

TapBeats: Accessible and Mobile Casual Gaming

Joy Kim
Computer Science & Engineering
University of Washington
Seattle, WA 98195
jojo0808@uw.edu

Jonathon Ricaurte
Computer Science & Engineering
University of Washington
Seattle, WA 98195
ricaurt2@uw.edu

ABSTRACT

Conventional video games today rely on visual cues to drive user interaction, and as a result, there are few games for blind and low-vision people. To address this gap, we created an accessible and mobile casual game for Android called TapBeats. The gameplay is based on audio cues and is similar to Simon Says – the game plays a rhythmic pattern on top of background music using four different sounds, and the goal of the player is to repeat the pattern accurately. In addition, we developed a gesture system that utilizes text-to-speech and haptic feedback to allow blind and low-vision users to interact with the game’s menu screens using a mobile phone touchscreen. A graphical user interface is also included to encourage sighted users to play as well. We aim to create an accessible and fun game through which both blind and sighted users can share a common game experience.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *input devices and strategies, voice I/O*.
K.8.0 [Personal Computing]: General – *games*.

General Terms

Design, Human Factors

Keywords

Accessibility, blind users, audio games, mobile games, haptics and gestures, user interfaces

1. INTRODUCTION

Video games can build bonds between people that play together, teach productivity skills, and allow players to practice achieving difficult goals [5]. However, some people have limited access to video games. With the video game market placing an increased emphasis on graphics, the visual aspects of games are becoming more and more important. As a result, blind and low-vision people are often excluded from video game play [8]. In order to address this gap, we created an accessible and mobile casual game for Android called TapBeats. Casual games can effectively reduce stress and improve mood [7]; through TapBeats, we hope that the blind and low-vision population will have more opportunities to enjoy the benefits of casual gaming.

We designed TapBeats to be based on audio cues so that blind

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.

Copyright 2011 UW CSE 481H Accessibility Capstone.

and low-vision people could be included in game play. TapBeats is a musical rhythm game similar to Simon Says – the game plays a rhythmic pattern composed of four different sounds, and the goal of the player is to repeat the pattern accurately using four buttons placed in each corner of the touchscreen. While this happens, music that matches the rhythm of the pattern plays in the background. Blind and low-vision users are also able to navigate through the game menus through the use of text-to-speech and haptic feedback. TapBeats also includes a graphical interface to encourage play by sighted users as well.

2. RELATED WORK

There have been several attempts to design and build video games that are accessible to blind users. Blind Hero [9] was a modified version of an open source form of Guitar Hero called Frets on Fire [2]. Guitar Hero is a game where the player presses certain buttons on a guitar-shaped controller based on visual cues displayed on the screen. Blind Hero attempted to translate these visual cues into vibrotactile cues using a glove worn by the player; vibration on a finger prompted the player to press the corresponding button on the controller. A user study showed that blind and low-vision players were able to successfully play and enjoy Blind Hero. For TapBeats, we decided to employ audio cues to prompt user interaction rather than using tactile cues because we wanted our game to run on a mobile phone without need for extra equipment.

There are also a number of mobile applications and games aimed towards teaching Braille, including a previous capstone project called BrailleBuddies [1] which also used tactile and audible cues to drive interaction with the user. However, for TapBeats, we wanted to develop an entertaining game without any particular educational goals.

AudiOdyssey [3], a game developed by MIT for the Nintendo Wii, is very similar to TapBeats in that it employs gameplay similar to Simon Says where the player mimics some pattern given by the game. In addition, AudiOdyssey was created with the goal of allowing blind and low-vision people and sighted people to enjoy the same level and quality of gameplay. We aim to achieve this same goal with TapBeats, though by making TapBeats mobile rather than for a console, players will be able to carry the game with them and play it whenever they want.

3. TAPBEATS

3.1 Gameplay

TapBeats was designed to have two functions: to provide gameplay similar to Simon Says and to act as a creative musical tool. As a result, we developed several different modes for TapBeats that the player can choose from (Figure 1).

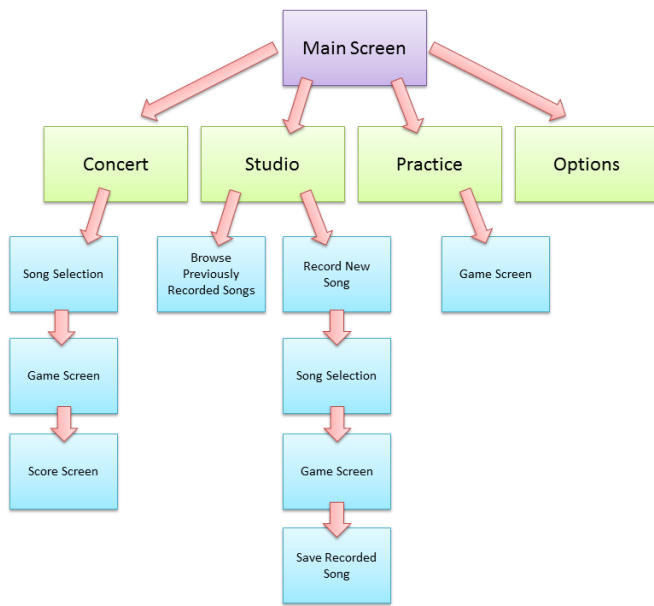


Figure 1. TapBeats system map.

3.1.1 Concert Mode

Concert Mode is the main gameplay mode of TapBeats. After selecting Concert Mode, players are taken to a song selection screen where they can choose a song. Each song has a difficulty of Easy, Medium, or Hard. Once a song is chosen, the user enters the game screen which is divided into four equal quadrants that each emit the sound of a different instrument when pressed. Currently, the four instruments the player can use are based on drum sounds (Figure 2).

As the song plays in the background, the game will play a pattern of drum beats along with the rhythm for one measure. In the following measure, the player is to mimic the pattern of drum beats using the buttons on the screen. The game will then play another pattern of drum beats, and the player will mimic the new pattern. This goes on until the song finishes playing.

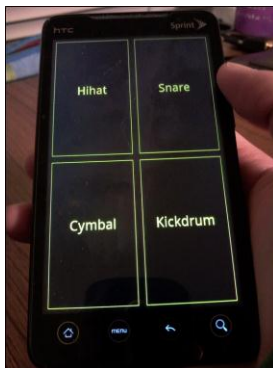


Figure 2. The game screen.

In music games similar to TapBeats, such as Dance Dance Revolution and Guitar Hero, a common feature is feedback given to the player regarding how well they are doing. These games give visual and audio feedback to do this; for example, in Dance Dance Revolution, a rating such as “Perfect!” or “Bad!” will appear on the screen with every dance step the player takes. In addition, when the player achieves certain milestones, such as successfully hitting the right steps 50 times in a row, an in-game announcer

will audibly praise the user while something like “50 combo!” appears on the screen. We wanted to give the same kind of feedback to our users without depending on visual methods to do so.

To achieve this, we added audio feedback to the gameplay. If the user presses the wrong note, or presses a note at the wrong time, a cowbell will play at the same time as the regular instrument sound to indicate to the player that they are doing the wrong thing. Also, when the player achieves a certain milestone, such as hitting several correct notes in a row, the game will play a short sound clip of a cheering audience.

3.1.2 Studio Mode

One of the goals of TapBeats was to allow the user to use it as a creative musical tool. In Studio Mode, the player can either record a new song or browse previously recorded songs. When recording a new song, the player chooses a song to “freestyle” to, without having to mimic any patterns given by the game. The game records the player-created song simply by turning on the phone’s microphone, which listens to the phone’s speaker. The player can add more to their song by singing or playing other instruments into the microphone.

Once the recording is completed, the user can choose to save the recording, play back the recording, redo the recording, or go back to the main menu. When saving a recording, the user speaks a name into the microphone, which gets saved as a sound file.

The player can also browse and listen to previously recorded songs. While exploring each song option, the user-given name that was recorded for that song will play. Selecting a song will cause the recorded song to start playing.

3.1.3 Practice Mode

Based on a preliminary review of our application by a blind colleague, we found that it was important for blind users to be able to explore and develop a feel for how the gameplay worked. Therefore, we added a Practice Mode where users can press the instrument buttons without the added stress of being recorded or having to correctly mimic patterns given by the game. There is no music playing in the background.

3.2 Menu Navigation and Accessibility

To give blind users the ability to interact with TapBeats via a mobile phone touchscreen, we developed a gesture system based on Slide Rule [4]. To explore menus and figure out the available options, the user can slide their finger around on the screen (Figure 3). As the user’s finger touches a menu item, the phone will speak out the menu item name and generate a small vibration. Once the user hears the desired item, they can select it by double tapping anywhere on the screen. To return to a previous menu screen, the user can swipe to the left or to the right.



Figure 3. A user sliding their finger to explore menu items.

During gameplay, the user can pause the game by swiping their finger to the left or to the right on the screen. While the game is paused, the game will tell the user how to exit or resume the game. To exit the game, the user can swipe their finger to the left or right once more. To resume playing the game, the user can double tap anywhere on the screen.

3.3 Songfiles

Because we wanted to eventually allow users to play content created by themselves or by other players, we devised songfiles as a flexible way to define Concert Mode gameplay (as opposed to hard-coding the gameplay for each song).

Each song in TapBeats corresponds to an .ss file, or a songfile. A songfile is a plain-text file that contains metadata about the song (such as title and difficulty level) that TapBeats uses to populate its song lists, as well as the timing and instrument information that is used during Concert Mode gameplay.

3.3.1 Format

The name of each property is defined by a '#' character followed by a word in all capital letters. On the following line, the value for that property is defined.

The #NOTES property must be the last property in the songfile, and has several lines of values rather than just one. The lines in the #NOTES property contain (instrument button number, time since beginning of song in milliseconds) pairs, which define what time the user should be pressing which button. Figure 4 is an example of a songfile.

```
#TITLE
Classic Rock Short
#ARTIST
FreeDrumLessons.com
#MUSIC
classic_rock_easy_short
#DIFFICULTY
Easy
#CREDIT
Joy Kim
#PRELUDE
3261
#NOTES
3 7989
3 8871
3 9830
3 10799
4 15473
4 16446
4 17395
4 18324
4 18564
4 22985
3 23927
4 24863
3 25762
3 30500
3 31382
3 32341
3 33310
```

Figure 4. Example of a songfile.

As we develop ways to allow users to create their own songs, the format of the songfile may change in the future.

4. CONCLUSION

4.1 Future Work

There are several future features planned for TapBeats. In particular, we wish to allow user-created content. A built-in songfile editor would enable users to create songs that are playable in Concert or Studio Mode, and some sort of in-game “market” would allow users to share and download these songs. This way, users will be able to play along with songs they like and have a larger variety of songs to choose from. In addition, we plan to allow the user to select between multiple sets of instruments, rather than always having to use the default drum set. Users could even create and share their own custom instrument sets for use in TapBeats.

We also plan to add several multiplayer aspects to the game. First, a Multiplayer Mode will allow several players to participate in a “jam session” together, where each user plays using a different instrument set. While all the users are listening to a common song, each user will play their part. At the end of the game, the players will receive both team and individual scores. Second, we plan to add a board of Concert Mode high scores. This board would be updated through an Internet connection, so that players can see how well they did compared to other TapBeats players around the world.

We also would like to improve audio feedback during gameplay. Currently, a short sound clip of a cheering audience plays when a milestone (such as a certain number of correct notes in a row) is achieved. However, this only informs the user about how well they are during at that particular moment and not overall. Audio Space Invaders [6] was an accessible game that varied the volume of different types of sounds to inform the player about enemy position and strength; TapBeats could use a similar strategy. For example, the cheer of the audience could be continual and grow

stronger or weaker based on the player's performance as the song progresses.

Another thing we plan to do is extend the capabilities of Studio Mode. The user could opt to record a song with no music playing in the background using a certain instrument set, and then rerecord over the recording using different instruments as many times as they want.

Lastly, we plan to run a user study in which we evaluate the gameplay learning curve, the enjoyment level of players, and the usability of the TapBeats interface and gesture system. With feedback from blind users, we hope to make TapBeats truly accessible and fun.

4.2 Benefit

TapBeats is not only spontaneous and easy to play due to its mobility, but is also a casual game that both the blind and the sighted can play. Through this application, we aim to create a game that allows players to share the same game experience.

5. ACKNOWLEDGMENTS

We would like to thank Shaun Kane, our mentor, as well as our instructor, Richard Ladner, and our TAs Shani Jayant and Shiri Azenkot for all their guidance and help. Special thanks to the University of Washington, UW Computer Science & Engineering, and Google.

6. REFERENCES

- [1] Acuario, C. and Hollier, J. 2010. BrailleBuddies: A Mobile Game for Teaching Blind and Low Vision Children to Read and Write Braille. <http://mobileaccessibility.cs.washington.edu/projects/>.
- [2] Frets on Fire. <http://fretsonfire.sourceforge.net/>.
- [3] Glinert, E. and Wyse, L. 2007. AudiOdyssey: an accessible video game for both sighted and non-sighted gamers. In *Proceedings of the 2007 conference on Future Play* (Future Play '07). ACM, New York, NY, 251-252.
- [4] Kane, S.K., Bigham, J.P. and Wobbrock, J.O. 2008. Slide Rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility* (Halifax, Nova Scotia, Canada, October 13 - 15, 2008). Assets '08. ACM, New York, NY, 73-80. DOI=<http://portal.acm.org/citation.cfm?id=1414471.1414487>
- [5] McGonigal, J. 2010. Jane McGonigal: Gaming can make a better world [Video file]. http://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world.html.
- [6] McGrindle, R. J. and Symons, D. 2000. Audio Space Invaders. In *Proceedings of the 3rd International Conference of Disability, Virtual Reality & Associated Technologies* (Alghero, Sardinia, Italy, September 23 - 25, 2000). ICDVRAT '00. 59--65.
- [7] Russoniello, C.V., Fish, M., O'Brien, K., Pougatchev, V. and Zirnov, E. 2011. The Efficacy of Prescribed Casual Video Games in Reducing Clinical Depression and Anxiety [PowerPoint presentation]. http://www.ecu.edu/cs-hhp/rcls/biofeedback/upload/Matt-Fish-Slides_03-01-11.pptx.
- [8] Valente, L., Sieckenius de Souza, C. and Feijo, B. 2008. An exploratory study on non-visual mobile phone interfaces for games. In *Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems* (Porto Alegre, Brazil, October 21 - 24, 2008). IHC '08.
- [9] Yuan, B. and Folmer, E. 2008. Blind hero: enabling guitar hero for the visually impaired. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility* (Halifax, Nova Scotia, Canada, October 13 - 15, 2008). Assets '08. ACM, New York, NY, 169-176. DOI=<http://doi.acm.org/10.1145/1414471.1414503>