Bomb Bot: Looking At Communication In High-Stress Situations

Peter Buk
University of Calgary
2500 University Drive NW
Calgary, AB CANADA
pkpbuk@ucalgary.ca

Kevin Ta
University of Calgary
2500 University Drive NW
Calgary, AB CANADA
kta@ucalgary.ca

Abstract
Priorities can vary vastly across situations, especially in collaborative interactions, and this can sometimes lead to conflicts. We built a bomb defusing game that looks at how people prioritize their own tasks in order to accomplish a shared goal. By separating knowledge and operating environments, we observed creative ways in which people establish trust and resolve communication issues. Conducting a brief pilot study, BombBot provides a promising platform for analyzing collaborative communication and behaviours.

Introduction
We often find ourselves in situations where we have to collaborate with others towards a common goal. Even though the ideal solution would be to work together with perfect synergy, collaboration is often hindered by factors such as the environment, human behaviours, and conflicting priorities. For example, consider the situation where one person is giving driving directions over the phone to another person who is lost while driving. The driver may be too busy to give location clues, while the other person may be pressured to ensure the driver does not miss a turn or become lost. Since the two collaborators are in different situations, with one providing remote expertise, and the other interacting directly in a hazardous environment, their priorities will differ and may even clash.

Current literature often discusses how we can create novel technology to try to improve communication, however not very many look into people's behaviours when using collaborative technologies. In this paper, we will address the following research questions: (1) How do people collaborate in stressful environments? (2) What role do differing priorities have on remote collaboration?

To address the above questions, we introduce BombBot, a bomb defusing game that exploits people's differing priorities when put in a time-pressured situation. One player becomes the Bomb Expert, who knows the current threat level of the bombs and possess the knowledge to defuse them. The other player is the BombBot, who seeks out bombs within their proximic space (using mobile gyroscopes) and must describe their appearance to the Expert. The aim is to work together and communicate effectively in order to stay alive for as long as possible. We separate the two collaborators enough such that they can only communicate verbally. This is to limit redundant communication and focus on communicating only essential information to defuse each bomb.

This game provides a way to observe people's clashing priorities and their workarounds to overcome these issues. We discuss some of the creative ways that participants in a pilot study came up with to build a communication protocol, establish trust, and synergize in this real-time-pressured environment. While this type of simulation can potentially be stressful and cause conflict, we found it to be a good platform to analyze collaborative behaviours as well as being an entertaining game.

Related Works

Video Collaboration

BombBot was inspired by the remote expert and local user model of computer supported collaborative work (CSCW). A remote expert connected via a network connection assists a local user to solve a problem on their end. The local user will need to provide information about their environment and problem, as well as communicate information necessary to solve the problem. Whereas the expert will need to determine which course of action will solve the problem and relay that back to the local user. Normally this is done using some video and/or audio channel for real time collaboration.

Much of the literature we have observed focused on expressivity and situational awareness of the participants. BeThere [2], Gauglitz[4], and JackIn[3] offer ways to express gestures, record annotations, and provide self-serve environment awareness respectively. The focus however, is on adding novel technology to solve problems, which may increase behavioral complexity and opens up a slew of domain specific problems as a result of the technology.

Building from the literature, BombBot distinguishes itself from these studies as we aim to look at behaviours that arise from such systems. Building on a similar situation presented in BeThere [2], Gauglitz[4], and JackIn[3], we created a game in which we can look at behaviours that arise when communication is primarily focused on prioritizing and planning actions in a stressful real-time environment. In this manner, we can compare the effectiveness of expressivity versus communication protocol.

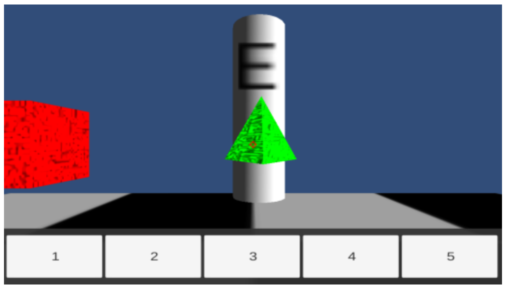

Figure 1. BombBot Mode

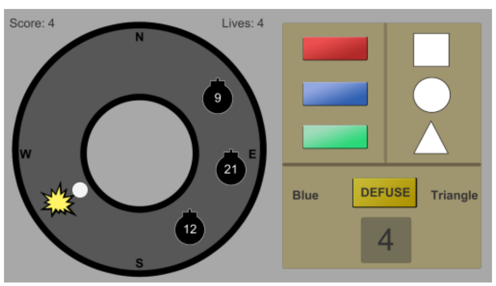

Figure 2. Expert Mode

Power and Territoriality
Thom-Santelli [1] looks at how experts may express territorial behaviour when collaborating with novices in collaborative systems. Experts may feel defensive of their work, especially if they put a lot of effort into it, and may not perceive the novice's contributions as having as much value. A study was conducted in which experts and novices were asked to collaboratively tag objects in a museum, and to evaluate the quality of each other's tags. The results showed that the experts felt their contributions were more valuable due to their domain knowledge and the inherent threat of competition in this task as only the top five tags were accepted. While our game is not a competition oriented game, experts are given enough knowledge and authority such that the blame can easily be put on the local user.

We should expect territorial behaviour if we run a study on a large group of participants; as well as when players become accustomed to our game and develop their own communicative strategies. In that regard, we may see some power struggles develop as one collaborator may decide to take on the role of leader and direct the other collaborator on what to do. Depending on individual behaviour, this may work in favour of the pair or introduce conflict that hinders them.

Despite advances in technology that enables us to be more expressive remotely, we find that there is still a lack of literature on the limits of human communication in a time constrained environment. BombBot aims to bring out communication problems through differing priorities in order to investigate the ways humans communicate and how they resolve their problems by employing various communication strategies. In the broader research frame, we hope to establish interest in supporting long term remote collaboration as users adapt their communication protocols and whether these have implications for overall system design.

BombBot Prototype System

Game Mechanics
BombBot is a collaborative game where pairs of players exchange information about bombs in order to defuse them. Each player is assigned a role of either the Expert or the BombBot, whom are the remote expert and local user respectively as per the literature. The goal of the game is to survive for as long as possible against endless waves of bombs with only seven lives. The game is divided into rounds of increasing difficulty, but players can take a break in between each round.

In BombBot mode (Figure 1), players must search for bombs that spawn in a circular plane around them. Using the gyroscope, players rotate on the spot to look around the virtual reality world for bombs. There are also four directional pillars to assist the BombBot with maintaining their orientation. When a bomb has been found, its shape and color must be told to the Expert to get the correct diffusion solution. Once a bomb has been defused, the BombBot then searches for the next bomb to defuse.

In Expert mode (Figure 2), players have an overview of the entire circular plane and an indicator of all the current bombs with their remaining timers. They also have an indicator for the current direction BombBot is facing. To help defuse the bombs, they have a decoding panel with two input panels for the colour and shape of the current bomb. Once they get this information from

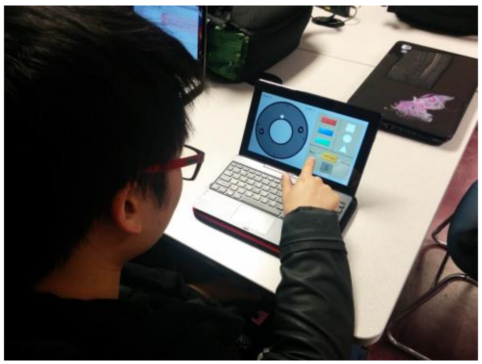

Figure 3. A participant playing Expert mode on the touchscreen PC

the BombBot, they can hit the input buttons and get an output solution which the BombBot can use to successfully defuse the bomb. If a bomb's timer runs out or if an incorrect solution is used, the bomb will explode causing the team to lose a life.

Implementation

BombBot is a game that was implemented on Unity 4.6.2f1 which utilizes the new UI game objects to construct the Expert mode. Both modes utilized Unity's Remote Procedure Call system for networking. We opted to do Remote Procedure Calls (RPC) for most of our needs instead of state synchronization as it was faster to prototype with as we developed the game. Hence for all network view components, we disabled state synchronization and instead throttled our own traffic for camera rotations from the BombBot. Unity also supports a broad range of platforms with its multiplatform engine which allows BombBot to be run on any Android device (tablets and smartphones) and most PCs. This allowed rapid development of the game for both our target platforms.

Our game utilizes a smartphone's gyroscope, which was found to be relatively reliable for most modern smartphones. Some performance issues may arise with the gyroscope when sending large amounts of rotational updates over the network, which sometimes interferes with accuracy. However traffic can easily be throttled for RPC calls, or implemented with state synchronization in the unreliable operating mode.

For testing, we ran the game on a 10.1" touchscreen enabled Laptop as the Expert and a Nexus 5 smartphone as the BombBot (with resolutions 1360x768 and 1920x1080 respectively). For networking we used the wifi hotspot feature on the same Nexus 5 for our LAN operated session. The game itself has extremely low graphical requirements and can be run on nearly any system.

Level Design

To ease players in, we designed the first 5 levels with consistently increasing difficulty. Level 1 helps demonstrate how to play the game for the first time by having only 2 bombs with very long timers. Level 2 was designed to have more bombs to help participants develop their own communication protocol while being in a relatively easy environment. Finally as they approach level 5, bombs have shorter timers and many will exist simultaneously if players are not fast enough. From round 6 onward, the game will add an additional two bombs from the previous level, up to a maximum of twenty. Players can play for as long as they like until they run out of lives.

Since it was difficult to pin down the exact scenario which enables us to observe conflicting priorities, we created a standard (and admittedly quite difficult) version, and a slightly easier version of the game. Both versions only differed from level 6 and beyond.

Initial Observations

Pilot Study

We conducted a study with a total of 4 teams of two (8 participants, 4 female). Three of the teams' pairs knew each other, and one was a stranger to stranger pair. Two of the teams played the game on the standard difficulty while the other two teams played on the easier difficulty. The study was conducted in a general study area with a moderate noise level. This reflects a realistic situation where the remote collaboration might not occur in an isolated distraction-free area.

Each team was first given instructions on how to play each role. Then we began the game by allowing the players to go as far as they can before they ran out of lives or when they gave up. We then asked each player to reflect on their communication strategy in a quick round of mid-session questioning. Afterwards they proceeded to switch roles and applied their new protocol. Again, they go through as many levels as they can or until they gave up. Finally we conducted a semi-structured interview.

Communication Protocol

We found that communication between partners often devolved into a Minimal Communication Protocol. Only the most necessary words such as “SE, Blue Square, 3 [the solution]” were spoken, and any commentary or non-game related discussion waited until the round was over. Teams became very efficient at defusing bombs one at a time when they were focused, although outside distractions sometimes interfered with our player’s during the task.

Most teams used the cardinal directions as a way to reference a position. We were surprised to find that deictic references such as “turn around, look to your left” were not used very often. Team 3 tried using clock references (such as 12 o’clock for North), but quickly found it was not very effective. This was most likely because we explicitly labeled the cardinal references on both modes, making it easy to visually reference as opposed to trying to use a non-fixed reference frame.

There were also cases of ambiguity because of imprecise communication. The cardinal directions are not very precise and this was apparent when two bombs spawned in close proximity to each other. If the Expert only mentions a direction, the BombBot might not know which of the two the Expert is trying to prioritize. This very mistake cost Team 4 one of their lives. The fact that we used 3D models on the BombBot mode and 2D shapes on the Expert mode also confused Team 4 because it created two sets of terms that required interpretation.

Power Disparity

Since each mode had its own distinct set of information and affordances, power disparities were often observed where one partner would dictate the flow of the game. We had hypothesized that the Expert would likely lead the BombBot because they had the advantage of knowing all the bomb positions. In most cases however, we found that the Expert actually let the BombBot take control and find bombs on their own. One of Team 3’s participants said that they did not want to order their partner (as BombBot) too often, as they were content finding bombs on their own when there was no sense of urgency.

Once the difficulty ramped up and teams were faced with 3-4 bombs with under 20 seconds, a shift in power was observed. The Expert would often direct BombBot towards a certain direction, ignoring the BombBot’s intentions for a more priority bomb. This led to a clash of priorities as the BombBot wanted to describe and finish defusing the bomb in front of them, but the expert is telling them to go elsewhere instead. We observed cases where the Expert would give in to the

BombBot's intentions which resulted in a bomb exploding elsewhere on the map.

The best performing teams often placed a lot of trust in each other. The BombBot would give up their independence and just wait for the Expert's directions, trusting that they would know the best path to take. On the other hand, sometimes the Expert let the BombBot work on a less urgent bomb, trusting that they can finish and move on faster than having to come back later on.

Design Decisions

Channel of Communication

Most CSCW systems use some form of video and/or audio channel for synchronous communication between users. However, a common problem with video collaboration is that the expert is restricted to the local user's point of view. The expert may know how to solve the problem, but has difficulty orienting themselves or has to constantly instruct the remote user on how to provide the necessary information. This environmental awareness information can introduce communication complexity and redundancy. JackIn [3] showed a way to give the remote expert more environmental awareness information without much effort from the local worker.

We decided to use an overview radar that reveals environmental awareness with no additional communication effort. Essentially we are eliminating the expert's dependence on the local user's field of view. By doing so, we can focus the communication more towards tasks that help solve the goal, such as planning and prioritizing. Due to our limited scope, the current implementation only has collocated voice communication between the collaborators. A further study would look into how using a remote voice channel might affect communication.

Separation of Information

In order to create a situation where conflicting priorities may occur, we had to have an environment in which the collaborators are required to rely on each other, yet aren't restricted by each other's actions. The game flow had to be carefully constructed in a way that would allow for various play styles through different styles of communication. We also wanted to place an element of high stress on the players in order to probe how communication changes depending on the situation.

The resulting design had a separation of knowledge to the Expert, and operating environment to the Bomb Bot. All of the information from the bomb locations, timers, lives, and solutions were on the Expert side. But only the Bombbot could interact with the actual bomb entities and defuse them. This created a mutual interdependence where information had to be passed back and forth between the two players to make progress. However, both players still have a sense of independence by being able to decide which bomb to work on; except not much progress will be made if the players disagree. By having two pieces of information to convey (shape and colour), on top of the time pressure, diffusing multiple bombs is no longer a trivial task. We can artificially modify the stress level by changing the parameters of spawn timers and bomb timers. In future studies, we can increase the complexity of the stress by introducing additional constraints such as new bomb types, diffusion methods, and other game elements.

Discussion

Our original intent was to try to create a scenario where priorities and intentions may conflict between collaborators. After conducting our pilot study, we found that conflict was not as evident as we had expected. Rather, we observed a lot of trust between the players, but that could be attributed to the small sample size. Trust building appears to play an important role in overall synergy survival for our game. As players become accustomed to each other's abilities, there is room for common courtesy such as not pestering the BombBot too much when there is no sense of urgency. This can be interpreted as a display of control disparity from one side, or a demonstration of trust in their partner. This building of trust may be an important factor over successive games as strategies evolve with multiple plays.

Furthermore, this trust is pushed to the limits when knowledge is separated. In certain instances, the game may become seemingly hopeless, and the expert would give up rather than focus on defusing as many bombs as possible. Since knowledge is exclusive to the expert, the BombBot may feel betrayed in a way because they had trusted their partner to assist them. What they did not realize was the stress placed on the Expert that caused them to give up in the first place. To conclude, we believe that trust is one of the keys to successful collaboration in stressful environments; when this trust breaks down is when collaboration starts falling apart.

In future work, BombBot will incorporate new level and situational design to enable more complex behaviours that may arise in collaboration. The intention is to balance the game in such a way that both players will have influences on the final solution to each bomb. Currently the BombBot never integrates their own contextual knowledge into the final solution. While there had been interesting observations, it is unrealistic for most collaborative dynamics to have an expert instruct the local user on exactly what to do without input from the local user's knowledge.

Additionally, another research question that would be worth pursuing is whether collaborative behaviours can be addressed by optimizing the interface or the communication protocol. As observed in our pilot study, there is a potential to optimize communication protocol in such a way that reduces the need to communicate extra information. In one such instance, BombBot had memorized bombs if they saw them spawn and mentally marked their priorities which partially eliminated their dependence on the Expert. Hence it would be interesting to look at tradeoffs between interpreting a partner's intentions versus expressivity through interface.

Conclusion

We designed BombBot, a collaborative game in which we can observe collaborative communication. Various literature have focused on the expressivity of participants and few have examined human behaviour and its consequences on CSCW design. Our model attempts to closely match that of a remote expert and local user situation while removing redundant environmental communication that can otherwise be solved by current literature. In our pilot study, we observed some interesting behaviours around communication protocols and power disparities between participants. In future work, we would like to continue to improve the game and design scenarios in which to observe how variability in human behaviours and personalities can affect remote collaboration.

References

1. Jennifer Thom-Santelli, Dan Cosley, and Geri Gay. 2010. What do you know?: experts, novices and territoriality in collaborative systems. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 1685-1694. DOI=10.1145/1753326.1753578 http://doi.acm.org/10.1145/1753326.1753578

2. Rajinder S. Sodhi, Brett R. Jones, David Forsyth, Brian P. Bailey, and Giuliano Maciocci. 2013. BeThere: 3D mobile collaboration with spatial input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 179-188. DOI=10.1145/2470654.2470679 http://doi.acm.org/10.1145/2470654.2470679

3. Shunichi Kasahara and Jun Rekimoto. 2014. JackIn: integrating first-person view with out-of-body vision generation for human-human augmentation. In Proceedings of the 5th Augmented Human International Conference (AH '14). ACM, New York, NY, USA, , Article 46 , 8 pages. DOI=10.1145/2582051.2582097 http://doi.acm.org/10.1145/2582051.2582097

4. Steffen Gauglitz, Cha Lee, Matthew Turk, and Tobias Höllerer. 2012. Integrating the physical environment into mobile remote collaboration. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services (MobileHCI '12). ACM, New York, NY, USA, 241-250. DOI=10.1145/2371574.2371610 http://doi.acm.org/10.1145/2371574.2371610