

# Guidelines for Visual Scale Design: An Analysis of Minecraft

Manivanna Thevathasan

June 10, 2013

## 1 Introduction

Over the past few decades, many video game devices have been introduced utilizing a variety of screen sizes and input modalities. Due to ongoing research in Human Computer Interaction, user interfaces for these games have been created specifically for each of these devices. However, it is becoming more apparent that video games will seek operate across different screen sizes. It is our belief that the design of user interfaces for games with consideration for the variety of screen sizes is an under-researched area of Human Computer Interaction. Thus, we seek to introduce guidelines for Visual Scale Design.

Visual Scale Design is the design of user interfaces with consideration for the properties of the underlying device's screen. We seek to provide a set of guidelines for the optimal development of user interfaces for video games with consideration for the visual scale. The result of which would allow a particular game to run across different screen sizes with the resulting experience being relatively the same. In order to develop these guidelines, we have chosen to analyze the game Minecraft as it is a current example of a game that operates across different screen sizes effectively. We will seek to provide rational explanations, supported by research in Human Computer Interaction and Cognitive Psychology, for the visual scale differences that exist between its different forms. These explanations will help form the resulting guidelines for Visual Scale Design.

The rest of the paper is divided into four sections. The first section of the paper will contain an introduction to Minecraft comprising of an explanation of its game mechanics and a presentation of its User Interface. The second section will contain our observations of the various differences that exist in the game across the different platforms. The third section will consist of a discussion on the design decisions made by the developers to create these differences. Finally, we will present our guidelines for visual scale design based upon the analysis and rationalization presented in the prior sections.

### 1.1 Platforms and Versions Considered

Minecraft exists on a few different platforms and, at the time of writing, there are plans to bring the game to more platforms. For the purposes of this paper, only the following three platforms and their respective versions will be considered:

Name	Platform	Version
Minecraft	PC	1.4.7
Minecraft: Pocket Edition	iOS and Android devices	0.6.1 (Alpha)
Minecraft: Xbox 360 Edition	Xbox 360	1.3.1

Minecraft was originally developed for the PC and as such that version will be considered to be providing the baseline “*Minecraft Experience*”. All other platforms simply strive to provide

an experience that is as close as possible to this baseline “*Minecraft Experience*”. Comparisons will not consider individual platform performance capabilities (ex. Graphical power, CPU processing power, etc.) or player modifications to game content/mechanics/UI.

Each platform is independently receiving constant updates and thus the game’s content on each platform will differ. We therefore do not consider the game’s content differences between platforms. Only the user interface differences for content that already exists on the mentioned platforms will be considered.

Note: Henceforth, “Minecraft: Pocket Edition” may be referred to as the “mobile version” and “Minecraft: Xbox 360 edition” may be referred to as the “console version”.

**TO DO: Add section explaining Fitt’s/ Cog Psych stuff?**

## 2 Minecraft

Minecraft is an open world sandbox style game involving the player collecting, destroying and using blocks to further the goals of their chosen play style. Minecraft doesn’t have a specific end goal. Instead, it provides the player with two modes of play, so as to offer some structure, but largely allows the player a form of non-directed play. This allows the player to play the game in a manner that is enjoyable to them. Minecraft can be played in one of two modes; survival or creative. In survival mode, the player must explore the game world, search for resources, craft items and fight enemy monsters to ultimately survive in the game world to achieve their desired goals. In the creative mode, players are provided with infinite resources and an open world free of any enemies. Typically, creative mode is used for players who wish to create elaborate in-game structures. Minecraft was originally developed for the PC but saw adaptations for the mobile and console platforms as well. As of this writing, the PC version remains the first platform to receive new game content. The other platforms receive the same or modified versions of the same content after they have been correctly adapted for that specific platform’s use.

In order to familiarize the reader with Minecraft, we shall provide an overview of content relevant to our analysis.

### 2.1 Heads Up Display

Figures 1, 2, 3 present the Heads Up Display, (HUD), as it appears in the PC, Console and Mobile versions respectively. The HUD is the key layer of the game’s UI that communicates vital gameplay-related information to the player. As such, adapting the HUD to the various screen sizes, will be essential to convey the same quantity and quality of information. The content of the HUD consists of a few elements that either provide static or dynamic information. Static elements include the toolbar, crosshair (PC and console) or the controls text (console only). These elements present information that doesn’t change often. Dynamic elements include the health, armor, experience and hunger bar. These elements present information that changes quite often.

It is quite obvious to observe that the HUD contains a different amount of content across the different versions of the game. This is due to the fact that there is a differing amount of game content within the different versions of the game. Thus, the UI reflects this difference.

#### 2.1.1 Mobile specific changes

The toolbar in the mobile version contains 6 - 7 item slots but this toolbar is modified to be different than its PC and Console counterparts. Depending on the screen size of the mobile

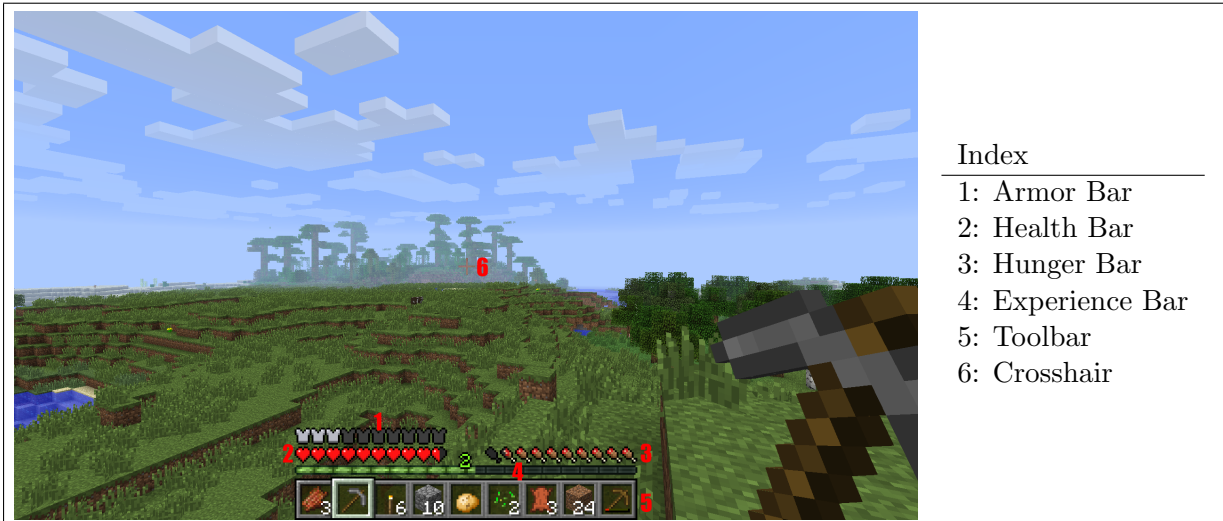


Figure 1: HUD from the PC version of Minecraft



Figure 2: HUD from the console version of Minecraft

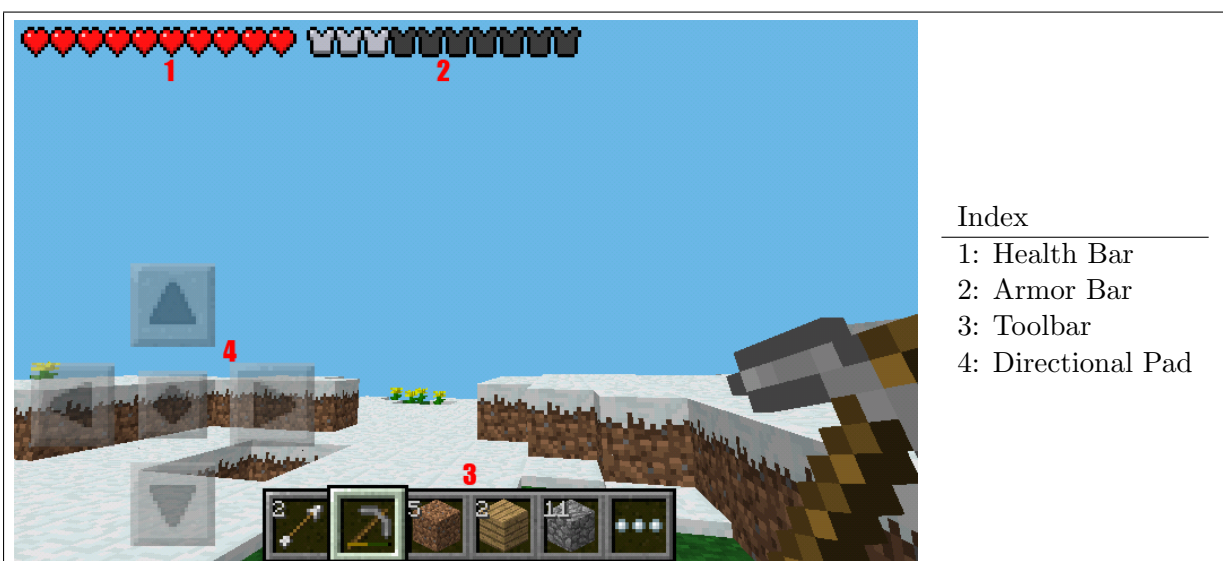


Figure 3: HUD from the mobile version of Minecraft

device, the toolbar will have a varying number of item slots; 6 - 7. However, due to the limited number of input commands, the last item slot is reserved for a button to open up the player's inventory; + 1. Thus, we refer to the mobile version's number of item slots in the toolbar as "5 - 6 + 1".

## 2.2 Input Controls

Minecraft possesses different input controls for each of the three platforms that it exists on . For PC, it works with a keyboard and mouse setup. For the Xbox 360, input is provided via the Xbox 360 controller. For mobile platforms, input is provided via touch screen input. In each of these cases, the UI elements as well as the gameplay elements were adapted to take into account the different ways a player could be providing input to the game.

### 2.2.1 Mobile specific changes

Repeated in-game actions, such as mining for blocks, are performed by holding the corresponding input command on the given input modality for a certain amount of time. In the mobile version, this means tapping on the desired target and holding one's finger on the screen for the required amount of time. The side effect of this input command is that the player's hand blocks the screen, considering that the screen size of mobile devices are smaller. As a result, in the mobile version, feedback to repeated actions is provided via the use of a circular progress bar. This progress bar indicates to the player the length of time they need to hold their finger down until the desired result appears.



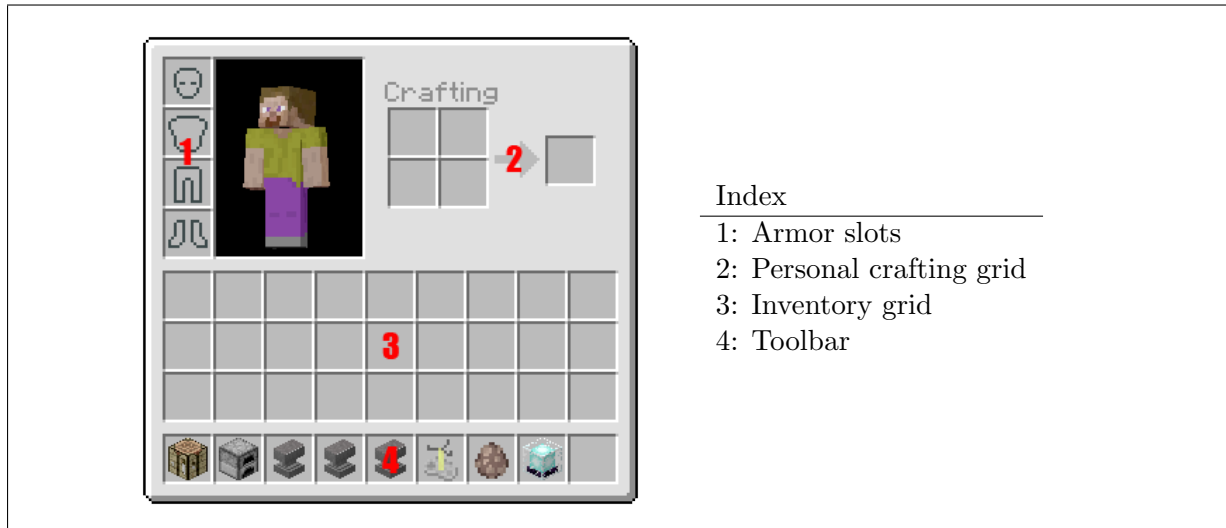
**Figure 4:** Circular Progress Bar (Mobile)

## 2.3 Game Mechanics

While there are a large number of game mechanics that exist within Minecraft, unfortunately the majority of them exist in the PC version exclusively. A subset of those mechanics exist within the Console version and an even smaller subset of that exists within the mobile version. Thus, we seek to analyze the user interfaces of only the most important game mechanics that exist between all three versions of the game. This results in a look at the inventory and crafting windows mechanics and their respective windows within Minecraft.

## 2.4 Inventory

The inventory window enables the player to access additional gameplay mechanics such as the use of armor and crafting, as well as presents the player with a look at their current inventory of items. It essentially facilitates the player's access, use and management of a greater number of items in the game. The inventory contains the items that the player isn't currently using in their toolbar, but may use later on if the situations calls for it. In addition, any new item obtained will be automatically inserted into the first available slot in the inventory if there are no free slots in the toolbar. Figures 5, 6, 7 present the inventory window as it appears in the PC, Console and Mobile versions respectively.



**Figure 5:** Inventory window from the PC version of Minecraft



**Figure 6:** Inventory window from the Console version of Minecraft



**Figure 7:** Inventory window from the Mobile version of Minecraft

### 2.4.1 PC specific changes

The inventory window is optimized for mouse input. The player can select items from the inventory grid and “drag and drop” them into any of the other slots in the window. Note however, that the armor slots are only for pieces of armor and any other item cannot be place there. Other tasks such as personal crafting is embedded within this window as it requires access to the player’s inventory.

### 2.4.2 Console specific changes

The inventory window is similar to the PC version with the exception of the personal crafting grid being removed. Personal crafting is performed in a different manner, in the Console version, and it doesn’t require access to the player’s inventory, so it is absent from the inventory window. Access to the inventory and personal crafting grid are provided via two separate buttons on the Xbox 360 controller.

### 2.4.3 Mobile specific changes

The inventory window is optimized for touch screen input. Items are presented in a grid but contained in a scrolling list. The size of each of slot in the inventory has been increased. Tabs have been used to separate different tasks into separate windows accessible from the inventory.

## 2.5 Crafting

Crafting is a gameplay mechanic, central to the entire concept of Minecraft. Players obtain various items that can be used as crafting materials in order to create new items in the game. An example would be chopping down trees to obtain wood and then using that wood to craft wooden planks. These wooden planks could then be used to craft wooden items such as wooden pickaxes, swords, beds, torches, etc. The possibilities for combinations of crafting items are numerous. Crafting in Minecraft is performed in one of two ways; either personally crafting the item or using an item called the crafting table. Both methods of crafting in the game follow the same principles of the crafting game mechanic; they only differ in the number and types of items that they can each create. Personal crafting allows a 2x2 grid whilst the use of a crafting table extends this grid to 3x3. This extension to the crafting grid allows the player to craft better items because they require more crafting materials.



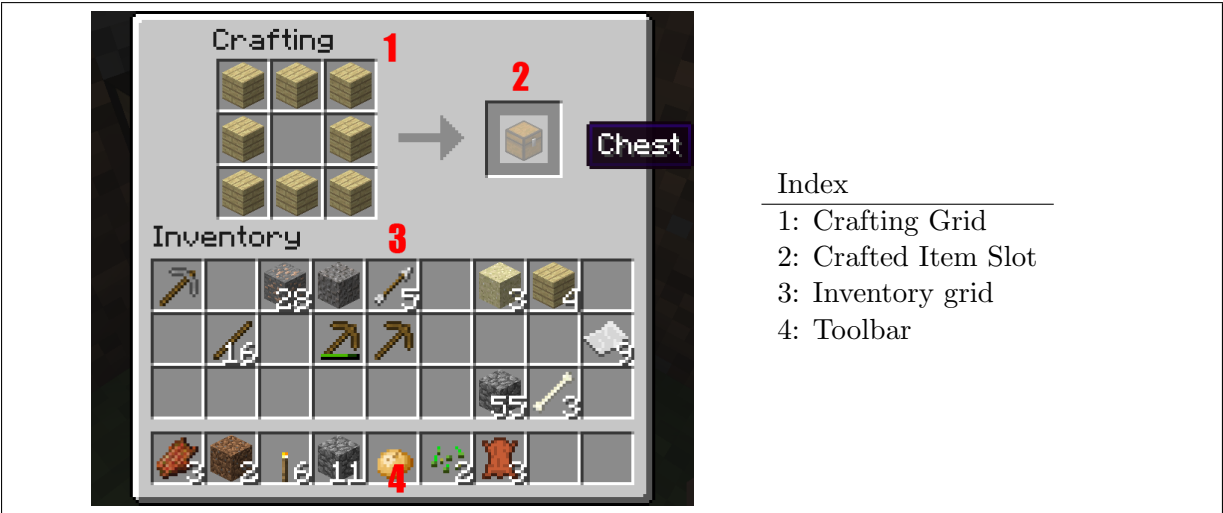


Figure 8: Crafting window from the PC version of Minecraft

- Index
- 
- 1: Crafting Grid
  - 2: Crafted Item Slot
  - 3: Inventory grid
  - 4: Toolbar



Figure 9: Crafting window from the Mobile version of Minecraft

- Index
- 
- 1: Tabs
  - 2: Scrolling List
  - 3: Craft Button



Figure 10: Crafting window from the Console version of Minecraft

- Index
- 
- 1: Tabs
  - 2: List of Items
  - 3: Crafting Grid
  - 4: Inventory Grid

### 2.5.1 PC specific changes

Crafting is performed by placing items of a particular type in a particular order within the crafting grid. Players can select an item from their inventory or toolbar and place them within either of the two grids. If they have placed the correct items in the correct order, then the resulting item will appear in the crafted item slot for the player to then select and place wherever they wish. If multiples of crafting materials are presented in the crafting grid, then the player can press the *shift* key when selecting the resultant item to automatically craft as many possible resulting items based on the number of crafting materials presented.

### 2.5.2 Console specific changes

Crafting in the console version, is a sort of hybrid approach between the approaches taken in the PC and mobile versions of the game. Elements of the PC version’s approach to Minecraft can be seen with the use of crafting grids and inventory presentation. However, influences from the mobile version’s approach are seen as well with the use of tabs to separate items into specific sections and the use of lists to select particular items to craft. Items are crafted by navigating to a particular tab, selecting an item from the list and if the required materials are in the player’s possession, then the item can be crafted. When an item is selected, the required crafting materials automatically appear in the crafting grid with their correct placement order.

### 2.5.3 Mobile specific changes

The mobile version does not possess any crafting grids. Instead crafting is performed via the use of the Minecraft Advanced Touch Technology Interface System, (MATTIS). MATTIS presents the player with a list of all possible items that the player can craft, that is organized into four sections separated with the use of tabs. The four sections include Blocks, Tools, Food & Armor and Decoration. Within each section, a scrolling list contains all possible section-specific items that can be crafted by the player. Items appear with either white or “grayed-out” text indicating whether the player can or can not craft the items, respectively, based on the current crafting materials in their inventory. The player can select an item in the list, which then updates the craft button on the right hand side of the window to show the required crafting materials. Pressing this button, when in possession of the required materials will then craft the specified item.

## 3 Differences

The table below presents a summarized look at the differences between the different versions of Minecraft.



Heads Up Display			
Comparisons	PC	Console	Mobile
Health Bar	Yes	Yes	Yes
Armor Bar	Yes	Yes	Yes
Hunger Bar	Yes	Yes	No
Experience Bar	Yes	Yes	No
Crosshair	Yes	Yes	No
Controls Text	No	Yes	No
D-pad	No	No	Yes
Toolbar	Yes	Yes	Yes
Toolbar Item Slots	9	9	5 - 6 + 1
Location of Elements	Bottom center	Bottom	Edges of screen
Input Controls			
Input	Mouse & Keyboard	Xbox 360 controller	Touchscreen
Input Context	No	No	Yes
Input Feedback	Subtle	Subtle	Explicit
Game Mechanics - Inventory Window			
Inventory Grid	Yes	Yes	Yes
Armor Slots	Yes	Yes	Yes - Separate Tab
Personal Crafting	Yes	No	Yes - Separate Tab
Toolbar	Yes	Yes	Yes
Item Management	Manual	?	Automatic
Game Mechanics - Crafting Window			
Crafting System	Manual	Hybrid	MATTIS
Crafting Grid	Yes	Yes*	No
Access to Inventory	Yes	Yes*	No
Use of Tabs	No	Yes	Yes
Multiple Items	Yes	No	No

\*Not Interactive

## 4 Analysis

As the reader can observe, there are several differences between the different versions of Minecraft. This section will seek to provide rational reasons for these differences and specifically attribute them to either changes in input modality or changes in screen size.

### 4.1 Due to Input Modality

Some of the differences that exist between the different versions of Minecraft are attributed to the changes in Input Modality. Input Modality isn't directly related to the issue of screen size in Visual Scale Design. However, since screen sizes span multiple devices with different input modalities, it definitely affects the design decisions. Design decisions that were made due to the change in Input Modality can be seen with the differences in the Location of HUD Elements, Input Context, Item Management, and Crafting Systems.

#### 4.1.1 Location of HUD Elements

The location of UI elements in the HUD remain fairly consistent between the PC and Console versions. The only real difference being that the HUD elements were moved slightly higher and

control texts were added to the HUD. At this time, no rational reason can be provided for this change.

In the case of the Mobile version, the HUD is drastically different. The Health and Armor bars are moved to the top left edge of the screen. We believe the reason for this is attributed to the side effects of the input modality. The player interacts with the game using their left hand on the overlaid directional pad and their right hand to interact with objects in the game world. While doing this, they may block the HUD elements that are located on the bottom sections of the screen. The health and armor bars are particularly important HUD elements since they provide dynamic information. Since the player would like to view this information frequently without much hindrance, the developers moved these two elements to the top left edge of the screen. Here, the player's hands would not block the screen and thus would allow the player constant access to this information.

#### **4.1.2 Input Context**

The PC and Xbox 360 platforms can map all possible in-game actions to specific buttons on their respective input modalities. Therefore, the context of the button press is not relevant to perform in-game actions. Simply press the corresponding button to perform the desired action.

In the case of the mobile version, the context of input commands matter. Input is provided via the mobile device's touch screen, so there are effectively only two input commands: either tapping on the screen or tapping and holding. As a result, the amount of in-game actions outnumber the number of input commands. To get around this limitation, the context of a given input command is necessary. For example, navigation is performed via the use of the directional pad that is provided as part of the HUD. In addition, interaction with the game world is performed via screen taps on the other parts of the screen. Thus, to perform the numerous actions that exist in the game; the context in which the player is currently interacting with the game and the use of one of these commands are both necessary.

#### **4.1.3 Item Management**

Manual Item management, that is to say the ability to manually place items into any slot within the inventory window or the toolbar, is possible in both the PC and Console versions of the game.

It is however not possible within the mobile version of the game. The input command of selecting an item and dragging it to its intended space does not exist with the given input modality. Instead players are left to work around this limitation. We believe that a design decision was made to exclude manual item management since the time it would take to manually perform item management with the given input modality would be too great to be accepted by the player. Add to this reasoning the fact that this action could be repeated numerous times and it becomes clear that the efficiency of the player's actions were paramount.

#### **4.1.4 Crafting Systems**

The crafting game mechanic is the most significant example of differences due to input modality across the different platforms. The differences can be observed in the crafting interface for each respective platform and were all made due to the importance of player efficiency.

Originally designed for the PC version, crafting is performed manually with the player moving their crosshair item to item, selecting and placing the required crafting materials in a specific configuration on the crafting grid. This manual placement of crafting materials reflected

the use of the mouse. Selection time with the mouse was fast, thus this was the acceptable standard.

With the console version, the time to select and place items on a crafting grid whilst using a game controller became too slow. Thus a new system was designed which utilized tabs and lists to allow the controller to rapidly scroll through and select items to craft. No movement of a crosshair was needed, and thus efficiency was preserved.

The idea to utilize tabs and lists may have come from the mobile version, since it was released before the console version. It was clear that limited input commands would prevent any ideas of efficient manual crafting. Instead, a design decision was made to use a new crafting system: MATTIS. With the use of tabs and lists, the player can simply select an item to craft. This allows the experience of crafting to be quick and efficient. The design decision to introduce MATTIS is also influenced by screen size.

## 4.2 Due to Screen Size

The remaining differences that exist between the different version of Minecraft can be attributed to changes in screen size. With the use of our knowledge from HCI and Cognitive Psychology, we can provide rational reasons for some of the differences that exist. **TODO:** In addition, we can also show that some of the design decisions can be improved. Design decisions that were made as a result of changes in screen size can be seen with the differences in the Toolbar Item Slots, Inventory Window, Crafting Window.

### 4.2.1 Toolbar Item Slots

Typically, the screen size on the PC and Console versions are sufficiently large enough to ensure that there is enough space for a full toolbar with 9 items slots. The reduction in items slots in the mobile version is due to the goal of efficiency and is directly related to the screen size. To ensure efficiency, changes were made to the toolbar to ensure adherence to Fitt's Law. First of all, the size of the individual slots in the toolbar were increased to improve selection time of the individual slots. The result of this size increase combined with a smaller screen size results in a toolbar with less slots, as it still needs to fit on the screen. **TO DO: (Insert explanation about why the toolbar has to be at the bottom. Cog Psych)** The resulting toolbar only contains 5 - 6 slots, (depending on the size of the mobile device), plus the modified slot which opens the inventory.

Another design decision we can observe is the inclusion of the modified slot to open the inventory as part of the toolbar. We believe that due to the limited number of input commands combined with the fact that mobile devices possess varying input standards; the most adaptable solution was to embed the functionality of inventory access within the UI. Since semantically the player's inventory and the player's toolbar are related, the button was added as the final slot in the toolbar. Thus, "5 - 6 + 1" item slots exist in the toolbar in the mobile version whilst 9 item slots exist in the PC and Console version.

### 4.2.2 Inventory Window

The differences in the inventory window between the PC and Console version are quite minute. The mobile version sees a much more drastic change. Given the smaller screen size, to fit a comparable number of item slots in the inventory grid, the grid was contained in a scrolling list. In addition, the size of each slot in the grid was increased. Both of these design decisions can be attributed to Fitt's Law and the goal to preserve the player's efficiency despite the smaller amount of screen space available.

Another design decision of importance is the focus on one separate task at a time within the inventory window. To ensure optimal use of the available screen space, each task within the inventory window is separated with the use of tabs. This helps provide a sense of separation of concerns, which actually follows guidelines put in place for optimal UI development. **TO DO: Provide guidelines**

### 4.2.3 Crafting Window

#### **TO DO: Section needs a review/better picture**

The crafting window in the Console version has differences from the PC version due to the different input modality. However, upon observing the crafting window, it becomes apparent that screen size contributed to those differences as well. The crafting window presents the player with additional information such as a view to their inventory and a crafting grid. Yet, these two views are not interactive nor are they required, (in this format), for the crafting mechanic in the Console version. At this time, there is no rational reason for why these additional pieces of information are provided when they serve no direct purpose to the player. However, we suspect it is related to the screen size, since the additional space allows the inclusion of these non essential views. Comparing the Console version's crafting window with the Mobile version's, further reiterates this screen size suspicion. The Mobile version also uses tabs and lists to present the player with items to craft, however no direct view of the inventory is shown and required crafting materials are shown in the most basic way possible.



**Figure 11:** Required components of Crafting window from all versions of Minecraft

Figure 11, shows a comparison of the crafting window across the different platforms focusing on the required information that the player needs to craft. In the PC version, the entire window is needed since access to the inventory is required. The player receives feedback to know whether they can or cannot craft an item, if the resulting item appears or does not appear, respectively, in the Crafted item slot. In the Console version, only the left portion of the screen is needed as access to the inventory is not required and feedback is provided right there. The right portion of the screen seems irrelevant. In the Mobile version, the crafting button shows the required items, the number of required items the player has and has the functionality to craft the item all in one. It is with this comparison that it becomes apparent that as screen size reduces, an emphasis on functionality is key and as screen size increases, an emphasis on detail is key.

### 4.3 Due to both Input Modality and Screen size

There are differences that exist between the different versions of Minecraft that can be attributed to **both** changes in input modality and screen size. Input Feedback is a great example of this. In the Mobile version, the use of explicit feedback via the circular progress bar shows that the side effect of that input modality was considered. However, this design decision was also influenced by the smaller screen size of mobile devices. If the screen size was bigger, there is no reason to say that the progress bar couldn't have been placed elsewhere on the screen. In fact if the screen size was sufficiently bigger, feedback could've been provided in the same manner as in the PC and Console versions.

## 5 Guidelines for Visual Scale Design

After analyzing the various differences that exist between the different versions of Minecraft, we have highlighted what we feel were the key factors that influenced the design decisions when porting the game across visual scale. Based on this, we present guidelines for Visual Scale Design. Adherence or consideration to these guidelines would result in a game that is better prepared for operation across visual scale by providing the player with a consistent *experience*.

### Guidelines for Visual Scale Design

1. Ensure that the player's efficiency does not deteriorate across visual scale.  
Adhere to Fitt's Law.
2. Consider the side effects of Input Modalities.  
Account for the player's hand blocking the screen with touchscreen mobile devices.
3. Prioritize the core mechanics of the game.  
This is to ensure that the core functionality is preserved across scale, while detail is added only when feasible.
4. **TO DO: More?**

## 6 Conclusion

**TO DO: porting may not be the right word..**

Visual Scale proves to be an important factor when *porting* a game across different screen sizes. We looked to analyze a game currently on the market which operates at different visual scales, in this case Minecraft. Our analysis of Minecraft across visual scale allowed us to discover several differences between the different versions. We provided a rationalization, backed by research in HCI and Cognitive Psychology, for these differences and developed guidelines for Visual Scale Design. With adherence to these guidelines, we believe that games operating across visual scale will convey a consistent *experience* to the player.

We believe this is a first step towards further developments in the area of Visual Scale Design. Future work will build upon this research to further aid developers in creating games across visual scale.