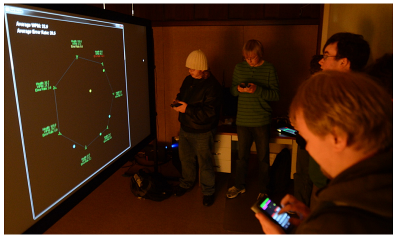

Figure 1: A group playing Text Blaster. Each player uses an Android mobile device to control their ship. The global state of the game is shown on the large projection wall.

Text Blaster: A Multi-Player Touchscreen Typing Game

Keith Vertanen
Montana Tech
Butte, Montana USA
kvertanen@mtech.edu

Justin Emge
Montana Tech
Butte, Montana USA
jmemge@mtech.edu

Haythem Memmi
Montana Tech
Butte, Montana USA
hbmemmi@mtech.edu

Per Ola Kristensson
University of St Andrews
St Andrews, UK
pok@st-andrews.ac.uk

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.
ACM 978-1-4503-2474-8/14/04.
http://dx.doi.org/10.1145/2559206.2574802

Abstract
Text Blaster is a multi-player shoot 'em up game based on players typing sentences on a mobile device's touchscreen keyboard. Players attempt to be the last player standing by using the speed, precision, and timing of their typing to annihilate competing players. Our game utilizes a sentence-based decoding approach in which users type an entire sentence before our auto-correction algorithm infers the most likely text. Text Blaster provides an engaging and competitive game for use in investigating performance and design aspects of touchscreen text entry interfaces.

Author Keywords
Text entry; game design; touchscreen keyboard

ACM Classification Keywords
H.5.2 [Information interfaces and presentation: User Interfaces]: Input devices and strategies.

Introduction
In this paper we describe Text Blaster, a multi-player game that we created based around users typing sentences on a touchscreen mobile device (Figure 1). Our game depends on players typing sentences both quickly and accurately. Each player controls the torpedo launcher of a spaceship by typing a series of sentences on a virtual QWERTY touchscreen keyboard. The destructive power of

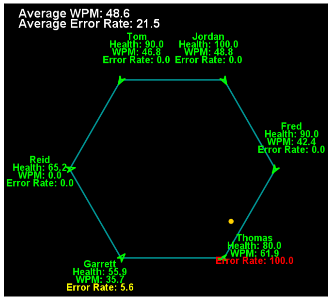

Figure 2: Six players near the start of a game. Each player's ship shows the player's name, the ship's current orientation, and the player's remaining health. Additionally, the entry rate and error rate of the player's last shot is shown. In the upper-left is the average entry rate and error rate of this game. One photon torpedo is currently in-flight, the color of the torpedo indicates how powerful the shot is.

a player's shot is influenced by how fast a given sentence is entered. The exact trajectory of a player's shot is influenced by how accurately a sentence is typed. Additionally, players must time each shot in order to target a particular enemy ship or to defend themselves against an incoming torpedo. Players attempt to be the last player standing by eliminating all other players.

Not only is our game fun, but it can also serve as a research platform for investigating performance and design questions related to touchscreen keyboard text entry. This is similar to past work that has attempted to advance text entry research by developing a game (e.g. [4, 9]).

Sentence-Based Text Entry

Researchers have explored many alternative text entry methods for mobile devices such as keyboards that model uncertainty [15], optimized keyboards [1, 10, 11, 16], gesture keyboards [7, 17], multimodal speech-touch systems [6, 12] and keyboards modeling people's walking style [2] or grip [3] (see Kristensson [5] for a recent overview).

In our mobile text entry interface, we delay automatic typing correction until after the user enters an entire sentence. Such *sentence-based decoding* may enable fast text entry since users will not be slowed by frequent monitoring and correction of word-at-a-time recognition results. It may also allow the keyboard to make fewer total recognition errors since the recognition algorithm can leverage the complete tap sequence during its search.

Our decoder takes a sequence of touch events and searches for the most probable set of sentences given that sequence of touches. The decoder uses a probabilistic keyboard model, character language model, and word language model. During the decoder's initial search, we use a 12-gram language model. Our initial search finds a list of up to the 50-best sentence hypotheses. This N-best list is then rescored with a 4-gram word language model. Our decoder searches the hypothesis space of all possible character sequences given a sequence of taps. In order to keep the search tractable, we employ beam width pruning.

Due to the large memory footprint of our language models, the recognition occurs on a nearby desktop computer. The mobile devices used by the players communicate with the desktop over a wireless network. Recognition time, including network latency, was fast with an average recognition delay of about 0.2 s per sentence.

Game Mechanics

The game is played by two or more players. Each player uses a touchscreen mobile device to control a spaceship. Each player's spaceship is located at the vertex of a polygon (Figure 2). A player's ship rotates automatically back and forth, sweeping its torpedo launcher through all other players.

Players are presented with short memorable sentences from the Enron mobile dataset [13]. Players type each sentence on an onscreen QWERTY keyboard displayed on their device. After typing all the letters of a sentence, the player signals the entry is complete by tapping with two fingers. This causes our sentence-based decoder to find the most probable sentence given the player's noisy tap data. It also causes the player's ship to fire a photon torpedo.

A torpedo launched by a player is given a destructive power proportional to the player's entry rate on the stimuli sentence. Entry rate was measured in words-per-minute. As is commonly done in text entry research, we defined a word as five consecutive characters including spaces.

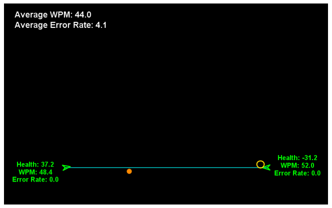

Figure 3: A final shootout between two players. The yellow circle near the right player is an explosion caused by a recent torpedo impact on that player.

We measured the accuracy of a player's entry using character error rate (CER). CER is the number of insertions, substitutions and deletions required to transform the recognized text into the reference text, divided by the number of characters in the reference text. If the player enters a sentence with a 0% CER, the torpedo goes in a straight line. Non-zero error rates cause a player's torpedo to shoot at a random offset angle proportional to the magnitude of the error rate. Thus, success in the game depends on players trying to enter text both as quickly and as accurately as possible.

In addition to entry speed and accuracy, the game also relies on timing. To target a particular opponent, a player must time the end of an entry to coincide with pointing at that opponent. Since torpedoes can collide in-flight, players can also time shots to deflect incoming torpedoes.

Photon torpedoes travel across the playing field, hopefully impacting one of the other players. Each player's ship has only a limited amount of health. Once a player's health is zero, their ship explodes and the polygon collapses by one vertex. Eventually, the final two players take part in a head-to-head shootout. The last two players are situated on a line with their ships oscillating back and forth (Figure 3). The last player remaining is declared the winner.

Players are penalized for sloppy text entry. If a player's entry has a 5-10% CER, the player's torpedo launcher malfunctions and fails to fire. If a player's entry has a CER of over 10%, the torpedo launcher backfires and ends up damaging the player's own ship.

Players view the entire state of the game on a large display. All players may view the same large display, or they may view separate large displays. All displays are synchronized to show the same game. On the large screen, all players and all photon torpedoes are visible. The large screen also shows the average entry and error rate for the current game. After a player makes a shot, the shot's entry and error rate are displayed next to the player's ship.

Each player's mobile device provides visual feedback (Figure 4). The device shows the current orientation of the player's ship as well as any torpedoes in the vicinity of the player's ship. The device displays the current sentence that needs to be entered as well as the last recognition result along with its entry and error rate. The device makes sounds and vibrates whenever the player taps letters, fires torpedoes, or has their ship damaged.

Future Work and Conclusions

The competitive nature of Text Blaster encourages users to enter text both quickly and accurately. We hypothesize that these properties might make Text Blaster an ideal platform for conducting both laboratory and crowdsourced text entry experiments.

We have two experiments currently planned for Text Blaster. First, the game could be modified to remove the visual keyboard. Players would tap out letters with respect to a keyboard they imagine at a size and a location of their own choosing. This would allow us to investigate eyes-free touchscreen text entry. In preliminary research [14], we found blindfolded participants could type at over 20 words-per-minute with our decoder recognizing over one-third of sentences with no errors.

Second, instead of tapping a spacebar between words, players could make a right swipe gesture (similar to [8]). Using swipe for spaces has the potential to significantly improve recognition accuracy as such gestures provide reliable information to the recognizer about one of the most commonly entered characters.

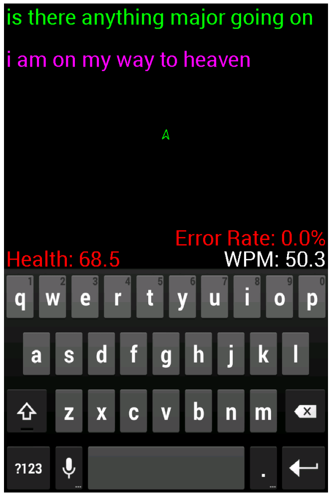

Figure 4: Screenshot of a player's device during the game. The current orientation of the player's ship as well as any torpedoes in the player's area is shown in the center. The green text at the top is the next sentence to be entered. The purple text is the previous recognition result. Directly above the keyboard, the player's remaining health is shown as well as the error and entry rate from the player's last entry.

In summary, we created a fast-paced and fun multi-player game. Players attempt to blast each other by entering sentences quickly, accurately and with precise timing. Our sentence-based interaction style encourages users to type fast by avoiding traditional feedback distractions such as the display and the correction of the last word entered. We plan to use our game to explore the design and performance of touchscreen mobile text entry interfaces.

Acknowledgments

We thank Matt Morris for helpful discussions.

References

[1] Dunlop, M. D., and Levine, J. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proc. CHI* (2012), 2669–2678.

[2] Goel, M., Findlater, L., and Wobbrock, J. WalkType: using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proc. CHI* (2012), 2687–2696.

[3] Goel, M., Jansen, A., Mandel, T., Patel, S. N., and Wobbrock, J. O. ContextType: using hand posture information to improve mobile touch screen text entry. In *Proc. CHI* (2013), 2795–2798.

[4] Henze, N., Rukzio, E., and Boll, S. Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. In *Proc. CHI 2012*, ACM Press (2012), 2659–2668.

[5] Kristensson, P. O. Five challenges for intelligent text entry methods. *AI Magazine 30*, 4 (2009), 85–94.

[6] Kristensson, P. O., and Vertanen, K. Asynchronous multimodal text entry using speech and gesture keyboards. In *Proc. Interspeech 2011*, ISCA (2011), 581–584.

[7] Kristensson, P.-O., and Zhai, S. SHARK^2: a large vocabulary shorthand writing system for pen-based computers. In *Proc. UIST 2004*, ACM Press (2004), 43–52.

[8] Kristensson, P. O., and Zhai, S. Relaxing stylus typing precision by geometric pattern matching. In *Proc. IUI* (2005), 151–158.

[9] Kristensson, P. O., and Zhai, S. Learning shape writing by game playing. In *Proc. CHI (Extended Abstracts)* (2007), 1971–1976.

[10] MacKenzie, I. S., and Zhang, S. X. The design and evaluation of a high-performance soft keyboard. In *Proc. CHI* (1999), 25–31.

[11] Oulasvirta, A., Reichel, A., Li, W., Zhang, Y., Bachynskyi, M., Vertanen, K., and Kristensson, P. O. Improving two-thumb text entry on touchscreen devices. In *Proc. CHI* (2013), 2765–2774.

[12] Vertanen, K., and Kristensson, P. O. Parakeet: A continuous speech recognition system for mobile touch-screen devices. In *Proc. IUI 2009*, ACM Press (2009), 237–246.

[13] Vertanen, K., and Kristensson, P. O. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proc. MobileHCI* (2011), 295–298.

[14] Vertanen, K., Memmi, H., and Kristensson, P. O. The feasibility of eyes-free touchscreen keyboard typing. In *Proc. ASSETS* (2013), Article No. 69.

[15] Weir, D., Pohl, H., Rogers, S., Vertanen, K., and Kristensson, P. O. Uncertain text entry on mobile devices. In *Proc. CHI* (2014), in press.

[16] Zhai, S., Hunter, M., and Smith, B. A. The Metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In *Proc. UIST* (2000), 119–128.

[17] Zhai, S., and Kristensson, P. O. The word-gesture keyboard: reimagining keyboard interaction. *Communications of the ACM 55*, 9 (2012), 91–101.