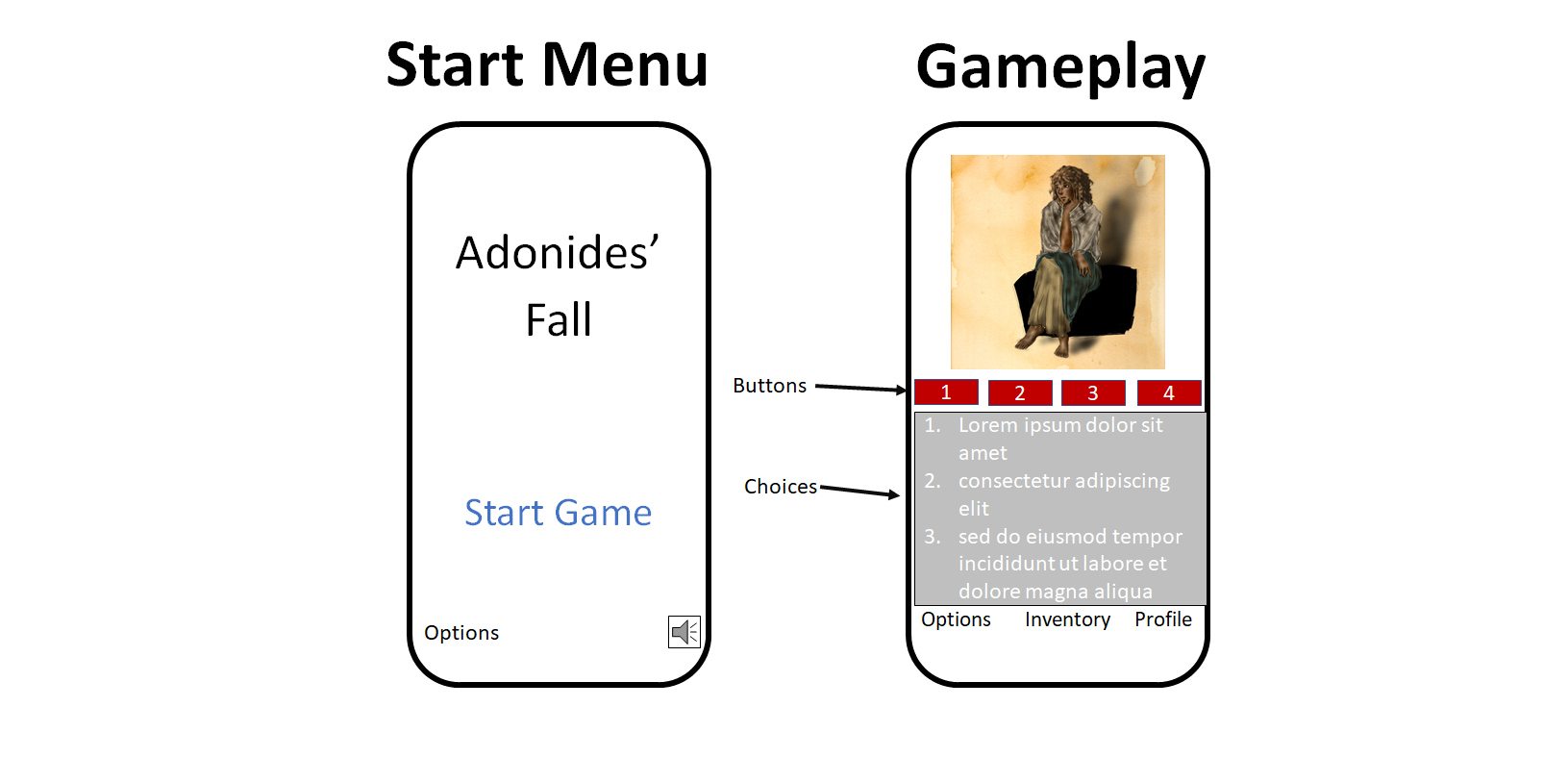
342ConnellyDesignDocV1

* Ciabhan Connelly
* Graded By: Vijeet Sharma
* 02/04/2020

Adonides’ Fall will be a mobile single player roleplaying game for the android phone. The core mechanic will simply be navigating a dialog menu. As the player talks to NPCs and makes choices, the world will update and respond to their actions.



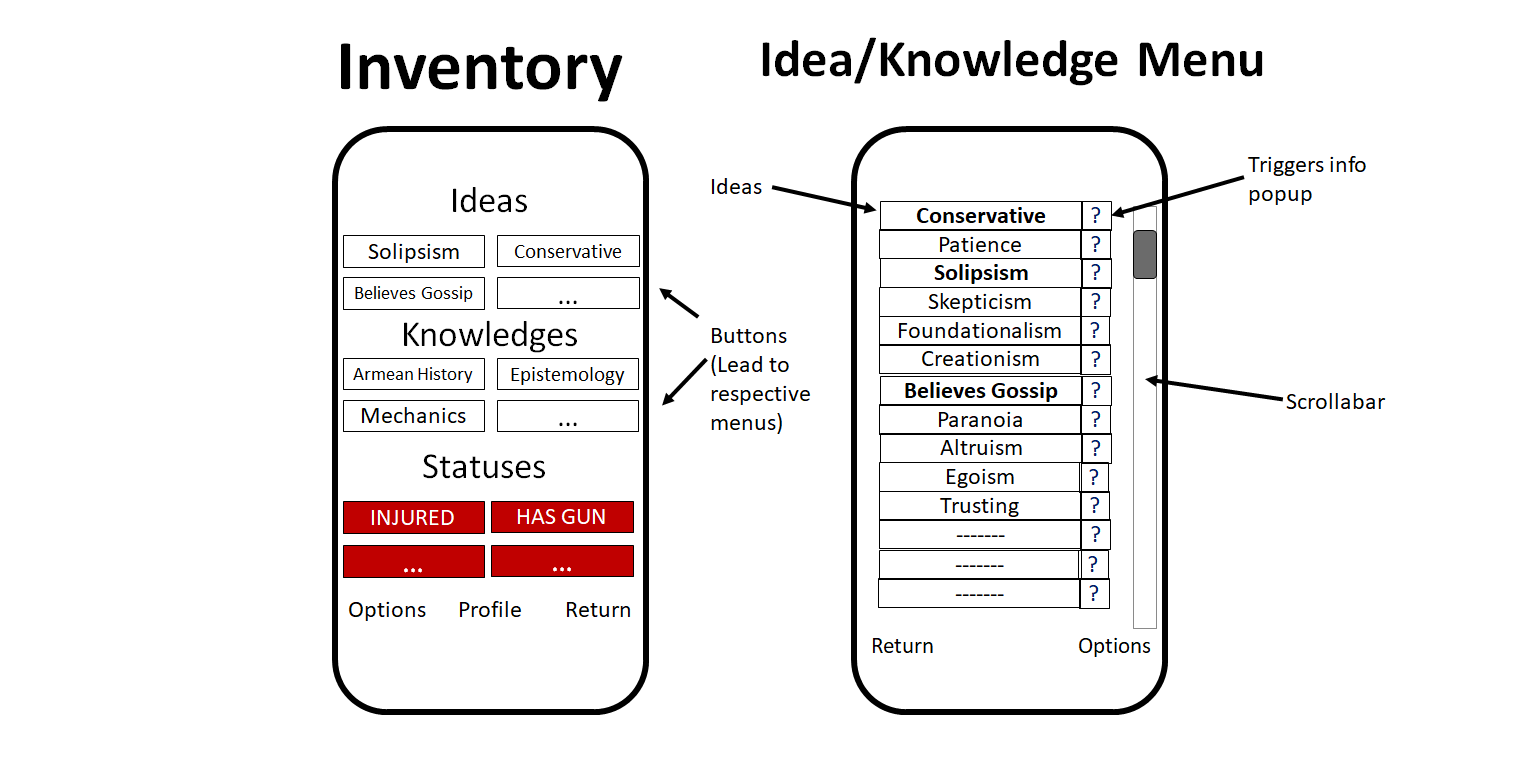
When the player first opens the game, they will be greeted with the screen on the left. The “Start Game” button will either take them to the beginning of the game, or wherever they last left off. The game can be reset from the options menu.

Most of the game will progress as shown in the right frame. The top section will be reserved for images and text that the player can react to by selecting dialog options and other choices from the pictured menu. One planned feature is for certain words in the dialog choices to include drop down menus, which can include nick-names or alternate names the player can use to refer to non-player characters, for example. The pictured buttons represent the player’s mood, which will cycle each time the button is pressed. Mood can change the content of the choices as follows:

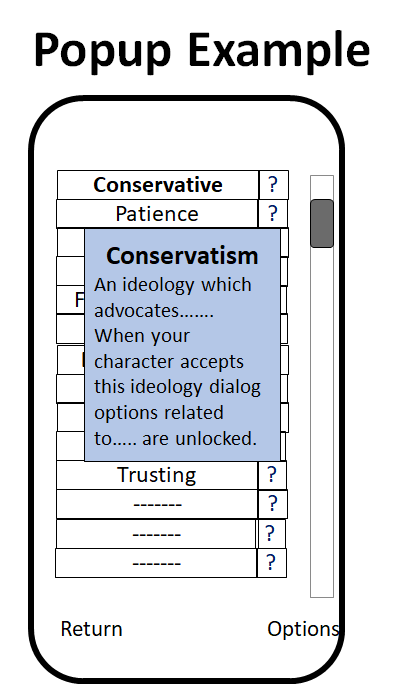
Each of the pictured buttons has three modes.

1. Represents the player’s mood. Sad, happy, or neutral.
   1. In the neutral mood, the text will have no coloration
   2. In the happy mood, player text will be red.
   3. In the sad/melancholy mood, player text will be blue.
2. Represents how intense the player is being
   1. In the neutral mood, text will be slightly transparent
   2. In the low intensity mood, text will be more transparent
   3. In the high intensity mood, text will be completely opaque
3. Represents how authoritative the player is being
   1. In the neutral mood, text appears as normal
   2. *In the unsure/follower mood, text is italic*
   3. **In the authoritative mood, text is bold**
4. This button determines whether or not “lie” dialog options appear
   1. In the normal setting, all dialog options are truthful based on what the player knows
   2. In the middle setting, some dialog options may be misleading, but technically true
   3. In the lie setting outright lies appear. These can be distinguished because r**a**ndom letters won’t reflect the coloration for the player’s current mood

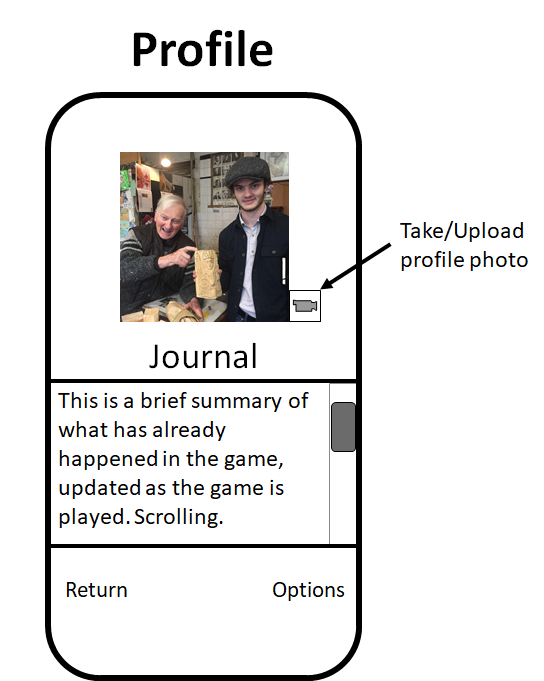
All of these moods can interact, leading to 27 permutations of text formatting, plus additional variations for lies.



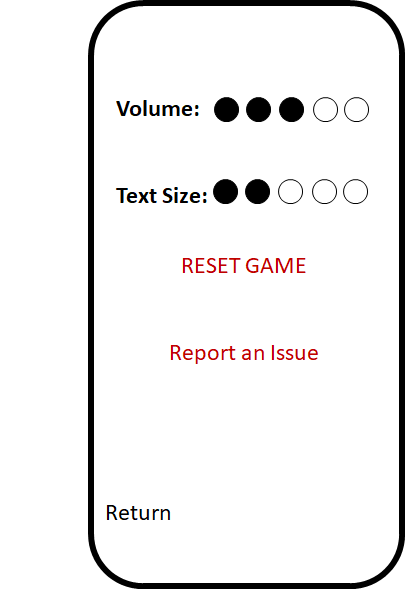
In the inventory, the player can select the knowledges that are on the forefront of their mind, as well as the ideas that they believe in. These will effect their dialog options. New options for ideas and knowledges can be acquired through gameplay. Clicking on one of the boxes in the inventory section will lead to a menu where they can change it out. The currently selected ideas will appear in bold. Statuses just inform the player, and are not interactable.



The info popup gives the player a short blurb about some interactable piece of text. I plan on implementing popups for certain keywords in dialog in addition to the example pictured in the inventory.



The profile screen is where the camera will be implemented. It also includes a journal so the player can catch up on what they’ve been doing after a break.



Finally, the options menu will allow the player to control text size and volume, as well as reset the game so they can start from the beginning. I’m stealing the idea of adding a “report” feature that will send an email in order to fulfil the social media requirement.

**Timeline:**

Most of the features for this app can be implemented as we cover them in class. The thing that will require the most work is the logic on the back end. The logic gates that check what information/choices should be presented to the player will be the most complex part of the app, and therefore take up most of the timeline.

|  |  |
| --- | --- |
| **02/14** | Have basic classes/data structures for the app to access data from. The data itself will be hardcoded in by this deadline, but eventually this will be the SQL database. |
| **02/21** | This is when the “Basic GUI” is due. I will have interactable frames like those shown in the “gameplay” example at the top of this document. These frames will chain together as the players make choices. |
| **02/28** | The basic functionality of the inventory/options menu will be implemented (though I don’t expect to have text size/volume up and running yet). |
| **03/06** | I plan on setting up the SQL database before the deadline since the logic here is so important to work on for the rest of the project. |
| **03/13-03/27** | In between the SQL deadline and the deadline for audio/video I will spend time tuning and improving the logic, and fleshing out the features that use the database. I expect this to be a lot of work, but I do not yet know which elements will require the most time and effort. |
| **04/03** | Add a simple soundtrack to the game, which can be turned up/down and muted through the volume menu. Add the ability for the player to take/upload pictures for the profile menu. |
| **04/20** | Include the report feature to implement email. Add and remove the maps function to show competency with that feature. |
| **05/01** | Finished app. This should be a playable first chapter of the game with all or most of the functionality described here. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Done?** | **Feature** | **Mobile App Notes** | **PA #** |
|  | Design Document:  Target device  Use cases  App description  Wireframes  Android Design | X  X  X  X  X | 1 |
|  | Look and feel | Have concept art, but still a lot to work on | 2 |
|  | Multiple screens | X | 2 |
|  | Screen transitions | X | 2 |
|  | GUI Components:  Labels  Buttons  Drop down box  Checkboxes  Menu | X  X  X  ? Not currently planned. Will experiment  X | 2 |
|  | Structure data | Data hierarchy and SQL will be required for this project. Data management will be a big part of the programming challenge for this app | 3 |
|  | SQL |  | 3 |
|  | Take, save and/or display picture/video | X | 4 |
|  | Record, save and/or play audio | Audio will play in game, no planned feature for the player to record. | 4 |
|  | Collect location | Not planned. Will likely have to implement and remove as I don’t see a way to integrate it into this game. | 5 |
|  | Display location | See Above | 5 |
|  | Send SMS or email | X | 6 |
|  | Final polish of app |  | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Max Points** | **Suggested**  **Score** | **Actual**  **Score** | **Requirement** |
| 5 | 5 |  | Overview: device category |
| 0 | - |  | Overview: testing device |
| 10 | 10 |  | Overview: overview of app |
| 15 | 15 |  | Overview: detailed list of app functions and timeline for completion |
| 10 | 10 |  | Overview: use cases |
| 10 | 10 |  | Wireframes: at least 3 screens (5-10 is typical) |
| 15 | 10 |  | Wireframes: follow Android design patterns and design guidelines  *Didn’t follow spacing guidelines w/these sketches, but how it will generally be able to fit and appear on an android device is clear.* |
| 10 | 10 |  | Wireframes: explanation of navigation/touchable features |
| 10 | 10 |  | The Feature Checklist table with “Mobile App Notes” filled in for your app |
| 5 | 5 |  | Make sure to include your name, the date and the name of your grading AI at the beginning of the document |
| 5 | 5 |  | File naming convention followed |
| 5 | 5 |  | Filled in suggested score and explanations for missing points |
| **100** | **95** |  | **TOTAL** |