



Software Development UI and Evaluation plan

B087928

March 4, 2016

1 Introduction

We shall report on the development and evaluation of a prototype User Interface (UI) for a deck building card game that we are improving.

The code currently has a very basic command line interface that is only accessible to a narrow variety of users and still unclear and cumbersome to developers and people who understand the rules.

We seek to design a usable and accessible UI. We must consider who we are aiming at, what information we wish to give to game players and if we have developed the right way of interacting. This may sometimes only be discovered once we have a completed the prototype and so we shall have a plan for evaluating the UI prototype.

It is expected that the reader of this report has an understanding of the rules of the game however we develop the prototype with the hope that it will grant full user freedom and it will be difficult to make actions 'by mistake' or without knowledge of what they're doing.

2 UI Design Ideas

I believe a Graphical User Interface (or GUI) is most appropriate for this game, as it will allow us to simulate a real deck building card game and shall appeal to players as a simulation of real life making what's happening in the game easy to understand if we understand the rules of the game. In this way we can then navigate through the game using a 'point and click' interface. Sound is not a necessary component to a deck building card game however background music may add to the imagination of the game.

When playing the game it is easy to notice that in most cases there is no point in playing only some of the cards, especially when the cards would add to the user's attack. Thus it would be a good idea to include an alert message that not all cards have been played if the user tries to end her turn without doing so. This will help increase user control and freedom as ending turn is a large commitment and the user would benefit from been given a second chance to fulfil important tasks. We intend to make the UI that gives allows the user to more ready access to relevant information. When we wish to buy a new card we'd like to know what we already have in our deck. This could be figured out anyway from the knowledge that we begin with 8 serfs and 2 squires in our deck and the memory of what we bought in previous rounds but it will help decision making if this information was readily available. It is obviously against the rules for a player to see his deck and know what cards shall come out next so we should limit this information to simply 'Deck contains: Serfs x8; Squires x2'. We can see this in figure FIGURE!.

3 Prototype

In this section we shall present a control flow of the program as well as a full prototype of the design we have outlined above. We shall conclude this section with a walkthrough of the game showing all the accessible features of the design.

3.1 Control Flow

The control flow illustrates how the user will use the application. It will show where, in a broad sense, each action shall lead the player. It shows what will follow each game action and in what order.

In figure 1 we can see the flow chart for the deck building card game. Once we have indicated that we would like to play the game the computer asks what style opponent would we like (Aggressive or Acquisitive). Now we reach the main interface of our program (see figure FIGURE!!!). From this section the user may play several or all of their cards; buy cards to add to their deck; attack the opponent or end their turn. Only 'End Turn' shall progress the game flow. Selecting 'End Game' prematurely may cause the player to miss the opportunity to do important this therefore the user will be asked to confirm if this option is selected. If the user attacks and is fortunate enough to reduce the computers health to zero then the game shall also progress to the conclusion. There is no warning here since winning is, or at least should be, the user's aim.

When the user confirms that they wish to end their turn the computer gets to take its turn. It shall play cards, buy cards and attack the user. The user shall be notified if she takes damage and especially if her health goes to zero as then the game would have ended.



Figure 1: Control Flow of the deck building card game.

3.2 Prototype

We shall now present the prototype of the two main screens that make up the user interface.

The GUI shall resemble a playing where a player may see all her cards and easily make a decision. We have two interfaces because I believe when we are purchasing cards we'll only take into account how much money we have and what's already in our deck.

Note that do not get to see what cards the computer has; its attack value or its money. The only information we are allowed about the computer is its remaining health which can be seen as part of the interface in figure 2.

In our user interface we use a point-and-click implementation. Consider figure 2. If we click on one of the face-up cards in front of the player it will move out to the active cards area. It dies not ask for confirmation however one may move them back from the active area to the hand by clicking on them in the active area if a mistake is made. When a card is moved from the hand to the active area, or vice versa the money and attack value in the top left corner of the screen is adjusted accordingly.

Pressing 'Play All' shall move all cards from Hand to the Active area. Attack will reduce the Computer's Health by the current attack value and set the attack value back to zero.

The discard pile and Player Deck won't be interactive as the application will automatically discard cards and draw more cards at the appropriate times. However these are here to make the user interface more realistic.



Figure 2: Mockup: This screen forms the main user interface for the game.

When the user selects 'End Turn' she will be prompted to give confirmation. This window will include a reminder that unspent money shall be set to zero as will unused attack.

If 'Buy Card' is selected the user shall be brought to the screen seen in figure 3. The user may then purchase a card by selecting one of the available cards or return to the previous screen by clicking 'End Buy'. The user's deck is shown in this screen, this should help us to decide which cards we'd like to have in our deck and estimate what are the chances of getting it in our next hand.



Figure 3: Mockup: Purchase interface. The user will be taken here when they click on buy.

A background audio soundtrack may add effect to the game as well as ensuring the player that the program has frozen or broken down.

3.3 Walkthrough

We shall now play through a typical game illustrating the choices at each step.

If we consider our control flow (figure 1) we begin by selecting an opponent type. This will be selected via a simple point and click on the following window



Figure 4: Mockup: First screen we encounter in the user interface.

Clicking on one of these options now brings us to the main screen (figure 2). We click 'Play All' to move all the cards in our hand to the active area. We played four Serfs and one Squire so our money value changed to 4 and our attack value changed to 1. We now click on 'End Turn' and we are reminded that if we confirm that we'd like to end turn then we shall lose our money and attack value (see figure 5). Therefore we click cancel and instead press 'Attack!'. This reduces the computer health by 1 and reduces our attack value to 0.

We now decide that we want to spend our money so we click on the 'Buy' button. This brings us to the purchase screen as seen in figure 3. We see that available to buy is a Knight; Thief; Assassin; Tailor and Caravan. We decide to buy an Assassin which reduces our money to 0 and adds Assassin to our deck. We can now see that Assassin is part of our deck by looking at our deck in the bottom of the purchase screen (figure 3). Note that we do not know where in the deck the Assassin will be and when it will appear but we have a reasonable idea what cards we should get.



Figure 5: Mockup: This screen forms the main user interface for the game.

Now that we have played all the cards in our hand; attacked and spent all our money we can happily end turn. The computer now takes its go. We must know what the computer buys (this adds a deep level to the strategy where we may try to keep track of what cards the computer has in its deck!) as well as what damage it does to us this turn. We can see this from the Computer's move screen, as seen in figure 6.



Figure 6: Mockup: This screen is displayed as the computer makes its move.

We now return to the main screen (figure 2) and our next turn commences. The game continues in a similar fashion until either the player all or the computer loses all their health.

4 Design Evaluation

In this section we attempt to justify our design choices. A Graphical User Interface (GUI) was chosen so that we could get the best match between the system and the real world. Players of a deck building card game may be used to playing with real cards, so being able to display information on the cards to the user is believed to be the best, most understandable interface for the majority of users.

When we play a game we've never played before we become experimental and may be unsure what the best order to do things in is. Luckily in our game it doesn't really matter whether we attack or buy cards first. What is important is that we don't end turn before we have attacked or still have purchases that we would like to make. Thus by having to confirm ending turn- and being reminded that we will lose our attack and money value- the user is given more freedom to experiment different options without making an inadvisable move. Being able to move cards between our hand and the active area and seeing our money and/or attack values going up/down as we do will allow a user quickly get user to the way the money and attack values work.

I believe that it is very important to know what cards are already in our deck before we purchase new ones since sometimes it is even advisable to waste money by pressing end turn rather than buy another card we have too many of. Since this is an important aspect of our choice of purchase we can understand the setup of the purchase screen as seen in figure 3.

It is always important for the player to know her own and the computer's health. This could be a large factor in deciding whether to make a more acquisitive or aggressive purchase. Therefore it is important to have both of these values visible from the main screen as seen in figure 2. Health is not displayed on the purchase screen as the user may easily toggle between the Player's Turn screen and Purchase screen and therefore it would be a waste of space on the purchase screen. When the computer is attacking us we'd like to see what our health is after it attacks and not only how much damage it has inflicted. We do not have the computers health here too though because it might be confusing.

Purchasing can be potentially misleading as a user may think that a purchase goes straight to their hand. This is why it is extra important to only have relevant quantities in this window such as money (attack value clearly isn't relevant and excluding it reminds the user that these bought cards aren't coming straight into play).

Finally, when we hover over a card it becomes enlarged and displays its Cost; Attack and Money values. This helps the screen be less cluttered and the user can discover more information about a card if desired. An example of what an enlarged card looks like can be seen in figure 6.

5 User Evaluation Plan

We have presented a prototype and given a walkthrough to illustrate exactly how the game flow works. We have furthermore justfied our design choices from a developer's point of view. A truly appropriate design, however, may be different from the perspective of the user. This is why, in this section we seek to layout a plan as to get and analyse feedback from the user in order to improve the UI. We shall collect feedback on several different components of the UI.

5.1 Visibility

We want the GUI to look aesthetically pleasing, for it to look like a fun card game that the user can be excited to play. We also hope that the game is very visibly understandable, that the user can see the information they want at a glance.

We may be able to determine if we've included enough clearly visible information and options by asking the user if they able to do everything they wanted to do and if all the information they needed on each screen was readily available. We may also directly ask users what they would change about the interface. Once we've received feedback we can incorporate it to help us improve visibility and then get further feedback on the improved UI. Many iterations of this process of feedback and improvement should give us a good UI layout.

5.2 User Freedom

As was discussed previously, it is important that the player is free to experiment with the controls without the risk of doing an inadvisable move accidentally. This was the main purpose of asking the player to confirm that they wish to end turn.

We may ask users if they were frequently making actions accidentally, if so; which actions? We could then cross reference feedback to see what other actions should require confirmation in order to be executed.

5.3 Interaction

Finally we must consider the communication between the application and the user. Here we seek to determine if the buttons suit the user's needs.

We could determine if the interaction was good by asking the player if actions were intuitive. For example, is it easy to get the information they need on a card by hovering over it? Do they find it easy to get information about cards that aren't in their hand or active area since this would involve toggling to the purchase screen? Are they happy for their hand to be drawn automatically? Or would they prefer to draw the hand themselves by say clicking on their deck at the beginning of their turn?

6 Conclusions

We have designed a user interface by what we, the developers, believe to be most usable and aesthetically pleasing. It is obvious however that our understanding of the game and the program is different to that of potential users. Therefore our prediction of a good UI may be very different very different from that of our prospective users. Furthermore, players with the same level of game/program knowledge/familiarity may still have different needs, requirements and preferences. Also, what seems, objectively, to be a good idea may cause problems in practical use. So we may conclude that evaluation and user feedback is a vital part of developing a user interface.

It is important to develop a UI prototype with an open mind since it is likely we shall need to change our design many times when we encounter problems and issues from user feedback or wish to make improvements and implement user's suggestions.

We believe that the UI prototype presented here is simple to understand, visually pleasing and interacts well with users and is hence a perfect starting point for further development and the iterative, circular process of development and feedback.