Assignment P3

Cleo Zhang

yzhang3761@gatech.edu

# 1 QUESTION 1

## 1.1 Principles that support the invisible interface design

### 1.1.1 *Discoverability*

To support invisible interface design, the principle of Discoverability narrows the gulf of execution by simplifying the action identification phase. Based on the discoverability principle, the relevant functions should be designed straightforward and visible, so users can discover them and identify the actions as opposed to learning the functions via documentation and tutorials. Therefore, users do not need to look for the tool/interface they need but the task they want to accomplish.

### 1.1.2 Affordances

The principle of Affordances supports invisible design by making the design of the thing hint at the way they are supposed to be possibly used. This principle considers both novices and experts when designing interfaces, reduces the time needed to learn how to use the interface, and speeds up the action execution phase. Once users identify the actions they need to take to complete a task, they can start executing the actions - the interface is invisible because the interface itself affords the way it is meant to be used.

### 1.1.3 *Consistency*

The principle of Consistency allows users to leverage the usability from existing tools when interacting with the interface, so they do not need to spend extra time figuring out how to use the interface. This principle narrows the gulf of execution by better assisting users in the action execution phase and, therefore, supports the invisibility of the interface design.

## 1.2 Principles that emphasizes the participant view of the user

### 1.2.1 Flexibility

The Flexibility principle emphasizes the participant view of the user when interacting with an interface. It requires the interface design to cater not just to the user's expertise level but also the external factors that could affect the user interface - physical environment and possible access constraints, for example. With various accessibility and usability designed, users can choose the way they engage most naturally and have different interactions with the interface to accomplish their tasks.

### 1.2.2 Constraints

The Constraints principle also improves users' efficacy when interfacing with the interface from the participant view. This principle helps reduce the errors by setting a limited set of actions and preventing users from taking incorrect steps against the interface in the first place. The interface should let users know if they are heading in the wrong direction when using the interface and assist them to go back on track to complete their task.

# 2 QUESTION 2

Throughout my everyday work, my manager requires each team member to anonymously submit a feedback form after each meeting, and the feedback form itself is intolerant to errors. My manager needs all the questions, but the users can still submit the form with empty answers to any questions. The existing interface design does not respond to user error of leaving the question unanswered and relies on users themselves to discover the problem. Such user errors likely happen because people are very used to answering the easy questions first and leave the hard ones to later - and they may forget to answer those questions at all.

Setting constraints will improve the current design of the feedback form. The feedback form should prevent the users from submission if there are any questions left empty. Users should also be notified about why they cannot submit the form so they would know what kind of issue they should be looking to fix. In such a way, users would not provide any questions with null answers, and the manager will receive more meaningful feedback for the meetings.

Mappings can also improve the user experience of filling out the feedback form. For example, a progress bar on the form can give users an overview of the current form status - solid colour for answered questions and grey for unanswered ones. This visualization can help remind users of the unanswered question by leveraging the existing knowledge of how a progress bar works in other interfaces, such as a video player's progress bar.

Affordance can be an excellent approach to avoid errors as well. First, required questions should be marked with a red asterisk, so new employees know they need to answer all the questions before the submission. Secondly, the unanswered questions should be marked by a red bracket when users hit the submit button - then users should know why the submission is not successful and how to correct the action to fix the issue.

# 3 QUESTION 3

In this question, I will be discussing the slips and mistakes that could happen in my favourite co-op cooking game - Overcooked. In each subsection, I will be describing the slip/mistake and why it happens, followed by a brief suggestion on the existing design of how to possibly prevent such slip/mistakes in the future.

## 3.1 Slip

In Overcooked multiplayer mode, the most common slip is that the players think they are controlling other players' characters and wondering why the controller is not doing what they expect. There are many reasons why the players may make this slip. First, they may still think about the character they controlled from the last round of the game and forget that character is controlled by other players now. The digital characters may look very alike, so the players get confused after a while.

One possible approach to eliminate this slip is to offer an onboarding session before the actual game starts. So the players can be more familiar with the character they are controlling for the current round of the game.

## 3.2 Mistakes

One possible mistake that players can make is putting the dishes in the sink, expecting the dishes to wash and be ready to be used after a few seconds. Players who are new to the game are most likely to make this mistake as this dishwashing task has two sub-tasks which are different from cropping and serving. Players make this mistake because they do not logically know how to "wash a dish" in this game.

The game interface should give players a quick hint to let them know they need to put the dishes in the sink then stand by the sink to wash the dishes. Such feedback from the game will help make the game experience more smooth.

## 3.3 Something Challenging

Mapping the controller buttons to the character's behaviour can be difficult because we can barely leverage the existing knowledge of controlling from other games. Players usually need to practice more to memorize the commands/gestures required to play the game better.

This learning curve is hard to be flattened. It makes the game challenging as the game controllers are only used under gaming scenarios - not daily life. The controllers interact with different types of games in different ways. Players need to spend some time adapting to the controller rules that are unique to the game they choose to play.

# 4 QUESTION 4

## 4.1 Good interface

Canvas gives a good representation for providing students with class-related information. The underlying content of the Canvas interface is class details - lessons, reading materials, assignments and so on. The Canvas organizes and represents the relevant class details by stating the explicit relationship between different types and hiding the extraneous information.

To make the relationship explicit, Canvas provides different visual blocks/menus to help students navigate to the content they are looking for. For example, students will see all the classes they have registered for the current semester once they log in. After they click on one of the classes, the learning materials, syllabus and assignment submission portals for this specific class show up. If students go to this class and submit an assignment, they would explicitly know they are submitting the work for this particular class.

Canvas excludes extraneous details for students, and canvas does not have a class registration portal or tuition payment option displayed as a gateway for students to check on their classes. Keeping only relevant information can help optimize students' cognitive load so they can focus more on their tasks.

## 4.2 Poor interface

The Tuya app can be an example of poor representation of the interface. As an app that is supposed to help users to configure their smart/wireless outlets, the interface does not expose the natural constraints of connecting natural outlets. It does not construct explicit relationships for its underlying content.

This interface has no required voltage or network information to connect the wireless outlet to the home network. As a result, users sometimes do not know the root cause of why the outlet cannot be connected and spend a lot of time figuring things out and fixing the problems. Not setting the constraint may lead to more user errors than it should be.

The relationship between the intelligent outlets, home network and the app is not explicit. The interface does not have clear instructions on how those three components work together, and users do not always know where to start - especially for novices who are not familiar with the smart home system.