GamiCAD: A Gamified Tutorial System For First Time AutoCAD Users

Wei Li, Tovi Grossman, and George Fitzmaurice
Autodesk Research
firstname.lastname@Autodesk.com

ABSTRACT
We present GamiCAD, a *gamified* in-product, interactive tutorial system for first time AutoCAD users. We introduce a *software event driven finite state machine* to model a user's progress through a tutorial, which allows the system to provide real-time feedback and recognize success and failures. GamiCAD provides extensive real-time visual and audio feedback that has not been explored before in the context of software tutorials. We perform an empirical evaluation of GamiCAD, comparing it to an equivalent in-product tutorial system without the gamified components. In an evaluation, users using the gamified system reported higher subjective engagement levels and performed a set of testing tasks faster with a higher completion ratio.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

Keywords: Game, Tutorial, Learning

INTRODUCTION
Today's software applications can contain hundreds or thousands of features [27], making the process of learning to use the software difficult for new users [20]. A long line of HCI literature has investigated the problems associated with learning and many types of learning aids have been proposed. Common formats include online help [4, 5, 41], discussion boards [16, 39], and video assistance [19, 33].

For new users, documentation may not be effective. Carroll suggests a user has a *production bias* which "reduces their motivation to spend any time just learning about the system" [4]. A more promising approach may be to provide interactive tutorials, where users get to learn by accomplishing tasks in the application itself [2, 14, 17, 24].

While many types of interactive tutorials exist, they may lack a level of engagement that will provide enough motivation for the user to follow through with required or recommended lessons.

One technique to increase engagement levels, which recently has become commonly used in many domains, is gamification [10]. Gamification is commonly defined as using elements of video games in non-game situations to enhance user experience and engagement levels with a product [38, 36]. Gamification has been rapidly growing in popularity, with workshops [9] and entire conferences[1] now being dedicated to the topic.

Given its current popularity, it is reasonable to consider the appropriateness of gamification for software tutorials. Some previous commercial systems have developed tutorial systems that contain some gamified elements[2] [3]. However, little research literature exists on this topic [12]. In particular, it is unclear if there are any empirical advantages of a gamified software tutorial compared to a non-gamified counterpart [25, 31]. Without data to refer to, software designers must make uninformed decisions to develop gamified tutorials, without an awareness of the benefits or tradeoffs.

In this paper, we investigate the use of gamified tutorials to help new users both learn and improve their performance with a software system. Our work is composed of two main contributions:

First, we present GamiCAD, a gamified tutorial system for AutoCAD[4]. GamiCAD builds upon the recent work of Dong et al's Jigsaw system [12], introducing several novel features. In particular, a *software event driven finite state machine* is used to model a user's progress through a tutorial, which allows the system to provide real-time feedback and recognize success and failures. In addition, GamiCAD provides extensive real-time visual and audio feedback that has not been explored before in the context of software tutorials.

Our second core contribution is an empirical evaluation of GamiCAD, comparing it to an equivalent in-product tutorial system without the gamified components. Our evaluation shows that not only does a gamified tutorial increase subjective engagement and enjoyment levels, it can also improve learning. In particular, users completed 10% more testing tasks and were 20%-76% faster after using the tutorial with the gamified components.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
UIST '12, October 7–10, 2012, Cambridge, Massachusetts, USA.
Copyright 2012 ACM 978-1-4503-1580-7/12/10...$15.00.

[1] The Gamification Summit
http://www.gsummit.com (retrieved April 13, 2012)
[2] http://success.adobe.com/microsites/levelup/ (retrieved April 13, 2012)
[3] http://www.ribbonhero.com/ (retrieved April 13, 2012)
[4] http://usa.autodesk.com/autocad/

RELATED WORK

In this section, we discuss previous research in the areas of software learning, tutorial systems, gamification, and software gamification.

Software Learning

Software learnability [11, 30] is an important research problem that has been prevalent since the beginnings of HCI. Important early work includes Carrol et al.'s investigations of learning within word processing tools [4, 5, 26].

A recent survey of software learnability is provided by Grossman et al [20]. Their taxonomy of learnability distinguished between *initial learnability*, the initial performance with a system and *extended learnability*, the change in performance over time. In targeting new users, GamiCAD focuses mostly on initial learnability.

In the psychology literature, the Dreyfus model proposes that a learner passes through five distinct stages during the skill acquisition (Figure 1) [13]. According to this model, experts increase their capability of executing tasks at a subconscious level without guidance or instructions. This indicates that tutorial systems for novice users should provide a high level of guidance and feedback. Using game mechanics may be one way to do so.

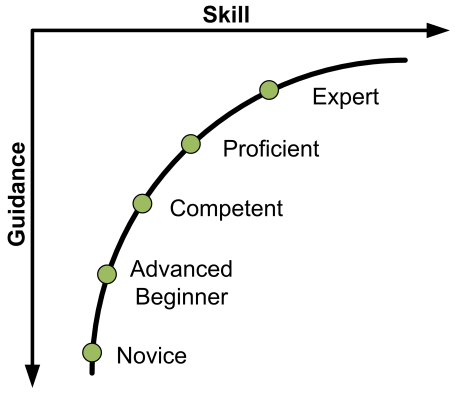


Figure 1. Dreyfus model of skill acquisition.

Tutorial Systems

Delivering effective help material and documentation can be a challenging task, because of the active user's *production bias* [4]. Early efforts to improve software documentation included minimalist and task-centered help [4, 5, 41].

An alternative to static help systems is animated assistance [19, 32, 33]. Such materials typically enforce a passive learning process, or force users to work at the pace of the video [17]. Pause-and-Play is an interesting new system that aims to improve upon this limitation, by automatically pacing the video based on a user progress [33].

Alternatively, tutorial systems can also be integrated within the software application itself, providing a rich interactive learning experience [2, 14, 24, 33, 35]. For example, Stencils-based tutorials [24] guide a user to perform the correct step by overlaying a stencil with a hole to click-through. The level of engagement they provide could potentially be increased even more by integrating gaming components.

In particular, Fernquist et al. suggest that tutorial experiences should be *as compelling as possible*, so that users stay engaged [14]. One guideline for creating such experiences is Csikszentmihalyi's concept of flow [8], which describes a user's state of "optimal experience". Qualifying factors for achieving flow include: perceptions of clear goals, immediate feedback, and matched skills and challenges [6]. GamiCAD is designed with these factors in mind.

Gamification

Gamification is the use of design elements, which are characteristic for games, in non-game contexts [10]. Gamification has been used in numerous domains, such as work environments [15, 36], the design industry [23], marketing [43] and, particularly, educations [18, 34, 37].

One main motivation for gamifying a learning experience is to make the experience more engaging for the user [38]. However, there is "a lack of empirical evidence" that the use of gamification is effective in learning contexts [25]. Previous studies of gamified learning environments [7, 31] have been confounded in that "positive educational outcomes can be attributed to the instructional design of the educational programs and not to games as a medium" [25]. Our work contributes empirical data from a controlled experiment, comparing a gamified and non-gamified learning experience with equivalent instructional design.

Software Gamification

There are several games designed for learning software applications. Commercial examples include Microsoft Ribbon Hero[1], and the now expired Adobe LevelUp[2]. In the research literature Jigsaw [12] is also related to our work.

In these examples, users need to finish small tasks to collect points and badges. Jigsaw promotes discovery-based learning, where users can explore multiple strategies for completing a task. While Dong et al. found that users enjoyed the Jigsaw system [12]; none of these systems have been empirically compared to non-gamified counterparts.

Furthermore, in these examples, the extent of the gamified components is limited. For example, Ribbon Hero provides a game like menu system for choosing levels, but once a task is chosen, the system runs like a typical in-product tutorial. In addition, these previous systems give users exposure to a broad range of tools, but users do not necessarily learn the best way to use them.

GamiCAD builds upon these systems and increases the extent of which gamified mechanics are used during individual levels. Furthermore, GamiCAD exposes users to multiple ways to use individual tools, so users learn how to use tools in both basic and advanced ways.

Summary

A large amount of literature exists on software learning and tutorial systems. Results from this literature indicate promise in providing interactive tutorial systems, and using gam-

[1] http://www.ribbonhero.com/ (retrieved April 13, 2012)
[2] http://success.adobe.com/microsites/levelup/ (retrieved April 13, 2012)

ified mechanics to achieve high engagement levels. While some gamified software tutorial systems exist, the extent to which they are gamified is limited, and no empirical evaluations have been performed. In general, there is a lack of empirical evidence that the use of gamification is effective in learning contexts.

GAMIFICATION ELEMENTS

In our work we aim to push the envelope in terms of the extent of gamified mechanics which are included in our tutorial system. For a complete discussion of gaming elements, we direct the reader to existing surveys [29, 22, 28, 40]. Here, we specifically discuss the gaming elements most appropriate for use in a software tutorial system. Each of these elements will be integrated into GamiCAD.

Fantasy

According to Maloney [29], "Fantasy is probably the most important feature of computer games that can be usefully included in other user interfaces." Fantasy evokes images of objects or situations that aren't actually present. This can make the experience more emotionally appealing to users.

Clear Goals

An aspect of games that has been strongly correlated with game popularity is the presence of clear goals [29, 40]. Clear goals are important so that players understand the task which they are trying to complete, so that they will stay engaged with the system [8].

Feedback and Guidance

Typical video games provide immediate feedback to users based on their progress and goals. Providing such feedback can also increase users' engagement levels [6, 8, 40]. In addition, if a user makes a mistake, they can become lost and disoriented [12]. As such, the system should lead users to recovering from error states [40].

Progressive Disclosure

A game helps players to continuously increase their skills by progressive disclosure of both knowledge *and* challenge [40]. This will help ensure that the challenges in the game match the player's skill levels [8]. For example, the system could provide more strict guidance to a novice user or more freedom to proficient learners.

Time Pressure

Time pressure is commonly considered an important and effective aspect of games. Adding time pressure is effective as it establishes clear and challenging goals [42, 29, 36].

Rewards

A common approach to directly motivate users is to provide points and rewards [36, 42]. Performance feedback can be provided by assigning scores after the completion of a game level, which can facilitate a user's progress assessment. Another type of reward mechanism is to unlock new levels based on the user's achievements.

Stimuli

To ensure high engagement levels, games should provide stimuli that is worth attending to [1, 40]. High-quality stimuli can be provided in a number of different formats, such as unique animations, sounds, and appearances.

GAMICAD SYSTEM AND IMPLEMENTATION

We now described GamiCAD, a new gamified software tutorial system, designed for new AutoCAD users. GamiCAD consists of several components which interact with one another (Figure 2). The Mission Console is used to select levels to complete. The in-product task panel is used to provide instructions and feedback during individual levels. For accurate feedback to be provided, the in-product task panel requires knowledge of the tasks, knowledge of how the tools work, and also needs to be able to monitor events and data from the AutoCAD environment. The system is implemented as an AutoCAD plugin using the AutoCAD ObjectArx SDK.

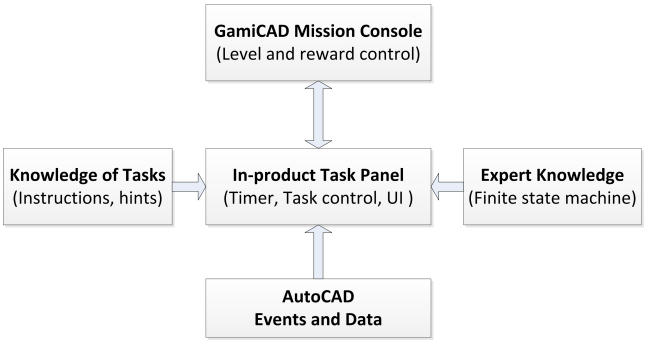


Figure 2. GamiCAD Architecture.

System Goals and Contributions

Our ultimate motivation is to provide an *effective* and *engaging* learning system for new users. In terms of *effectiveness*, we aim to help users learn basic *and* efficient ways to accomplish tasks. In terms of *engagement*, we aim to provide an environment that users will enjoy, and be motivated to repeat tasks to improve their own performance levels.

Compared with previous systems, GamiCAD contributes several novel features.

- GamiCAD provides learning tasks in two level formats. Highly structured guided tasks are provided for first-time users, but as users progress, they are rewarded with "arcade style" bonus levels.
- GamiCAD stimulates repetition of tasks through level unlocking mechanics. GamiCAD uses approaches to make repetitive tasks challenging and interesting, so that users can refine and improve their skills before advancing to the next level.
- GamiCAD provides immediate feedback to help users recover from errors. An event driven finite state machine is used to model possible states of a task. This allows the system to guide users back from incorrect states, preventing user frustration.
- GamiCAD provides multiple detailed lessons for similar tasks, so that users can be exposed to alternative task flows to accomplish a task. This allows users to learn how to not only use a command, but use it efficiently.

Target Application and Content

We study our gamified learning system in the context of AutoCAD, a software application for both 2D and 3D de-

sign. AutoCAD is considered challenging for novices, making it a good candidate for our efforts.

We focus the learning content of GamiCAD on two fundamental CAD commands, LINE and TRIM. Both LINE and TRIM tools have complex subtleties and are difficult for novices to learn, yet fundamental for achieving higher level expertise. The lessons are delivered in a set of comprehensive in-product learning modules covering multiple options and parameters for working with these functions. In addition, the learning modules also provide basic skills of using AutoCAD, including command line input, document navigation skills and object selection.

Command Flow Modeling

Many AutoCAD commands are designed to follow a specific action sequence. For example, with the TRIM command, a set of cutting edges has to be selected before selecting the objects to trim. If the system lacks an awareness of a user's progress through these steps, adequate real-time feedback cannot be provided, which can cause users to become lost and confused [10]. In GamiCAD we use an event driven finite state machine to monitor a user's progress, and help the user actively recover from incorrect states.

Finite state machines (FSMs) have been broadly used in video games. Popular games like Quake II and Warcraft III take advantage of FSM systems to control their levels, environmental conditions and computer-controlled characters.

In our FSM, the transition from one state to another is triggered by software events or messages. Our FSM is also enhanced with parameters, for example, to keep track of the number of points in a line. Most software applications can formally abstract their features or specifications using event driven FSMs [21]. In AutoCAD, the LINE and TRIM command can be represented by the FSMs in Figure 3.

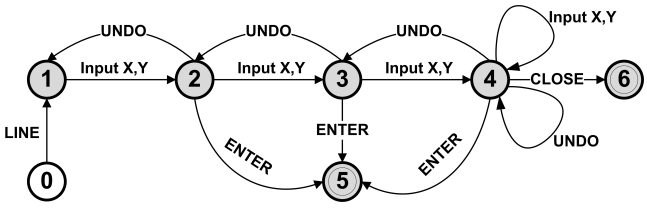


Figure 3. FSMs for LINE command.

We define a *command flow* by a specific path to reach a final state. For example, the *command flow* for drawing a triangle with the LINE command is [LINE]→[Input X,Y]→[Input X,Y]→[Input X,Y]→[CLOSE]. This will trigger the transitions of the FSM from states 0 to 6.

In GamiCAD, once the user begins a task, the finite state machine begins to parse the user's action stream and match it with the associated command flow, so their progress throughout the task can be accurately monitored.

This is particularly helpful for guiding users to recover from error states. When the user makes an error, GamiCAD searches the FSM to find the shortest path to return to the last correct state. We use Figure 4 as an example, where a user's task is to draw a single line segment. If the user instead draws two line segments, using three points, the FSM will incorrectly be in state 4. To guide the user back to the *command flow* path, the system can prompt the user to return to state 3 by typing UNDO.

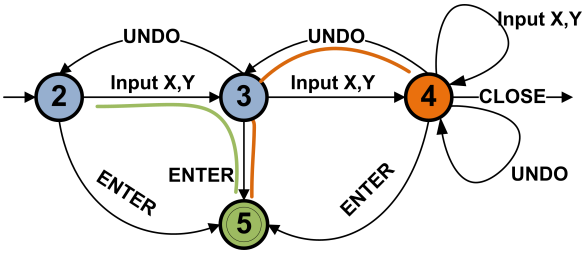


Figure 4. Error recovery. User currently in state 4. Correct flow is green line connecting states 2→3→5. Orange line shows the error recovery path 4→3→5.

GAMICAD DESIGN

We now discuss the design of GamiCAD, highlighting which design choices utilize the previously described Gamification Elements, shown in italicized brackets.

The Mission Console

The *Mission Console* is the top-level interface used to navigate through the game and its levels. The mission console is displayed in an external window.

We choose to use the design intensive Apollo program as the backstory for GamiCAD (*Fantasy)*. Users are introduced to a story in an initial screen when they first launch the game, which explains that the user will be responsible for helping NASA build components of a spacecraft.

In addition to the initial start-up screen, the Mission Console contains four Mission Pages. Users can navigate through the mission console with Forward and Back arrows. Initially the user can only access the first mission, with missions 2-4 being locked.

The four missions are sorted by topic (Figure 5). Each mission contains multiple levels. The individual levels take one of three mini-game formats. SkillBuilder provides a guided task level, and QuickClick and TrimMaster provide arcade style bonus levels.

The Bonus Missions are designed to be rewarding experiences (*Rewards*). To unlock the next mission, users must achieve at least 4 stars in each level of the current mission (*Progressive Disclosure*). This can encourage learning through repetition.

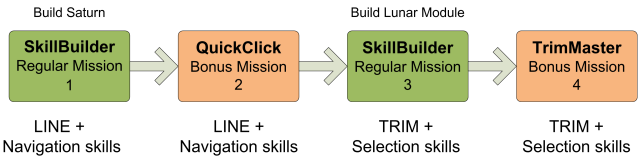


Figure 5. Four Mission Design Topics.

Mission Pages

Each Mission has a dedicated mission page in the Mission Console (Figure 6). Each mission page lists the individual levels within that mission. The levels are carefully ordered

to first expose users to the basic functionality, and then more advanced options and features. Users must complete the tasks in order (*Progressive Disclosure*). Clicking on a level launches the level inside of AutoCAD.

For each completed level, the user's best score and highest number of stars achieved are displayed (*Feedback and Guidance*). An icon is used to indicate the levels which are still locked.

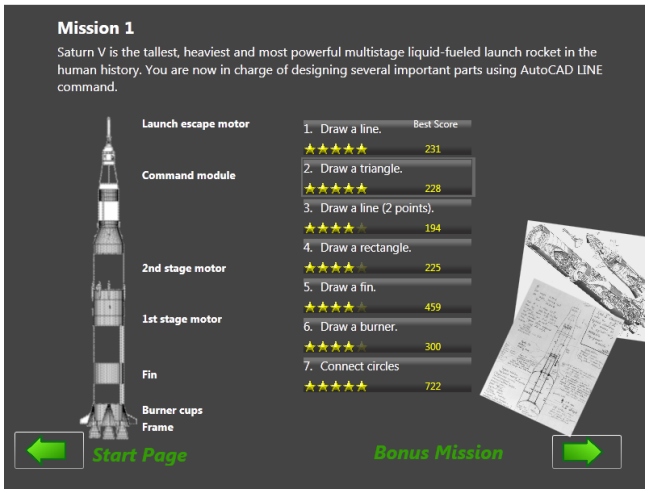


Figure 6. Mission Page with 7 completed levels.

SkillBuilder

SkillBuilder provides a structured task with step-by-step guidance to learn a new skill. A palette in AutoCAD is used to show the instructions and information about the task (Figure 7). An image and instructions describing the goal of the task are displayed at the top of the palette (*Clear Goals*) (Figure 7A). Users click a "Start" button to begin a level.

Step-by-step instructions

Under the task description, an area is reserved to show step-by-step instructions. The step-by-step instructions are hidden by default, so users can try to complete the task without assistance. A "show hints" button can be clicked to reveal the hints. When visible, each step contains a simple and clear instruction (*Clear Goals*) (Figure 7B).

After the completion of a step, the FSM will trigger a step complete message, and a green check mark is displayed next to that step (*Feedback and Guidance*). If the user makes an error, a message is displayed directing the user to the required corrective action (*Feedback and Guidance*). If a user diverges from the required command path by more than two steps, a message informs the user that they must start over. This is equivalent to dying/failing a level in a video game, and prevents users from becoming too lost.

Some steps have a hint button, which provides an animated visual aid to help the user locate an associated component or tool from the AutoCAD UI (*Stimuli*) (Figure 7C).

Speed bonus

At the bottom of the palette, a timing progress bar and Speed Bonus Points are displayed (*Time Pressure*). The Speed Bonus Points start at 150, and begin to continually decrease immediately after the task is activated, decreasing by 1 every 0.2 seconds until it reaches 0. The progress bar also reflects this time pressure by shortening its length and changing its color from green to red (Figure 7D).

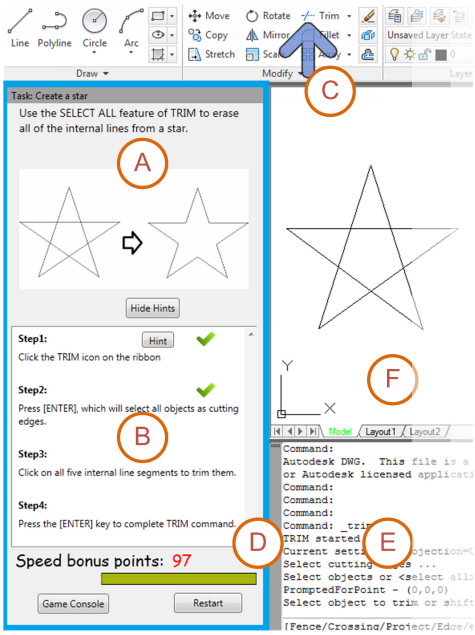


Figure 7. Task panel for SkillBuilder (A) task description; (B) step-by-step instructions; (C) UI element visual aid; (D) speed bonus bar; (E) AutoCAD command line; (F) AutoCAD drawing area.

Scoring

When the user completes a level, a display provides feedback on how they performed *(Rewards)*. We display a total score, bonus points, and 0 to 5 stars (Figure 8). A customized message is also displayed based on their performance.

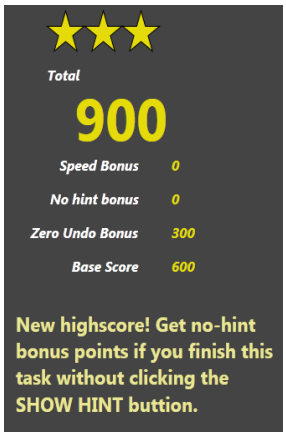


Figure 8. SkillBuilder score board.

The score and stars earned per task are computed based on four factors:

$$total\ score = base\ score + speed\ bonus + no\ hint\ bonus + zero\ undo\ bonus$$

The base score is pre-assigned to each task according to its complexity, varying from (50 to 600). The no hint bonus is either 0, if the user viewed the hints, or 50% of the base score, if they did not. Similarly, the zero undo bonus is either 0, if the user used an undo operation, or 50% of the base score if they did not. The number of stars assigned is based on the ratio of the current score over the total score.

Bonus Mission: QuickClick Game

Once all tasks for the LINE tool are completed in Mission 1, users are rewarded with an arcade-style QuickClick bonus Mission, where they can apply their newly obtained skills.

QuickClick is a connect-the-dots tracing game that requires efficient use of the LINE command and navigation tools to complete (Figure 9). The player is required to navigate in the drawing area and connect a series of small circles by using the LINE command. Only one dot is displayed at a time. When the user successfully clicks within the displayed dot, the next dot is revealed.

The dots are extremely small, so the user must zoom in to be able to click within their boundaries. A larger blue circle is displayed around the next goal dot, so that the user knows where the target is. The trick of the game is to quickly locate the next dot, zoom in to it, and then zoom out or pan to the next target location. This is a common and important behavior for efficient use of AutoCAD.

For Mission 2 there are 3 QuickClick levels. In each level, the dots get smaller making the task more difficult, as more zooming is required (*Progressive Disclosure*).

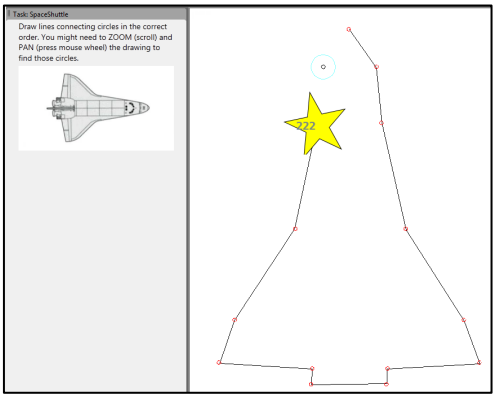


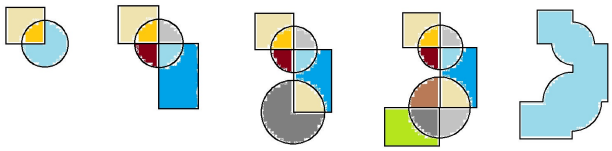
Figure 9. QuickClick bonus game.

Scoring

QuickClick provides feedback every time a user successfully clicks a dot (*Feedback and Guidance*). A step score is calculated based on how fast the user clicked the dot (*Time Pressure*). An overlay displays the step score and a spinning star animation when the user successfully clicks on a dot (*Stimuli*). Higher scores generate bigger stars. The total score of the task is the sum of each step score.

Bonus Mission: TrimMaster Game

TrimMaster is the other bonus game that users can play after completing the TRIM tool levels in Mission 3. In TrimMaster, players are required to remove the internal boundaries among geometries and maximize the internal area of merged geometries (Figure 10). Similar to puzzle-arcade games like Tetris, TrimMaster keeps adding new geometry, so that the level gets more difficult as it continues. Figure 10 shows the progression of levels. Each time the user executes the TRIM command, a new object is added to the scene, until a predetermined number of objects have been added. If the player performs perfectly, the final object will have no internal lines.



(1) (2) (3) (4) (5)

Figure 10. (1-4) shows TrimMaster getting harder from round 1 to round 4. (5) shows the final geometry if the player successfully merged all geometries

Scoring

As in the QuickClick bonus game, no step-by-step instructions are provided, but feedback is provided each time a user executes the TRIM command (*Feedback*). The same animated visual overlay from QuickClick is displayed, each time a TRIM is completed (*Stimuli*). The score is based on the largest internal area in the geometry, and the time taken to complete the step (*Time Pressure*). The higher the score is, the bigger the star will be. The total level score is the sum of these scores.

Music and Sound Effects

To provide a more immersive experience, GamiCAD uses real time sound effects and dynamic background music (*Stimuli*). Sound effects are played throughout the game. For example, a sound effect is played: when a user changes pages in the mission console; when the user completes each step in the bonus mission levels; and when a level is completed and the score is displayed.

In addition, background music is played throughout the entire game, and is updated dynamically based on the user's progress and the game's state. Each mission has its own background music, with the bonus levels having faster paced music. While viewing the Mission console, the background music is more ambient. The Mission console background music updates each time a user completes an entire mission, to strengthen the user's sense of progress. The background music was taken from the award winning Assassin's Creed II video game original sound track[1].

We direct the reader to our video figure to hear the sound effects and background music.

USER STUDY

While informal subjective evaluations of gamified software tutorials have been previously conducted, there is a lack of empirical evidence studying the differences between gami-

[1] http://www.amazon.com/Assassins-Creed-Original-Game-Soundtrack/dp/B002TURE9E

fication and non-gamified counterparts in the learning domain [10, 31]. Our goal of this comparative study is to answer two main questions: 1) Does the inclusion of gamified elements provide a more engaging learning experience? 2) Does a gamified software tutorial have a positive impact on learning?

We hypothesize that the gamified components will increase user engagement levels, which will in turn, also positively impact learning.

Conditions

To truly understand the impact of gamification, GamiCAD needs to be compared to a system with equivalent instructional design. As such, we built an in-product tutorial system that contains the exact same tasks and instructions but without the gamified mechanic. By comparing these gamified vs. non-gamified learning systems, we are able to conduct a controlled quantitative and qualitative analysis of the impact of gamifying software tutorials.

As such, there were two conditions in our study, associated with the two tutorial systems. The first tutorial system was *GamiCAD*. The second system, *TutorCAD*, was the equivalent version of the tutorial system, where only the gamified features were removed (Figure 11). The features which were removed were: the mission console (replaced with a tutorial list panel, Figure 11a), scoring system, animated feedback, music, sound effects, timers, and mission/level unlocking mechanism. Users were asked to complete tasks following the same order in the game. *TutorCAD* still provides many of the interactive features possessed by GamiCAD, such as the step-by-step instructions, step completion feedback, error recovery guidance, and UI element visual aids (Figure 11b).

Once a tutorial is selected in TutorCAD, a task panel will be opened in AutoCAD. The task panel looks almost the same as GamiCAD's SkillBuilder task panel, but the speed bonus points, progress bar and the game console buttons have been removed. Bonus levels were included, but presented as regular tasks in the tutorial list panel. The actual tasks were identical, with the exception of the TrimMaster tasks. Instead of presenting geometry one object at a time, the final round is present at the beginning of the task.

Apparatus

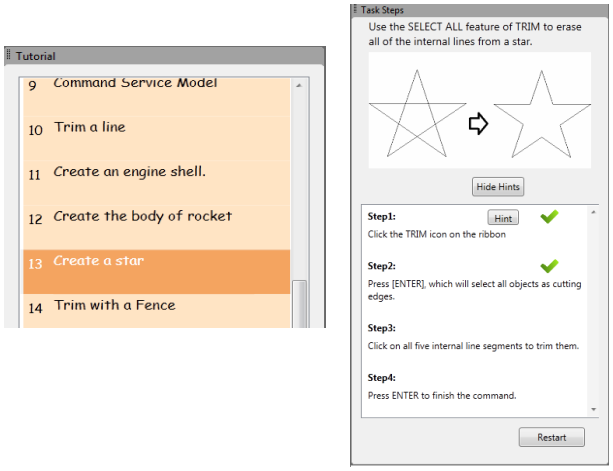
Our user study was conducted using a HP Z600 workstation with AutoCAD 2012 installed and a 24 inch monitor with a resolution of 1920×1200. A standard mouse, keyboard, and set of speakers were connected to the workstation.

Participants

Fourteen participants (10 female) between the ages of 19 and 62 without any AutoCAD experience were recruited via online postings. Five participants were regular gamers, eight played games occasionally. Only one participant reported not playing any video games at all.

Procedure and Design

A within participant design was used for the study, where task order was fixed but condition order was counterbalanced. All participants completed LINE tasks first and TRIM tasks second. Half the participants used GamiCAD for LINE (first) and the other half used TutorCAD for LINE (first). We fixed the order of the tasks to maintain progressive disclosure of the relevant AutoCAD skills. In the *GamiCAD* condition, users needed to achieve 4 or 5 stars in all regular levels to unlock the bonus levels. In the *TutorCAD* condition, subjects were told that they could repeat tasks, but there was no unlocking mechanism.



(a) (b)

Figure 11. TutorCAD a) Tutorial list. b) Task panel.

After completing each block, a set of testing tasks were administered to evaluate the user's abilities to complete related tasks without system assistance. We designed 6 tasks, two for "LINE and navigations" and four for "TRIM and selections." Table 1 shows all six test tasks.

Table 1. Test tasks.

ID	Drawing	Task	Knowledge
1		Draw the pentagon and star (coordinates were provided)	Line, coordinates
2		Draw the rocket by connecting circles	Pan, Zoom, Line
3		Remove lines inside small rectangle	Trim, crossing
4		Remove lines inside ellipse	Trim, fence
5		Remove lines between 3 shapes and rectangle	Trim, crossing
6		Remove lines inside star	Trim, fence

A moderator was present during the study to take observations. The moderator provided assistance if the user failed to complete a task after 8 minutes (480 seconds). The mod-

erator recorded anytime assistance was provided. Upon completion of the study, a questionnaire was administered to collect qualitative data and subjective feedback.

Quantitative Results

The main quantitative measures for the study were the completion time and completion rate of the testing tasks that were presented at the end of each block.

Analyzing task completion times can provide an indication of how well users learned the concepts of the tools during the tutorials. In calculating testing task completion times, we assigned the maximum duration (480s) to any task which the user did not complete. The average task completion times were 111s for *GamiCAD* and 261s for *TutorCAD*. Figure 12 shows the testing completion time for each of the six tasks. It can be seen that the times were faster for *GamiCAD* in all of the tasks. A paired sample t-test showed that this difference was significant for testing tasks 1, 3, 4, and 6 ($p<.05$).

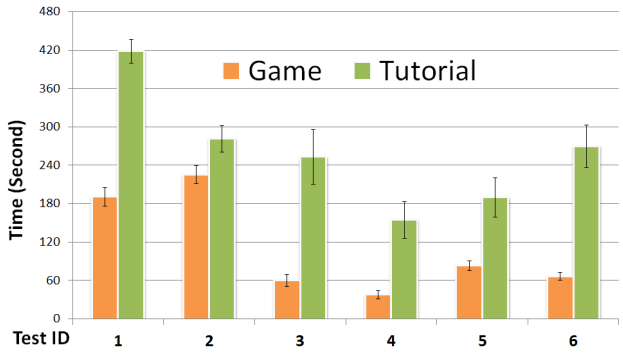


Figure 12. Completion times for 6 testing tasks.

We also performed an analysis of task completion rates. We considered a task to be completed if the user was able to successfully finish the task within the allotted 8 minutes. The testing task completion rate for *GamiCAD* was 100%. For *TutorCAD*, 90.5% of tasks were completed within the allotted time.

These results indicate that users hadn't fully understood the concepts after using the *TutorCAD* system. We believe the performance gains were in large part due to unlocking mechanism present in the *GamiCAD* condition, requiring user's to repeat levels to improve their performance. On average, tasks were repeated 7.8 times for the *GamiCAD* tasks and 1.1 times for the *TutorCAD* tasks.

As such, the above results should be considered with a degree of caution, due to unbalanced learning times. However, the results we find are important: without enforcing a threshold level of performance, users may struggle through a tutorial without adequately learning the core concepts, as evident by the lower completion rates and higher completion time in the *TutorCAD* condition.

Qualitative Results

Figure 13 shows the results of our subjective questionnaire. In general, participants found the game condition more enjoyable, fun, engaging and effective. It is interesting to note that three participants strongly preferred the standard tutorial style condition. Based on their comments, one participant did not like the fast paced music. The other two felt too much time pressure during the game.

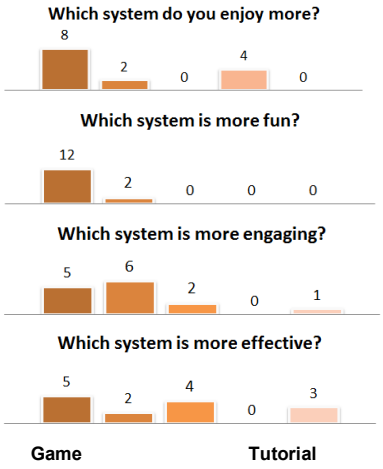
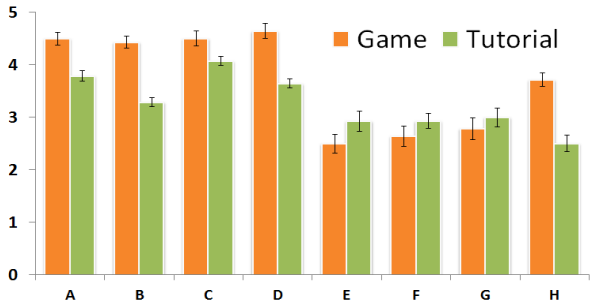


Figure 13. Results from subjective questionnaires comparing the game and the tutorial systems.

In the post-study questionnaire, participants were asked to rate 8 questions on a (strongly disagree) 1 to 5 (strongly agree) Likert scale for each system. We ran Mann-Whitney's U tests to evaluate their difference. The responses to question A, B, D, and H were significantly different ($p<0.05$).



A=I enjoyed using this system
B=It makes completing tasks fun
C=It is an effective learning tool
D=It is an engaging experience
E=Completing tasks was frustrating
F=Completing tasks was difficult
G=Had to work hard to complete tasks
H=Felt rushed when completing tasks

Figure 14. Subjective results comparing the game and the tutorial systems.

Engagement

In general, users commented that the game condition was fun and engaging. P1: "*Game [is] more fun – [it] didn't seem as hard. You tend to forget you're learning*" P2: "*The game was a lot more engaging and it made me excited to learn the task.*" P3: "*The game was definitely more engaging. You had to work hard in order to go to the next level.*" In contrast, the tutorial system received some negative feedback. P11: "*The tutorial was somewhat boring compared to the game.*" P3: "*Not engaging. Feels like it was just testing my skills to follow instructions in order.*" P6: "*Tutorial [feels] dry, hard to do many lessons in one sitting.*"

Challenge

We observed many users wanting to challenge themselves and repeat tasks to achieve higher scores or levels. P1: "*Completing against myself is fun*." P2: "*I liked that 'mission' format of the game and the element of challenge (timer, points, etc)*." P10: "*I am very competitive myself. I enjoyed adding the speed component even though it challenged me more*." P11: "*Good level of stress involved to keep me engaged (i.e. timed exercises)*."

There were participants who disliked the pressure and stress of the game. P5: "*I felt more pressured to do well*." P6: "*Time pressure could be exhausting after a few sessions* (suggesting a slower tempo)", P13: "*However, the game was fun but stressful with the clock, and the tutorial was a little boring but definitely made this a[n] easy learning experience*."

Feedback

Almost every participant commented on the gaming feedback. P4: "*[I] like [the] animation and score system*." P5: "*[I like] the fact it animated a task*." P12: "*game [is] fun, colorful, engaging, and appealing to the eye*." We did observe a diversity of preference about the game feedback. For example, P2 commented that the speed bonus was too fast. P1 wrote "*[game] graphics are a little distracting*." P4, P6 and P11 commented that the "*music was distracting*" or "*too rushing*."

Some participants did like the simplicity of the tutorial. P13: "*Tutorial: easier to use, clear interface, no rush/clock, simple to follow, quite effective at getting job done*."

DISCUSSION

In both the GamiCAD and TutorCAD learning conditions, users were able to have a sense of accomplishment in completing tasks and progressing through levels of a structured system. This is often not the case when users try to learn by only reading instructional content or reference material outside of an interactive in-product tutorial system.

Using the within-subject design, participants were able to compare both systems, and required fewer participants [15]. The risk of good-participant effects exists but quantitative results also indicate benefits of the GamiCAD condition. In our study, we also notice some ordering effects in users' AutoCAD familiarity. For example, several participants had trouble using the command line panel in the first system, but none of them had any problem in the second system. However, we believe the counter-balancing mitigated this ordering effect.

Within the consumer game industry, missions and levels are designed by professional level designers who use game level editors to create new environments and scenarios. For a CAD learning system, potentially the best level designers are the actual software users. Thus, GamiCAD is able to load AutoCAD drawing files and then automatically generate new levels for QuickClick and TrimMaster. We used these authoring features to create all of the levels for both games within a couple of hours.

CONCLUSIONS AND FUTURE WORK

GamiCAD provides guided training tasks for first-time users utilizing an "arcade style" gamification of the tasks to facilitate learning. By developing challenging levels, providing motivational feedback and encouraging repetitive task completions through score improvement, we are able to make improvements to the learning process compared to a standard in-product tutorial system.

In addition, the introduction of a finite-state system offers customized feedback and error-recovery messages. We believe this approach could serve to correct mistakes more quickly before the user gets frustrated by a lack of progress in learning. In the future, we would like to explore methods to model a user's task progress given uncertainty and implicit user observations.

Our user study comparing a gamified with a non-gamified in-product interactive tutorial system showed that the gamified system produced significantly faster test task completion times. Users also felt that the game condition was more enjoyable, fun, engaging and effective.

In the future, we wish to continue to study additional aspects of game design to facilitate guided instructions. We are also curious to determine if there is a minimal level of gamification possible that still yields similar performance and learning benefits. For example, in our GamiCAD system, can we remove the gaming music without reducing the performance benefits?

It would be interesting to conduct a longitudinal study on the operating speed of users trained on a gaming system compared to a standard in-product training system. Are game-trained users faster than standard-trained users outside of the learning environment?

Finally, since GamiCAD provides an authoring capability for AutoCAD users, it would be interesting to find out if we can leverage the existing AutoCAD users' community to create new gamified tutorials.

REFERENCES

1. Andersen, E., Liu, Y., Snider, R., Szeto, R., and Popović, Z. (2011). Placing a value on aesthetics in online casual games. *ACM CHI, 1275-1278.*
2. Bergman, L., Castelli, V., Lau, T., and Oblinger, D. (2005). DocWizards: a system for authoring follow-me documentation wizards. *ACM UIST. 191-200.*
3. Card, S. K., Moran, T. P., and Newell, A. (1983). *The Psychology of Human-Computer Interaction*, CRC Press.
4. Carroll, J. M., and Rosson, M. B. (1987). *Paradox of the active user*. *Interfacing Thought Cognitive Aspects of Human-Computer Interaction*. MIT Press, 80-111.
5. Carroll, J. M. (1990). *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill.* MIT Press.
6. Chen, H., Wigand, R.T. and Nilan, M.S. (1999). Optimal experience of web activities. *Computers in human behavior.* 15(5), 585-608.

7. Clark, R.E. (1983) Reconsidering research on learning from media. *Review of Educational Research, 53, 4,* 445-459.
8. Csikszentmihalyi, M. (1988). The flow experience and its significance for human psychology. *Optimal Experience: Psychological Studies of Flow in Consciousness*.15-35.
9. Deterding, S., Sicart, M., Nacke, L., O'Hara, K., and Dixon, D. (2011). *CHI 2011 Workshop Gamification: Using Game Design Elements in Non-Game Contexts.*
10. Deterding, S., and Dixon, D. (2011). From Game Design Elements to Gamefulness : Defining " Gamification. *ACM MindTrek.* 9-15.
11. Dix, A., Finlay, J. E., Abowd, G. D. and Beale, R. (2003). Human Computer Interaction. Prentice-Hall.
12. Dong, T., Dontcheva, M.., Joseph, D., and Karahalios, K., Newman, M. W., Acherman, M. S., (2012). Discovery-based Games for Learning Software, *ACM CHI*. 2083-2086.
13. Dreyfus, S. E., and Dreyfus, H. L. A Five-Stage Model of the Mental Activities Involved In Directed Skill Acquisition. *Technical report*, 1980.
14. Fernquist, J., Grossman, T. and Fitzmaurice, G. (2011) Sketch-sketch revolution: an engaging tutorial system for guided sketching and application learning. *ACM UIST*. 373-382.
15. Flatla, D., Gutwin, C., Nacke, L., Bateman, S. and Re-Mandryk, R. (2011) Calibration games: making calibration tasks enjoyable by adding motivating game elements. *ACM UIST*. 403-412.
16. Fiore, A.T., Tiernan, S.L., Smith, M.A. (2002). Observed behavior and perceived value of authors in usenet newsgroups: bridging the gap. *ACM CHI*, 323-330.
17. Grabler, F., Agrawala, M., Li, W., Dontcheva, M. and Igarashi, T. (2009). Generating photo manipulation tutorials by demonstration. *ACM SIGGRAPH*. 28(3):66.
18. Gee, J.P. (2003) *What Video Games Have to Teach Us About Learning and Literacy*. New York: Palgrave Macmillan.
19. Grossman, T., Fitzmaurice, G. (2010) ToolClips: an investigation of contextual video assistance for functionality understanding. *ACM CHI*. 1515-1524.
20. Grossman, T., Fitzmaurice, G. and Attar, R. (2009) A survey of software learnability: metrics, methodologies and guidelines. *ACM CHI*. 649-658.
21. Harel, D. (1987). A Visual Formalism for Complex Systems. *Science of Computer Programming* , 231–274.
22. Hunicke, R., LeBlanc, M., Zubek, R. (2004) MDA: A Formal Approach to Game Design and Game Research. *Proc.AAAI04 WS on Challenges in Game AI*, 1-5.
23. Inbar, O., Tractinsky, N., Tsimhoni, O. and Seder, T. (2011) Driving the Scoreboard: Motivating Eco-Driving Through In-Car Gaming, *CHI Gamification Workshop*.
24. Kelleher, C. and Pausch, R. (2005). Stencils-based tutorials: design and evaluation. *ACM CHI*. 541-550.
25. Linehan, C., Kirman, B., Lawson, S., and Chan, G. (2011). Practical, appropriate, empirically-validated guidelines for designing educational games. *ACM CHI*. 1979-1988.
26. Mack, R. L., Lewis, C. H., and Carroll, J. M. (1983). Learning to use word processors: problems and prospects. *ACM Trans Inf Syst.* 1, 254-271.
27. McGrenere, J., and Moore, G. (2000). Are We All In the Same "Bloat"? ACM GI, 187-196.
28. Malone, T. (1981) Toward a Theory of Intrinsically Motivating Instruction. *Cognitive Science*, 4, 333-370.
29. Malone, T. W. (1982). Heuristics for designing enjoyable user interfaces: Lessons from computer games. *ACM CHI*. 63-68.
30. Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.
31. O'Neil, H.F., Wainess, R. and Baker, E.L. (2005). Classification of Learning outcomes: evidence from the computer games literature. *The Curriculum Journal,* 16. 455-474.
32. Palmiter, S. and Elkerton, J. (1991). An evaluation of animated demonstrations of learning computer-based tasks. *ACM CHI*. 257-263.
33. Pongnumkul, S., Dontcheva, M., Li, W., Wang, J., Bourdev, L., Avidan, S. and Cohen, M. F. Pause-and-play: automatically linking screencast video tutorials with applications. *ACM UIST*. 135-144.
34. Prensky, M. (2001). *Digital Game-Based Learning*. McGraw-Hill.
35. Ramesh, V., Hsu, C., Agrawala, M. and Hartmann, B. (2011) ShowMeHow: translating user interface instructions between applications. *ACM UIST*. 127-134.
36. Reeves, B. and Read, J.L. (2009). *Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete*. Harvard Business School Press.
37. Rosas, R., Nussbaum, M., Cumsille, P., Marianov, V. etc. (2003) Beyond Nintendo: design and assessment of educational video games for first and second grade students. *Computers& Education,* 40 71–94.
38. Shneiderman, B. (2004). Designing for fun: how can we design user interfaces to be more fun?.*Interactions* 11, 5, 48-50.
39. Singh V, Twidale MB, and Rathi D. (2006). Open Source Technical Support: A Look at Peer Help-Giving. *IEEE HICSS*, 6, 118c.
40. Sweetser, P., and Wyeth, P. (2005). GameFlow: a model for evaluating player enjoyment in games. Computer Entertainment, 3, 3.
41. Tuck, R. and Olsen, D. R. (1990). Help by Guided Tasks: Utilizing UIMS Knowledge. *ACM CHI*. 71-78.
42. Von Ahn, L. and Dabbish, L. (2008). Designing games with a purpose. *Communications Of ACM* 51, 8, 58-67.
43. Zichermann, G. and Linder, J. (2010). *Game-Based Marketing: Inspire Customer Loyalty Through Rewards, Challenges, and Contests*. Wiley.