#### Developer’s wishlist

##### V1.0

**Focus:** Making it calculate the results

|  |  |  |
| --- | --- | --- |
| **Implemented** | **Feature** | |
| **Storing information about battle participations** | | |
|  | **Using struct to store:**   * The name of the entity **(1)** (including civilization) **(4)**   string unitName   * The armour class **(4)**   bool armorClass[20];   |  |  | | --- | --- | | * Archer [0] * Building [1] * Camel [2] * Castle [3] * Cavalry [4] * Cavalry\_Archer [5] * Eagle\_Warrior [6] * Gunpowder\_Unit [7] * Infantry [8] * Monk [9] * Ram [10] | * Ship [11] * Siege\_Weapon [12] * Spearman [13] * Standard\_Building [14] * Stone\_Defence [15] * Turtle\_Ship [16] * Unique\_Unit [17] * Wall\_&\_Gate [18] * War\_Elephant [19]. |  * The attack bonuses vs amour classes **(5)** * The health **(7)**   int unitHealth;   * The ranged damage (RD) value **(8)**   int rangedDamage;   * The standard damage (SD) value **(9)**   int standardDamage;   * The Age pre-requisite of the card **(10)**   int unitAge;   * The point value **(12)**. Sum of resource cost. What’s awarded for killing the entity   int pointValue;   * The garrison value, which appears on some buildings in case it’s ever relevant   int garrisonValue;   * The quantity of the entity   int unitQuantity;  **Example “Crossbowman” card**    Using struct as I can store multiple data types per entity and return this entity. | |
|  | Having a blank entity, which can be used to reset the values of a selection | |
|  | Having two entities that represent what players 1 and 2 are putting into battle | |
| **Applying the effects of modifiers (attack bonuses, event cards, and technologies) as well as the quantity of units** | | |
|  | Modifying the values of player 1’s selection and player 2’s selection based on modifiers  Approach works because two targets cannot attack one simultaneously | |
| **Calculating the outcome of different round of combat** | | |
|  | Calculating the outcome of an archer round of combat (ranged entities may attack and may retreat). The round is negated if fighting cavalry | |
|  | Calculating the outcome of the standard two rounds of combat (can retreat after 1 round) | |
|  | Calculating the outcome of a monk round of combat | |
|  | Calculating the outcome of a bombardment round of combat | |
| **Getting information about each player’s “play state”** | | |
|  | **Reading info from .csv files:**   * Each player’s entities * The quantity of each player’s entities * Each player’s technologies * Whether these technologies are in play   Not using std::cin for this as there’s too much that would need to be entered and it inserts odd symbols for capitals and underscores https://cdn.discordapp.com/attachments/442575858096668672/442694350489518085/unknown.png | |
|  | **Validating the input**   * Converting names to uppercase format. Seeing if the name entered matches one of the accepted names of entities (in uppercase). * Making sure the user fills out all the fields * Making sure that there are no spaces as I’m using the spaces to split up the fields. * Making sure that the quantity of participating entities is > 0 * And < 2 for buildings * And < 6 for all other entities | |
|  | Using the info from the .csv files to search for the corresponding thing and filling in the rest of the details so no user input is needed for that   |  | | --- | | **Example** | | if(entityName == “Archer\_(Saracen)”) then{  currentSelection = {“Archer\_(Saracen)”,1,entityQuantity,6,4,5,2,  true,false,false,false,false,false,false,false,false,false,false,false,false,false,false,false,false,false,false}; // One of the player’s entered Archer\_(Saracen)  } | | |
| **Playing sound effects from Age of Empires 2 (in moderation, don’t want it to be obnoxious)**  https://docs.google.com/spreadsheets/d/1bczdFQksnbLnjI5zAkw-mSpb9MnnxxEkHDiz1PftIHw/edit#gid=123661276 | | |
|  | | Having AoE II sound effects for the UI |
|  | | Having AoE II sound effects for RNG elements   * Successful “conversion attempt” * Successful “healing attempt” |
|  | | Having AoE II sound effects for significant events   * Destruction of a wonder |
|  | | Having an option to enable and disable SFX |

##### V2.0

**Focus:** Converting it from a terminal application into a cross-platform GUI application

**Goal:** Making it easier to use for users and because of that, calculating the results will take even less time

**Constraints:**

* Screen space. Could perhaps have the calculated result,

**When released:** Package into an installer

**Wireframe:**

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| **Implemented** | **Feature** |
| **Adding essential functions** | |
|  | Exit button |
|  | About button |
|  | View user guide button |
|  | View developer guide button |
| **Sending information about the player’s *“play state”* from the GUI to the files so that the backend can read it.** | |
|  | QListWidget for the entity (unit, building) names |
|  | QLineEdit for the quantities of entities |
|  | QLineEdit for the quantities of monks |
|  | QLineEdit for filtering the entity names |
|  | Limiting the quantity of entities to 1 - 5 |
|  | Limiting the quantity of monks to 0 - 5 |
|  | CheckedListBox in QT for events & technologies <https://www.walletfox.com/course/qtcheckablelist.php> |
| **Getting user input** | |
|  | Supplying variables with answers the user provides. Was using std::cin for this. Perhaps we’d need to use a popup if we’re using QT framework |
| **Displaying output** | |
|  | Writing std::cout statements to the GUI |
|  | Making it so it doesn’t output the same result twice |
|  | Making it so it clears the existing output when clicking the “Calculate results” button again |
|  | Making it so the output text is coloured, perhaps using HTML |
|  | |
|  |  |
|  |  |
|  |  |
| **Working out to represent a few challenging event cards in code** | |
|  | Unimplemented event card X |
|  | Unimplemented event card Y |
|  | Unimplemented event card Z |

##### V3.0

**Focus:** Connecting a SQL database to a C++ program

**Goal:** Making it easier to use for designers that expand upon the Age of Empires board game

|  |  |
| --- | --- |
| **Implemented** | **Feature** |
| **Having data about the entities (units, buildings) stored in a SQL database instead of hard coded into the program** | |
|  | Units table |
|  | Buildings table |
| **Supplying the program with this data** | |
|  | Perhaps running a SQL query in C++ and storing the result of this as a variable |
| **Having a developer window** | |
|  | Add new entities |
|  | Delete existing entities |
|  | Modify the entity values |
|  | View the entity values |