Heuristic Evaluation for Games: Usability Principles for Video Game Design

David Pinelle
University of Nevada, Las Vegas
4505 Maryland Parkway
Las Vegas, NV 89154-4019
pinelle@cs.unlv.edu

Nelson Wong
University of Saskatchewan
110 Science Place
Saskatoon, SK, Canada S7N 5C9
nelson.wong@usask.ca

Tadeusz Stach
Queen's University
25 Union Street, Goodwin Hall
Kingston, ON, Canada K7L 3N6
tstach@cs.queensu.ca

ABSTRACT

Most video games require constant interaction, so game designers must pay careful attention to usability issues. However, there are few formal methods for evaluating the usability of game interfaces. In this paper, we introduce a new set of heuristics that can be used to carry out usability inspections of video games. The heuristics were developed to help identify usability problems in both early and functional game prototypes. We developed the heuristics by analyzing PC game reviews from a popular gaming website, and the review set covered 108 different games and included 18 from each of 6 major game genres. We analyzed the reviews and identified twelve common classes of usability problems seen in games. We developed ten usability heuristics based on the problem categories, and they describe how common game usability problems can be avoided. A preliminary evaluation of the heuristics suggests that they help identify game-specific usability problems that can easily be overlooked otherwise.

Author Keywords
Usability evaluation, heuristic evaluation, video games

ACM Classification Keywords
H5.2 [User Interfaces]: Evaluation/methodology

INTRODUCTION

One of the main goals in video game design is to entertain and engage the user. This can involve several aspects of design, including game story, pacing, challenge level, and game mechanics [7]. However, since most games require constant interaction, game designers must also pay careful attention to usability issues. Failure to design usable game interfaces can interfere with the larger goal of creating a compelling experience for users, and can have a negative effect on the overall quality and success of a game.

In this paper, we define game usability as *the degree to which a player is able to learn, control, and understand a game.* Our definition is based on an early informal survey of usability problems cited in critical game reviews and on playability heuristics described by Federoff [12] and Desurvire et al. [7]. Game usability does not address issues of entertainment, engagement, and storyline, which are strongly tied to both artistic issues (e.g. voice acting, writing, music, and artwork) and technical issues (graphic and audio quality, performance issues).

Some usability issues seen in games are similar to those seen in other application areas, such as the need to design for visual consistency and readability. However, games also have usability considerations that are not seen in other areas. For example, user errors are usually undesirable in other domains, but are expected in many games since they are designed to challenge users and to force them to develop new skills so that they can achieve in-game objectives.

Game designers need methods for identifying usability problems both in early designs and in more mature prototypes [7]. Playtesting is one of the most common ways to uncover design problems [13], yet this method needs a playable prototype that only exists in the later stages of the development process. Formal methods do not exist to allow designers to carry out less expensive usability inspections of games, and to evaluate early, non-functional prototypes. Further, many common usability inspection techniques are not appropriate for games since they either rely on formal specifications of task sequences [3,4,19,29] or are oriented around user interface concepts used in desktop applications [24].

One technique that has the potential to be useful in allowing evaluations of game prototypes is heuristic evaluation. Heuristic evaluation is an inspection technique where evaluators explore an interface using a set of usability principles, called heuristics [24]. Heuristic evaluation does not make assumptions about task structure, and it is flexible enough to be adapted to specialized domains [25,10].

We believe that a custom set of design principles are needed so that heuristic evaluation can be used to find usability problems in video games. Several researchers have

CHI 2008, April 5–10, 2008, Florence, Italy.

previously developed heuristics for game design [5,7,12]. However, their heuristics are strongly oriented around engagement and fun, and do not consider usability in detail. Also, the heuristics are primarily based on literature reviews or author introspection, rather than on detailed information about design problems that commonly occur in games.

In this paper, we introduce a new set of heuristics for evaluating the usability of video games. They are the first design principles that focus exclusively on game usability and that are based on a structured analysis of usability problems in a large number of games, covering several major genres.

In the rest of this paper, we describe the process that we used to develop the heuristics, which included three main steps. First, we identified video game usability problems in 108 PC game reviews from a popular gaming website. Second, we categorized the problems descriptions. Third, we developed ten usability heuristics based on the problem descriptions and problem categories. We also present a preliminary evaluation of the heuristics, where five evaluators used them to evaluate a PC game demo.

USABILITY EVALUATION AND VIDEO GAMES

Usability inspection techniques have the potential to improve the game design process. Unlike playtesting [13], they do not need user participation, so they can be carried out with early mockups (as well as with functional prototypes). Usability inspections rely on skilled evaluators who inspect the user interface and identify usability problems. They are inexpensive and can be carried out in a short amount of time, so inspections can be used iteratively during the design process [24].

Most current usability inspection techniques are not appropriate for games since the design considerations are significantly different from those seen in other applications. Techniques such as cognitive walkthrough [19,29], pluralistic walkthrough [3,4], and task analysis [1,30] are partially based on the assumption that people will use an application to carry out predetermined tasks. However, the notion of task sequences is not necessarily useful in games since people play them differently depending on their strategy or motivation; further, some games are designed to promote unstructured exploration, which means that significant variability can be seen in how people choose to interact with game interfaces.

Heuristic evaluation has the potential to be a valuable evaluation tool for games since it does not make assumptions about tasks and about the purpose of an application. Heuristic evaluation is an inspection technique that gives evaluators significant freedom in how they conduct the evaluation [24]. Evaluators explore the interface while looking for instances where it does not adhere to a set of usability principles called heuristics. Several studies suggest that one of the benefits of using heuristic evaluation is that it helps designers find important classes of problems that are not always found with user testing [15,18].

The unstructured nature of heuristic evaluation has also led to several criticisms of the technique. For example, Cockton and Woolrych [6] point out that heuristic evaluation does not encourage people to take a comprehensive view of how software will be used, and that analysts must pick sample tasks and system features at random. They also indicate that using heuristic evaluation, and discount methods in general, can cause evaluators to miss problems and to identify false problems. They conclude that discount methods are best suited to use in early design iterations rather than as a summative evaluation technique.

Nielsen developed his heuristics primarily for desktop applications [24]. They refer to common user-interface concepts, such as dialogs, undo and redo, and error prevention. However, many of these ideas have limited meaning in the game context. While they may be useful in uncovering some usability problems in games, they do not address several important usability issues, such as using proper camera angles when displaying the game world or providing intuitive control mappings.

We believe that heuristic evaluation can be adapted so that it can be used to find important usability problems in games. Our goal in this research was to develop a new set of heuristics that can be used by game designers to evaluate both early mockups and functional prototypes. In the next section, we discuss related literature on game design principles.

DESIGN PRINCIPLES FOR VIDEO GAMES

Design considerations for video games are different from those of other types of software [2,16,17]. This means that new principles are needed for game design and evaluation. Past work on game design has mainly focused on engagement and entertainment issues, and most researchers have not considered usability in detail.

Malone [20] developed a set of guidelines for designing enjoyable interfaces, and they were partially based on studies of educational games. His research focused on software design in general, rather than on identifying principles that can be applied to other games. Clanton [5] developed a set of game design principles that have a similar focus to Malone's work. His principles were based on introspection and on an informal inspection of several games. Clanton's design principles address how games can be designed to engage users (e.g. "pressure can be fun" and "avoid linear, monotonous pacing"). They do not emphasize usability issues, although several related concepts are briefly discussed (e.g. "confusion is not fun", "provide a gentle on-ramp").

Federoff [12] compiled a list of game heuristics that focus on three areas: game interface, game mechanics and game playability (design areas that were proposed by Clanton

[5]). Federoff created the heuristics by reviewing literature on game design, and further refined the set after spending a week with designers at a game company. Most of her heuristics focus on gameplay, engagement, and storyline issues. She also developed heuristics addressing usability issues, but they are not comprehensive and are not described in detail. For example, heuristics include: "for PC games, consider hiding the main computer interface", "minimize control options", and "the interface should be as non-intrusive as possible."

Desurvire et al. [7] developed Heuristics for Evaluating Playability (HEP), a set of game heuristics that have strong similarities to those developed by Federoff [12]. HEP was developed from literature on game design, and the organization is roughly based on the categories described by Clanton [5]. The HEP heuristics place a strong emphasis on engagement, with limited coverage of usability issues.

Current game heuristics can be useful in evaluating some aspects of game design. However, they do not consider usability in detail, so they are not well suited for carrying out usability evaluations of video games. We believe that a set of heuristics that focuses on game usability can help improve the video game design process. In the next section, we describe how we developed a set of heuristics from usability problems found in a database of game reviews.

HEURISTIC EVALUATION FOR GAMES

We created a set of game heuristics to fill two main roles in the game design process. First, the heuristics can serve as a set of design principles that can be used during the formative stages of game design and development. Second, they can be used to carry out usability inspections where evaluators use them to critique the design.

Overview of Research Process

Our method for defining game usability heuristics was initially motivated by an approach used by Nielsen when he defined a new set of general-purpose heuristics [23]. He evaluated several existing sets of heuristics to determine which ones provided the "broadest explanatory coverage" of a set of 249 usability problems. Our considerations were somewhat different, however, since literature on usability heuristics for games is limited, and since studies of usability problems in games have not been previously reported in detail.

Our approach is more closely aligned with that described by Dykstra [10], who suggests that heuristics can be developed for specific software categories by evaluating existing products, and by developing principles that describe the usability problems that are found. However, given the wide variability that we expected to see in game usability problems, carrying out the evaluations ourselves was not practical. We expected to see significant variation in usability problems found both between games, and between game genres. Therefore, we felt that we needed broad coverage of the design space, and that we needed to include a large number of games that represent the main game genres.

The process that we used to develop the heuristics had three main steps:

1. Identify real world usability problems that provide breadth and depth coverage of the game design space
2. Develop a set of categories that group similar usability problems
3. Create heuristics that are the inverse of the problem categories and that describe how common usability problems can be avoided

In the next sections, we describe each of the steps in this process. First, we describe the problem identification step, where we analyzed 108 PC game reviews from the GameSpot website. Second, we describe how we developed twelve problem categories based on the problems. Third, we describe how we developed ten usability heuristics from the problem descriptions and problem categories.

Step 1: Identify Problems in Game Reviews

We identified and classified usability problems that were described in game reviews posted on the GameSpot website (www.gamespot.com). GameSpot is one of the most popular gaming sites, and it maintains an extensive archive of video game reviews stretching back to 1996. The review archive provided us with an opportunity to study usability problems across several different game genres and in a large number of games.

GameSpot reviews are written by professional editors, and most reviews are relatively comprehensive, addressing a range of game design issues. The reviews usually consider how engaging and entertaining a game is, as well as the overall graphic and audio quality. Usability issues, such as control mapping and the understandability of visual representations, are also covered in most reviews, and analyzing games from the archive allowed us to identify common usability problems and to understand the common ways that the problems occur in game interfaces.

GameSpot editors assign each game a composite score based on its overall quality, and scores range from 0-10, with 10 representing a perfect game without any problems (a score that is never used in practice). We carried out an early pilot study and did not find usability problems in highly rated games, so we limited the formal study to games with scores of 8.0 or lower. We also limited the study to PC games since computers support a wide range of input devices (e.g. keyboard, mouse, gamepad, USB devices), and since games from most major genres are released for PCs.

We used 108 reviews from the GameSpot website, and we restricted our selection process to reviews that were posted since 2001. Our goal was to identify usability problems that were relevant to current games, and that reflected recent trends in game design. Within the 108 reviews, we included 18 reviews from each of 6 major game genres: role playing, sports/racing, first person shooter/tactical shooter, action, strategy (both real-time and turn-based), and adventure. The genres were based on the main game categories listed on the GameSpot website, and are consistent with most of the major genres listed by the Entertainment Software Association [11]. The selected reviews included work from 24 different reviewers.

We assigned the reviews to three researchers, each of whom had significant experience with playing games and with conducting usability evaluations. Each person worked separately on the review sets to reduce bias, so that they could independently draw conclusions about the problem classes that are found in games. They identified single-user usability problems that were found in each review, and excluded technical problems and issues related to fun and engagement. They separately devised problem categories based on the problems found, and individually coded problems from the reviews using the categories.

After all reviews were coded, the researchers met to discuss the problem classifications and to develop a common classification scheme. There were 50 problem categories in total. Six problem categories were discarded because they focused on technical issues such as bugs, audio, or video issues; two were discarded because they focused on engagement issues; and two were discarded because they were only found by one person and the other two people did not believe that they were real usability problems. They aggregated the remaining categories based on similarities between problem descriptions, and a total of twelve new problem categories were identified. Nine of the categories combined similar problems that were described by all three people, and the remaining three combined problems described by two people.

The researchers recoded the reviews using the new problem categories and tracked the number of times each problem was found in the review set (see Table 1). They identified a total of 285 usability problems, an average of 2.64 problems per game, and at least one usability problem was found in each game review.

Step 2: Develop Categories that Group Problems

Table 1 shows the problem categories that were developed from the reviews. The categories provide high level descriptions of problems that are seen in games, but they do not give concrete details on how the problems are instantiated. For each category in Table 1, several associated key issues are listed. These issues describe the main ways that problems in the category occurred. For example, category 7 is “difficult to control actions in the game”, and the key issues (in column 2) are “oversensitive controls, unnatural controls, unresponsive controls.” Table 1 shows example problem descriptions for each problem category (column 3), and the example listed for category 7 illustrates the key issue of “oversensitive controls.”

The total number of problems found for each category is shown in Table 1. Problems from eight categories were found in all six genres (67%, categories 1,4,6,7,8,9,10, and 12). There were significantly fewer problems found in some categories than in others. For example, problem categories 2 and 5 only have 11 and 6 problems, respectively. They were included in the list because problems from each category were found in multiple genres (category 2 in 5 genres; category 5 in 2 genres), and because the problems had a significant impact on the usability of the game. For example, category 2 is “does not allow enough customization.” In some games, users cannot customize the game speed, and the default settings make the game too difficult for many users. Category 5 is “does not let user skip non-playable content.” Games with problems in this category force the user to watch video and audio sequences and do not provide a method for skipping the content. This is a significant problem since some clips are repeated several times during a game, for example, when a user moves her avatar into a room, or when she initiates a conversation with a computer-controlled character.

Step 3: Develop Game Heuristics

We used the problem categories and descriptions to develop ten usability heuristics. The heuristics describe design principles that are intended to help designers avoid common usability problems seen in video games. One of our main objectives was to create heuristics that could easily be operationalized by designers that have experience with games, so each includes a detailed paragraph that describes how problems associated with the heuristic can be avoided.

Each heuristic is based on problem categories in Table 1. The heuristics were created by translating the problems into principles that provide guidance on how they can be avoided. For example, the problem category “unpredictable / inconsistent response to user’s actions” became the heuristic “provide consistent responses to user’s actions.” In most cases, there is a one-to-one correspondence between the problems and the heuristics, except for problems 10 and 12, which we combined with others during the translation process (problems 6 and 10 were combined to create heuristic 6; problems 7 and 12 were combined to create heuristic 7). The heuristics are listed in Table 2.

Each heuristic also contains a paragraph that provides more detail on how related usability problems can be avoided (see Table 2). Each paragraph is grounded in our observations and the individual game reviews: they are based on the key issues associated with the problems that were used to create the heuristic, and on additional details that were found in the review set. The paragraphs describe how each problem stated in the key issues section of Table 1 can be avoided. The paragraphs were developed to help designers operationalize each heuristic by making it easier for them to map the heuristics to specific game features.

Problem category	Key issues	Example	Total
1. Unpredictable / inconsistent response to user's actions	poor hit detection, poor in-game physics, inconsistent response to input	"You'll often find yourself swinging away and watching your blade go right through your foes with no effect whatsoever." [21]	42
2. Does not allow enough customization	does not allow user to change video and audio settings, difficulty, or game speed	"Worst of all, there is no way to adjust the game's speed. You're stuck constantly veering between way-too-fast combat and way-too-slow travel to engage enemies and explore maps." [34]	11
3. Artificial intelligence problems	problems with pathfinding, problems with computer controlled teammates	"Your own teammates are a mixed bag in terms of intelligence, as they're sometimes good about getting behind cover and laying down covering fire, but they also do crazy things, like rushing forward into the open or constantly getting in your line of fire." [27]	22
4. Mismatch between camera/view and action	bad camera angle, view is obstructed, view does not adjust to user's action quickly enough	"If you leave the camera alone you'll either be running around blind or the camera will get stuck on something and just twitch randomly for a while." [22]	23
5. Does not let user skip non-playable content	cannot skip video and audio clips, frequently repeated sequences	"Making matters even worse, you can't skip over any of this blah-blah-blah wordiness and often have to exhaust all of the dialogue options to open up new discussion topics or the ability to solve a problem." [35]	6
6. Clumsy input scheme	bad input mappings, limited device support, limited control customization	"You can't type your name in—you must instead select letters with your mouse. And you can't use the keyboard to navigate." [14]	21
7. Difficult to control actions in the game	oversensitive controls, unnatural controls, unresponsive controls	"Chopper controls run the gamut from very touchy to absurdly touchy. For instance, it's almost impossible to keep an Osprey's rotors pointed toward the sky." [36]	31
8. Does not provide enough information on game status	does not provide adequate information on character, game world, or enemies. visual indicators, icons, and maps are inadequate.	"The map function reveals everything about a particular location except where you are, making it useless until you do some heavy exploring to get your bearings." [31]	28
9. Does not provide adequate training and help	does not provide default and recommended choices; does not provide suggestions and help; does not provide adequate documentation, instructions, tutorials, and training missions.	"A tutorial and a better online manual are sorely needed. There is a really nifty feature where you can ask specific questions of your wrestlers and booking team, but it's just not enough." [32]	36
10. Command sequences are too complex	learning curve is too steep; requires too much micromanagement; command sequences are complex, lengthy, and awkward, making the game difficult to play	"Not only does the main map have four modes and many subscreens, but nearly all the political activities take place on separate screens designed more to resemble scenes in Roman life than to maintain any continuity to the interface." [28]	20
11. Visual representations are difficult to interpret	bad visualization of information, too much screen clutter, too many characters or game elements on the screen at the same time, difficult to visually distinguish interactive content from non-interactive content	"Buildings are completely barren. And the satellite-styled overhead map is tough to read, because it consists of mashes of green terrain and colored blobs representing enemy soldiers and objectives." [33]	16
12. Response to user's action not timely enough	slow response time interferes with user's ability to interact with the game successfully	"Often, though, you'll need to hit the key before the indicator appears, as the game takes a moment to register your keystroke." [8]	29
		Combined Total	285

Table 1. Problem categories. Categories are listed with key issues, example problem descriptions, and total problems found.

1. Provide consistent responses to the user's actions.
Games should respond to users' actions in a predictable manner. Basic mechanics, such as hit detection, game physics, character movement, and enemy behavior, should all be appropriate for the situation that the user is facing. Games should also provide consistent input mappings so that users' actions always lead to the expected outcome.

2. Allow users to customize video and audio settings, difficulty and game speed.
The video and audio settings, and the difficulty and game speed levels seen in games are not appropriate for all users. The system should allow people to customize a range of settings so that the game accommodates their individual needs.

3. Provide predictable and reasonable behavior for computer controlled units.
In many games, the computer helps the user control the movement of their character, of a small group of teammates, or of a large number of units. Computer controlled units should behave in a predictable fashion, and users should not be forced to issue extra commands to correct faulty artificial intelligence. The game should control units so that pathfinding and other behaviors are reasonable for in-game situations.

4. Provide unobstructed views that are appropriate for the user's current actions
Most games provide users with a visual representation (i.e. a "view") of the virtual location that the user is currently occupying. The game should provide views that allow the user to have a clear, unobstructed view of the area, and of all visual information that is tied to the location. Views should also be designed so that they are appropriate for the activity that the user is carrying out in the game. For example, in a 3D game different camera angles may be needed for jumping sequences, for fighting sequences, and for small and large rooms.

5. Allow users to skip non-playable and frequently repeated content.
Many games include lengthy audio and video sequences, or other types of non-interactive content. Games should allow users to skip non-playable content so that it does not interfere with gameplay.

6. Provide intuitive and customizable input mappings.
Most games require rapid responses from the user, so input mapping must be designed so that users can issue commands quickly and accurately. Mappings should be easy to learn and should be intuitive to use, leveraging spatial relationships (the up button is above the down button, etc.) and other natural pairings. They should also adopt input conventions that are common in other similar games (e.g. many first-person shooters and real-time strategy games use similar input schemes). Games should allow users to remap the input settings, should support standard input devices (e.g. mouse, keyboard, gamepad), and should provide shortcuts for expert players.

7. Provide controls that are easy to manage, and that have an appropriate level of sensitivity and responsiveness.
Many games allow users to control avatars such as characters or vehicles. Controls for avatars should be designed so that they are easy for the user to manage, i.e. they are not too sensitive or unresponsive. When controls are based on real world interactions, such as steering a car or using a control stick in an airplane, the game should respond to input in a way that mirrors the real world. Further, games should respond to controls in a timeframe that is suitable for gameplay requirements.

8. Provide users with information on game status.
Users make decisions based on their knowledge of the current status of the game. Examples of common types of information that users need to track include the current status of their character (such as their health, armor status, and location in the game world), objectives, teammates, and enemies. Users should be provided with enough information to allow them to make proper decisions while playing the game.

9. Provide instructions, training, and help.
Many games are complex and have steep learning curves, making it challenging for users to gain mastery of game fundamentals. Users should have access to complete documentation on the game, including how to interpret visual representations and how to interact with game elements. When appropriate, users should be provided with interactive training to coach them through the basics. Further, default or recommended choices should be provided when users have to make decisions in complex games, and additional help should be accessible within the application.

10. Provide visual representations that are easy to interpret and that minimize the need for micromanagement.
Visual representations, such as radar views, maps, icons, and avatars, are frequently used to convey information about the current status of the game. Visual representations should be designed so that they are easy to interpret, so that they minimize clutter and occlusion, and so that users can differentiate important elements from irrelevant elements. Further, representations should be designed to minimize the need for micromanagement, where users are forced to interactively search through the representation to find needed elements.

Table 2. Game heuristics. Each heuristic is listed along with a paragraph describing how common problems can be avoided.

There are several similarities between our heuristics and Nielsen's heuristics [24]. For example, Nielsen's "consistency and standards", "visibility of system status", and "help and documentation" are similar to heuristics 1, 8, and 9 in our set. The main difference is that our heuristics use language that makes the links to the game domain more explicit. For example, Nielsen's "consistency and standards" heuristic states that, "users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions." Our heuristic that addresses consistency, "provide consistent responses to the user's actions", states that, "…basic mechanics, such as hit detection, game physics, character movement, and enemy behavior, should all be appropriate for the situation that the user is facing. Games should also provide consistent input mappings so that users' actions always lead to the expected outcome."

Many of our heuristics are new and are not similar to mainstream usability principles that are used for desktop application, such as heuristics 2, 3, 4, and 7. For example, heuristic 4 deals with the need to match the user's view of the game world with their current actions. This is often seen in 3D games, where different views are needed when users move their characters through doorways, open spaces, and confined spaces. Other heuristics address the need for customizable speed and difficulty settings; reasonable artificial intelligence when the computer helps to control units; and the need for controls that have the appropriate level of sensitivity.

Our heuristics have some similarities with heuristics developed by other game researchers [7,12]. For example, Desurvire et al. [7] describe a heuristic related to consistency: "Game should react in a consistent, challenging, and exciting way to the player's actions (e.g., appropriate music with the action)." They also address control design, artificial intelligence, and information on game status. However, the content and focus of the heuristics are significantly different. Their heuristics focus on engagement issues, but ours are strongly oriented around usability and are augmented by detailed information on common usability problems.

INITIAL EXPERIENCES WITH THE HEURISTICS

We conducted a preliminary evaluation of the heuristics where five people used them to evaluate a demo version of a PC game. Our main goal was to evaluate their overall usefulness at identifying usability problems in games. We also wanted to determine whether knowledgeable evaluators would be able to understand and operationalize the heuristics, and whether the level of detail provided in each was appropriate.

Method

We recruited five evaluators who had significant experience with playing video games and had formal training in carrying out usability evaluations. We gave the evaluators training on the game heuristics and asked them to evaluate a demo version of a real-time strategy game called Necromania: Traps of Darkness. The game was not part of the review set used to create the heuristics. Each evaluator worked individually, and they were asked to spend some time familiarizing themselves with the game. They were then asked to inspect the game, and to identify instances where it did not adhere to the principles outlined in the heuristics. If they felt that the mismatch was a usability problem, they were instructed to write the problem on a form and to record the heuristic that they used to find the problem. They were also asked to give each problem a severity rating using Nielsen's severity scale [26]: 1-Cosmetic problem, 2-Minor problem, 3-Major problem, and 4-Usability catastrophe.

Each evaluator completed an open-ended questionnaire at the end of the study. We asked participants to describe the strengths and limitations of using the heuristics to evaluate games. We also asked them whether they found any heuristics to be particularly useful, and whether any were difficult to operationalize. Finally, we asked whether they found problems using the heuristics that they would have overlooked otherwise. The entire session lasted approximately 90 minutes, and evaluators were paid for their participation.

We analyzed the problem descriptions. We removed problems that were repeated by the same evaluator, and those that did not describe real usability problems (e.g. commentary on gameplay, preferred features, etc.). We confirmed that the evaluators listed heuristics that were related to the specified problems. We counted the number of problems that were found using each heuristic, and we tracked the severity ratings that were assigned to each problem (see Table 3).

Results

Each person found between 7 and 10 problems (45 problems in total, an average of 9 problems per evaluator). Table 3 shows the number of problems found with each heuristic. The total number of problems in the table is higher than 45 since two evaluators listed several heuristics on some problem reports. Problems were found with all heuristic with the exception of heuristic 5. Heuristics 6,8,9, and 10 were used most frequently. The third column on Table 3 shows the mean severity rating for problems identified with each heuristic.

There was significant overlap in the problems found by the evaluators, but some problems were found by only a single person. Four people described the absence of an in-game help system (heuristic 9). One person wrote: "*No help system or tutorial. Still don't understand what I'm supposed to be doing.*" Other commonly described problems include the absence of a map or other indicator showing where the user is located in the game world (heuristic 8): "*No overview map or radar is given, making it very difficult to understand where you are, and easy to get lost*", and problems with default input mappings (heuristic 6): "*Default key mappings poor. Uses numberpad and arrow keys for left hand. Have to move between the two positions. Better to have them all together on the left side of the keyboard*."

Heuristic	Total	Mean severity
1. Consistency	4	3.00
2. Customizability	2	2.00
3. Predictability	4	3.25
4. Proper views	3	3.00
5. Skip non-playable content	0	-
6. Input mappings	8	2.25
7. Controls	4	2.30
8. Game status	10	2.75
9. Training and help	12	3.30
10. Visual representations	7	2.43

Table 3. Summary of results. First column shows heuristic, second shows total problems, and third shows mean severity.

Four participants indicated that the heuristics allowed them to identify problems that they would have failed to noticed otherwise. When asked whether they found any heuristics to be particularly useful, heuristics 1,4,6,8, and 9 were listed. Heuristic 9 (provide instructions, training, and help) and 6 (provide intuitive and customizable input mappings) were cited by two people, and the others were cited once. When evaluators were asked whether there were any heuristics that they had problems understanding or operationalizing, four indicated that they did not have any problems. One participant stated that he had problems operationalizing heuristic 6 because he struggled to determine whether the controls were intuitive.

Evaluators described two limitations that they encountered when using the heuristics. Two people indicated that the heuristics could cause evaluators to focus on a limited set of issues. One person wrote that, "*Some problems could be missed if they don't fall in the categories. It may also put the subject in a certain mindset, causing them to overlook certain issues.*" However, we believe that this is a criticism of conducting evaluations with guidelines and heuristics in general but is not a specific critique of the game heuristics. The second limitation was described by a single evaluator, who wrote that there can be "*lots of overlap between different heuristics for some problems*." Again, we do not feel that this is a significant problem since one of our main objectives was to provide broad coverage of the design space, and overlap, when it does occur, is far preferable to not providing enough coverage to detect significant usability problems.

The evaluators listed several main benefits of using the heuristics. All evaluators indicated that the heuristics were a good match for game interfaces. One evaluator indicated that "*their terminology was suited for the task*" and another wrote that the heuristics, "*fit game interfaces well…and helped steer criticism away from non-interface issues while reviewing*." Three others indicated that the heuristics were particularly well suited to uncovering usability problems in games. For example: "*Most are easy to remember, and nearly everything that is frustrating about the usability of the game tested was easily identified with a heuristic*," and, "*very focused on gameplay and appropriate for typical problems associated with usability*."

DISCUSSION

Summary of evaluation results

Our main goal was to develop heuristics that could be used to specialize the usability inspection process for video games. We also had three related objectives. First, we wanted to provide thorough coverage of usability problems that can occur in a wide variety of games. Second, we wanted the heuristics to be easy to operationalize for people with knowledge of usability principles and with some experience working with games. Third, we wanted people to be able to use the heuristics to identify real problems that will affect the usability of games.

Our evaluation provides initial insights into how successful the heuristics were at meeting these objectives. People were able to find a wide range of problems, and they found multiple problems with nine of the heuristics. The evaluation does not provide a comprehensive look at the heuristics since we do not expect any game to necessarily have problems in all areas of coverage. However, it does show that the evaluators were able to understand the heuristics, and inspection of the problem descriptions shows that they did match the area of coverage described by the specified heuristic. This also shows that people were able to use the heuristics to find real problems, and they had little difficulty with operationalizing them.

A thorough evaluation of the heuristics will require additional testing, where more evaluators are involved, and where the heuristics are used to evaluate games from different genres. However, we believe the results provide us with enough evidence to conclude that the heuristics are well suited to uncovering important usability problems in the game context.

Scope and generalizability

Since the heuristics were formulated from usability problems that were found in games from several major genres, we believe that they can be used to evaluate most games. However, it is not clear how they relate to games that are significantly different from the ones we included in the heuristic development process. In this section we discuss several issues related to the scope and generalizability of the heuristics, including their relationship to: console games, multiplayer games, and serious games.

We developed our heuristics from PC game reviews rather than from reviews of console games (e.g. PlayStation, Xbox, Wii, etc.). We expect that the heuristics will still generalize to most console games since many of the design issues are the same. However, further work is needed to verify this, and to guarantee that there are no additional design considerations for consoles. For example, it is possible that new heuristics may be needed to address recent changes in input technologies used in consoles, such as the motion and position tracking technologies used in the Wii remote.

We focused on single user issues when developing the heuristics. We made this choice because of the level of complexity seen in multiplayer games in recent years, such as the development of massively multiplayer online games. We believe that multiplayer issues require extensive study to develop a further understanding of the usability issues involved, such as team awareness, session management, and communication considerations. We found that focusing on single user issues heuristics suited the way that games are typically implemented, since many games support single user play only, or have separate single player and multiplayer modes.

We focused specifically on commercial games that have a primary goal of entertaining users. There is another class of games, called serious games, which are designed for non-entertainment purposes, such as training, education, or simulation. It is unclear how our heuristics relate to serious games, but it is likely that the differences in design goals mean that a new set of usability principles are needed.

There are possible limitations to the approach we used when we developed the heuristics. The editors who wrote the reviews are not usability professionals, and did not focus on usability exclusively in their critique of the games. The reviews were written primarily for the benefit of game players, and they consider design and technical issues that interfere with the overall playability of the game. We believe the reviews provide thorough coverage of major usability problems, but it is possible that more subtle design problems were missed. We expect that the heuristics will continue to grow and evolve in the future as we gain new insights by carrying our additional usability evaluations of video games.

The heuristics do not address design issues related to how fun and engaging games are for users. We intentionally focused on usability issues to keep the scope of our heuristics manageable. However, engagement issues are vital to the overall success of commercial games, and deserve careful consideration. Heuristics developed by Federoff [12], Clanton[5], and Desurvire et al. [7] provide coverage of engagement and entertainment issues in games, and comprehensive inspections of game designs could be carried out using both their heuristics and ours.

CONCLUSION

Knowledgeable designers that have significant experience with games can identify many usability problems without the need for a set of heuristics. However, there are still many games that are released with serious usability problems, so we believe that more organized evaluation approaches, such as usability inspections using heuristics, still have value. This is one of the reasons for the success of Nielsen's heuristics—the design principles described in the heuristics are widely accepted in the HCI community, and are well known by most designers. However, good designers still violate the principles. Heuristic evaluation can play an important role because it forces a more formal inspection, where each issue is considered in turn, and it helps uncover issues that are not necessarily obvious otherwise.

The heuristics that we present in this paper provide a new way to adapt usability inspections for games, and allow designers to evaluate both mockups and functional prototypes. Unlike other game heuristics, ours focus specifically on game usability, and they are based on a structured analysis of usability problems from a large number of games. The heuristics allow people to evaluate game usability by applying design principles that are based on design trends seen in recent games, and that are generalized across major genres found in commercial games.

In our research, we used critical reviews of software to identify usability problems and to develop a set of design principles for video games. We believe that this general methodology is a new approach that can be used by researchers to understand design issues seen in other types of specialty software. We also believe that this method can be used in further studies of video games and that new heuristics can be developed on engagement and for multiplayer video games.

In the future, we will carry out an expanded evaluation of the heuristics. We will include more evaluators and games from several different genres.

ACKNOWLEDGEMENTS

We thank Carman Neustaedter, Carl Gutwin, Miguel Nacenta, David Flatla, and Regan Mandryk for their suggestions and feedback.

REFERENCES

1. Annett, J. and Duncan, K.D. Task analysis and training design. *Occupational Psychology* 12, (1967), 211-221.
2. Barr, P., Noble, J., and Biddle, R. Video game values: Human-computer interaction and games. In *Journal of Interacting with Computers*, 19, 2. (2007), 180-195.
3. Bias, R. Walkthroughs: Efficient collaborative testing. *IEEE Software* 8(5), (1991), 94-95.
4. Bias, R. The pluralistic usability walkthrough: coordinated empathies. In *Usability Inspection Methods*,

Nielsen, J. and Mack, R.L., Eds. John Wiley& Sons, New York, (1994), 65–78.

5. Clanton, C. An Interpreted Demonstration of Computer Game Design. In *Proc. CHI 1998*, 1-2.
6. Cockton, G., and Woolrych, A. Sale must end: Should discount methods be cleared off HCI's shelves? *interactions*, 9(5) 2002, pp. 13-18.
7. Desurvire, H., Caplan, M., and Toth, J. A. Using heuristics to evaluate the playability of games. *Ext. Abstracts CHI 2004*, 1509-1512.
8. Dulin, R. Review of Druuna: Morbus Gravis, http://www.gamespot.com/pc/adventure/druunamorbusgravis/review.html, 2002.
9. Dyck, J., Pinelle, D., Brown, B., and Gutwin, C. Learning from Games: HCI Design Innovations in Entertainment Software. *Proceedings of Graphics Interface* 2003, pp. 237-246.
10. Dykstra, D.J. *A Comparison of Heuristic Evaluation and Usability Testing: The Efficacy of a Domain-Specific Heuristic Checklist*. Ph.D. dissertation, Department of Industrial Engineering, Texas A&M University, 1993.
11. Entertainment Software Association, Essential Facts about the Computer and Video Game Industry 2007, http://www.theesa.com/archives/2007/07/2007_essential.php.
12. Federoff, M. Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games. *Indiana University Master of Science Thesis*, 2002.
13. Fullerton, T., Swain, C., Hoffman, S. *Game Design Workshop: Designing, Prototyping, and Playtesting Games*, CMP Books, 2004.
14. Goble, G. Review of Rugby 2004, http://www.gamespot.com/pc/sports/rugby2004/review.html, 2003.
15. Jeffries, R., Miller, J. R., Wharton, C., Uyeda, K.M. User interface evaluation in the real world: a comparison of four techniques. *Proc. CHI '91*, 119-124.
16. Johnson, C. G., and Jones, G. J. F., Effective Affective Communication in Virtual Environments. In Daniel Ballin (editor), *Second Workshop on Intelligent Virtual Agents*, (1999), 135-138.
17. Jørgensen, A. H., Marrying HCI/Usability and computer games: a preliminary look. In *Proc. of NordiCHI*, (2004), 393-396.
18. Karat, C., Campbell, R. L., and Fiegel, T. 1992. Comparison of empirical testing and walkthrough methods in user interface evaluation. In *Proc. CHI 1992*, 397-404.
19. Lewis, C., Polson, P., Wharton, C., and Rieman, J. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *Proc. CHI 1990*, 235-242.
20. Malone, T.W., Heuristics for designing enjoyable user interfaces: Lessons from computer games. In John C. Thomas and M. L. Schneider (editors), *Human Factors in Computing Systems*, Norwood, NJ: Ablex Publishing Corporation, 1982.
21. Mueller, G. Review of Eragon, http://www.gamespot.com/pc/action/eragon/review.html, 2006.
22. Mueller, G. Review of Mage Knight Apocalypse, http://www.gamespot.com/pc/rpg/mageknightapocalypse/review.html, 2006.
23. Nielsen, J. Enhancing the Explanatory Power of Usability Heuristics. In *Proc. CHI 1994*, 152-158.
24. Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY, 1994.
25. Nielsen, J. How to Conduct a Heuristic Evaluation. www.useit.com/papers/heuristic/heuristic_evaluation.html, 2007.
26. Nielsen, J. Severity Ratings for Usability Problems. http://www.useit.com/papers/heuristic/severityrating.html, 2007.
27. Ocampo, J. Review of Vietcong 2, http://www.gamespot.com/pc/action/vietcong2/review.html, 2005.
28. Parker, S. Review of Pax Romana, http://www.gamespot.com/pc/strategy/paxromana/review.html, 2004.
29. Polson, P., Lewis, C., Rieman, J., and Wharton, C. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies* 36, (1992), 741-73.
30. Shepherd, A. Analysis and training in information technology tasks. In *Task Analysis for Human-Computer Interaction*, Diaper, D., Ed. Ellis Horwood, Chichester, (1989), 15-55.
31. Todd, B. Review of Paradise Cracked, http://www.gamespot.com/pc/rpg/paradisecracked/ review.html, 2003.
32. Todd, B. Review of Total Extreme Wrestling 2004, http://www.gamespot.com/pc/sports/totalextremewarfare/review.html, 2004.
33. Todd, B. Review of DMZ: North Korea, http://www.gamespot.com/pc/action/dmznorthkorea/review.html, 2007.
34. Todd, B. Review of Genesis Rising: The Universal Crusade, http://www.gamespot.com/pc/strategy/genesisrising/review.html, 2007.
35. Todd, B. Review of Secrets of the Ark: A Broken Sword Game, http://www.gamespot.com/pc/adventure/brokensword4/review.html, 2007.
36. Wolpaw, E. Review of Soldner: Secret Wars, http://www.gamespot.com/pc/action/soldnersecretwars/review.html, 2004.