Feedback, Latency, Accuracy: Exploring Tradeoffs in Location-Aware Gaming

[◇]Kieran Mansley [♠]David Scott [♣]Alastair Tse [♡]Anil Madhavapeddy

[◇♠♣]Laboratory for Communication Engineering
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK

[♡]Intel Research Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK

{[◇]kjm25,[♠]djs55,[♣]acnt2}@cam.ac.uk

[♡]avsm2@cl.cam.ac.uk

ABSTRACT

We are witnessing the development of large-scale location systems and a corresponding rise in the popularity of location-aware applications, especially games. Traditional computer games have pushed the limits of CPU and graphics card performance for many years and experience suggests that location-aware games will place similar demands upon location systems. Unlike traditional gaming platforms however, the mobile devices that interact with location systems are heavily constrained especially in the number of ways that feedback can be provided.

In this paper we describe a location-aware, fast-paced, close quarters action game and use it to experiment with three key components of future location-aware gaming platforms: *(i)* the location system, *(ii)* the network to connect the mobile devices, and *(iii)* the feedback and computational capabilities of the mobile devices themselves.

We investigate the tradeoffs that are possible between these components, the effect of the feedback channel and the suitability of Bluetooth as a network for mobile game devices.

Categories and Subject Descriptors

K.8.0 [**Personal Computing**]: General—*Games*

Keywords

Mobile Gaming, Location Systems, Bluetooth

1. INTRODUCTION

1.1 Location-Aware Games

Children[1] have traditionally enjoyed running around in groups playing games such as “hide-and-seek”, “musical chairs”, or simply shooting at each other with water pistols. In recent years, multi-player computer games have replaced some of these physical activities with networked interaction from a desktop computer or gaming console.

Location-aware technologies have begun to be deployed commercially, and integrated with popular mobile devices. For example, the Federal Communications Commission has mandated that all new mobile phone handsets sold in the United States have Automatic Location Identification capability [8] in order to allow emergency services to quickly locate accident victims. This has led to manufacturers integrating technologies such as Cell Identification, Global Positioning System (GPS) [13], and Assisted GPS [7] into their handsets.

Location-aware gaming aims to take advantage of these developments and combine the social face-to-face aspects of traditional games with the rich complexity that networked computer games provide.

There have recently been a number of projects taking advantage of the presence of GPS integration in mobile phone handsets (see Section 5). Since GPS only works reliably outdoors, these games generally take place in a wide-area urban setting. In this paper, we seek to prototype close quarters, low latency, high accuracy, location-aware games using the current generation of mobile phone hardware and the Active Bat system (see Section 2).

Our novel contributions include: *(i)* creating a fast-paced, close quarters, location-aware game, *(ii)* exploring the tradeoffs between the accuracy of a location system, the I/O capabilities of current mobile hardware, and the latency of user feedback, and *(iii)* investigating the viability of Bluetooth as a component in a low-latency location-aware gaming infrastructure.

We constrain our choice of gaming hardware to standard mobile phones and only use the Active Bat system to simulate what future off-the-shelf location systems may provide. This enables us to explore a new breed of games that will arise when advanced location technologies are integrated into next-generation mobile handsets.

The rest of this paper is structured as follows: Section 2 describes a flexible architecture for creating mobile, location-aware action games. Section 3 uses this architecture to build several game configurations with varying levels of feedback. These were deployed and tested, and our experiences playing them presented in Section 4. Section 5 presents some related work, while Section 6 concludes.

[1] and PhD students

SIGCOMM’04 Workshops, Aug. 30+Sept. 3, 2004, Portland, Oregon, USA.

2. GAME SYSTEM ARCHITECTURE

We have created a modular game system architecture which is ideal for conducting experiments with location-aware games. The architecture combines a number of existing technologies including: *(i)* a location system, *(ii)* a network, and *(iii)* mobile devices with built-in feedback and computational capabilities. We have built a number of location-aware game configurations using the architecture in order to demonstrate its utility. We have used these configurations to conduct experiments to help determine the necessary tradeoffs between feedback, latency and accuracy required to produce enjoyable games. Figure 1 shows each of the generic game components. Two of these components, the location system and the event system are described individually in the following sections. For expository purposes the other components are described in the context of the example games in Section 3.

2.1 The Location System

The Active Bat [10] system is a sophisticated indoor location system in which small battery powered tags ("Bats") are signalled periodically by radio to emit narrowband ultrasound pulses. These pulses are received by a network of sensors embedded in the ceiling and the time-of-flight information is used to multi-laterate the position of the Bats – a process which is accurate to about 3cm 95% of the time. Each Bat also contains a pair of push-buttons, a pair of status LEDs and a buzzer for basic I/O.

The Active Bat system can trigger 50 location updates per second per radio zone; there are two such zones in the current installation covering the entire lab.[2] The system adaptively schedules Bat updates and offers highly mobile Bats a higher Quality of Service by triggering them more frequently. Applications are also able to signal the scheduler to request more frequent updates for particular Bats for limited periods of time.

The Active Bat system was deliberately over-engineered to provide higher update rates and levels of accuracy than are likely to be found in any deployed commercial location system in the near future. Although the Bat system itself is unrealistic, it is nevertheless an ideal experimental platform for determining the properties *future* location systems must have in order to support fun location-aware games.

2.2 Game Event Distribution

We use the SPIRIT [1] spatial indexing middleware layer to take raw location events from the Active Bat system and convert them to more useful events such as *"Player X has entered Room Y"*. This is augmented by a game server which listens for SPIRIT events, and determines their impact on the game. It in turn issues (through SPIRIT) game events such as *"Player X has picked up a flag"*. This is illustrated in Figure 1.

A number of feedback components can connect to the SPIRIT middleware to receive the game events, and so augment the game. As each of these components is fully independent, we can add and remove them to experiment with different methods of feedback. These components include: *(i)* triggers to feedback through the Active Bat, *(ii)* displays of the game map, *(iii)* sound servers, and *(iv)* a Bluetooth proxy to allow mobile phones in the vicinity to receive game events.

In the following section we use combinations of these components to simulate a series of mobile gaming platforms with different feedback, location-accuracy and latency properties.

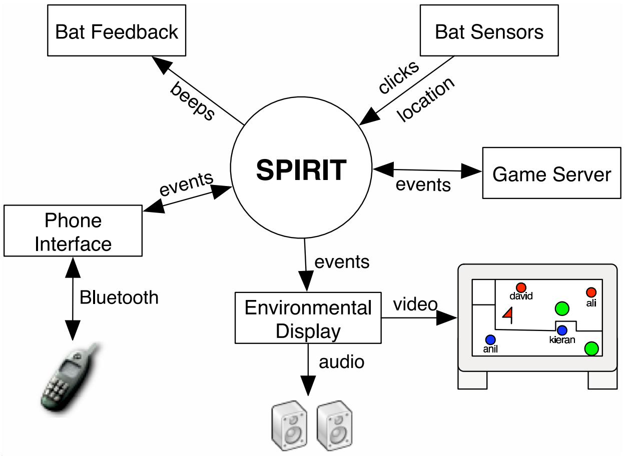

Figure 1: Architecture of the location-aware gaming platform showing how the generic components connect together.

3. THE LOCATION-AWARE GAMES

The games we have chosen to evaluate are based on the popular *Capture The Flag* and *Counter-Strike* PC games. Each game has two teams, and each team has a "base" positioned somewhere within the building. The teams compete to either capture a flag from the opponents base and return it to their base, or plant a bomb at the opponents base. To thwart the other team each player is equipped with virtual weapons, including land-mines and a shotgun.

As a minimum each player must carry an Active Bat in order to participate. This device provides both player location information and allows the player to perform actions in the game. In the system described here game actions are invoked by pressing one of the two push buttons on each Bat. The first button is used to fire weapons; the choice of weapon is indicated using *gestures*. Land-mines are laid by clicking near the floor, while the shotgun can be fired by clicking twice (to determine direction) at chest level. The second Bat button is used to request a status update (giving information about the player's health and ammunition levels).

Flags are picked up and dropped automatically simply by being in the right place at the right time. It does not require an explicit request from the player.

All objects in the game (flags, bombs, land-mines, shotguns etc.) are entirely virtual and have no physical representation. This allows for a very flexible deployment of the game (you can play it anywhere you have access to the location system) but removes a key feedback component from traditional games: the ability to see where things are.

Three configurations of the system were created in order to investigate what levels of feedback, accuracy and communication latency were required in order to ensure an enjoyable game. Each configuration was extensively play-tested while parameters — such as the Bluetooth communications latency — were varied. After playing the games an informal survey assessed how enjoyable the games were, and what the major problems or limiting factors might be. Each configuration is described in detail in the following sections.

3.1 The Bat Buzzer Component

The Bat buzzer component allows games to use the I/O capa-

[2] A previous deployment in AT&T Laboratories Cambridge had three separate radio zones – one for each floor of the building.

bilities of the Active Bat itself. Bats are extremely minimalistic devices, possessing only the most rudimentary forms of I/O: a buzzer (capable of playing a set of pre-programmed simple tunes) and a pair of push-buttons. Active Bats are therefore good prototypes of ultra-cheap consumer location devices of the future.

The games associate different tunes with certain game events allowing us to convey approximations of the game state to the player. There is an inherent tradeoff between signalling every event (possibly overloading both the user and the Bat with lots of trivial events) and signalling only selected events (which risks leaving out a critical event and confusing the user). We found the following set of events to represent a usable compromise: *(i)* object picked up, *(ii)* object dropped, *(iii)* gun fired: hit *(iv)* gun fired: missed, and *(v)* you have died. The user can also request their current status, and this is again conveyed using tones on the Bat: either "you are dead", "you are alive and have the flag", or "you are alive and have *N* spare mines".

This configuration, while primitive, could be seen to represent one possible commercial system. The mobile device is extremely small, low power, and has no expensive components such as a CPU or screen. It operates very much as a *thin client*. However, the extreme limitations of the mobile device required greater sophistication in other components to compensate. For example, the location system must be able to support gestures in order to distinguish different weapons. The very low bandwidth of the feedback channel emphasises the need for low-latency and jitter in order to guarantee timely (and hence meaningful) feedback to the user.

In practice, this game configuration enjoyed only partial success. The use of gestures (made possible by the high-accuracy of the location system) was indeed successful in mitigating problems caused by the limited number of buttons on the Bat. Despite this, the game was found overall to be confusing by the game participants. This was attributed to two factors: firstly, fast action games like *Capture The Flag* often generate events at a rate faster than we could remotely trigger the buzzers on the Bats, even when being very selective about which events are transmitted (consider a gun battle where each shot is associated with a beeping sequence). Secondly, when a beep is late there is no way for the player to tell if the event was missed completely (i.e. they should repeat the command/gesture) or whether the game is experiencing temporary lag (e.g. if the Bat radio channel is currently congested and the event notification signal has been delayed).

3.2 Sound and Display Components

Sound Servers and Ambient Displays are PC-based components for playing audio and video respectively. They both offer high-quality feedback but suffer from being immobile, unlike the Active Bats and mobile phones. The addition of these components allows for experimentation with the level of feedback provided.

A second configuration of the game added a small number of these components. Game events fall into two categories: *(i)* those that apply to an individual player (e.g. "You are dead"), and *(ii)* those that apply to multiple players (e.g. "Team A has captured Team B's flag"). Active Bats are inherently personal devices and are therefore best at handling events intended for a single player. Events intended for multiple players are broadcast as audio through the sound servers. The sound effects were high quality audio samples taken from `Quake III` and hence conveyed much more information than simple beeps from the Bats.

An ambient display was placed near each team's base displaying a continuously updated map of the virtual world. Due to the inherent immobility and field-of-view of these components they are not used to broadcast events to players "in the field" in the way the sound servers do, but instead allow them to retreat to their base to gain a view of the entire current game state. As many game objects do not have any physical representation this is the only way that players can find out where objects such as the virtual land-mines are.

The use of sound and video together greatly reduced the confusion experienced by (particularly new) players. The constant audio broadcasting of the sound servers kept everyone very much "in the loop" (as well as reducing the load on the limited Bat buzzer feedback channel) and players knew that, should they become confused, they could always retreat to their base and consult the ambient display.

3.3 The Mobile Phone Components

The final component consisted of commodity mobile phones, complete with colour screens, cameras, sound capabilities, and networking in the form of GPRS and Bluetooth[5].[3] It is likely that future phones will also incorporate some form of location-sensing technology. Such devices would make an attractive platform for running mobile location-aware games.

With this in mind, the third game configuration gave some players a mobile phone in addition to their Active Bat. The sound server and map display components from the above setup and are then run on the phones. A Bluetooth proxy bridges between SPIRIT and the phones, transmitting the relevant game events. This gives those players with phones a detailed map of their surrounding area (Figure 2).

The inclusion of the mobile phones into the architecture significantly changed the game dynamics. Those users with phones tended to act as the eyes and ears of their team, directing their team-mates, and coordinating play. The capabilities of the phone rendered the fixed map display and sound server redundant, allowing for a much more ad-hoc game (assuming a omnipresent network connection and location system). Interestingly we found that the game became *less* social if everyone had their own phone because then the players did not have to collaborate explicitly with their team-mates. It seems that using less hardware made the game more social. [4]

The increased feedback to the players with phones also reduced the requirements on the other components in the system. It could, for example, tolerate more latency before becoming confusing. Weapon selection could also easily be done on the phone, removing the need for gesture support and tolerating lower location-accuracy.

Using Bluetooth mobile phones increases the demands on the wireless network and expose the critical role of this particular connectivity method. We outline some important considerations when using Bluetooth in fast-paced location aware games.

The effective gaming area is now constrained to the coverage of the communication network. By using the location system, we made detailed signal strength coverage maps to accurately survey areas of acceptable reception. Figure 3 shows more than 30000 samples obtained by querying the master of the piconet for the perceived link quality of a connection to a single slave. The re-

[3] We used a number of Nokia 6600 phones for our experiments.

[4] In the same way that "hide-and-seek" requires less hardware in that it does not require the purchase of a games console.

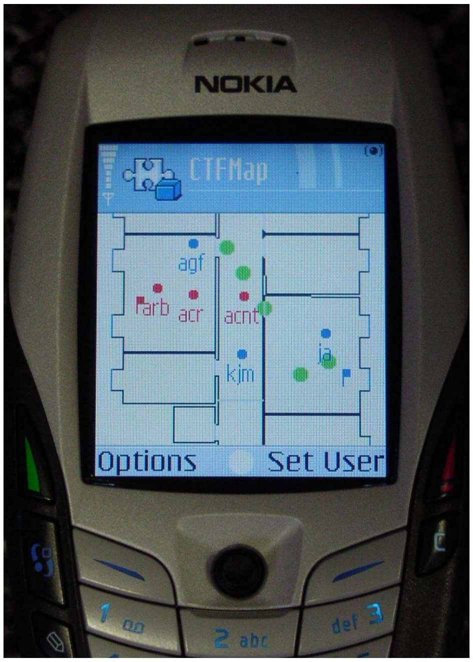

Figure 2: The game map displayed on a mobile device
The six small initialled circles are the game players. The larger (green) blobs are mines that have been laid. The flags are both currently in their bases at either side of the map.

sults show that a single class 1 Bluetooth device could not reliably maintain a usable ACL[5] link when more than 20 metres to a class 2 device typically found in mobile phones. We could overcome this by using high powered antennas or managing handovers between multiple Bluetooth devices.

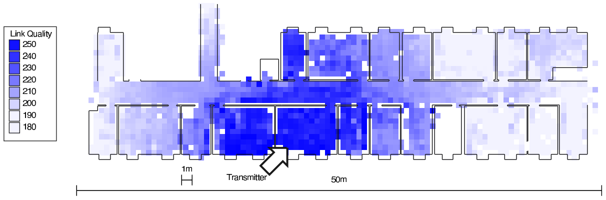

Figure 3: Bluetooth signal strength in an office environment
In this map the darker colours represent stronger signal strength. The occasional white spots in an otherwise dark area are due to inability to collect data because of obstructions (e.g. furniture) in the physical environment. (Grid size: 0.5mx0.5m)

Handovers are not well supported under the Bluetooth specification [2]. Traditional methods of using signal strength as a trigger for handovers do not work efficiently under Bluetooth because obtaining signal strength requires clients to establish and maintain a connection to the server. Such clients would regularly need to do inquiry scans to discover reachable servers, and in the process halt all Bluetooth traffic. In addition, most current mobile phones can only establish a single remote ACL connection, meaning multiple signal strengths cannot be measured simultaneously. Ritter et al [17] proposed a solution whereby a second wireless network is used to control Bluetooth handovers.

Latency plays an important role in networked multiplayer games as observed by Henderson and Bhatti [12]. Using our Bluetooth setup, we observed 20-40ms round trip times between the server and a single game client with an average of 34ms. However, there were significant variations in the scheduling behaviour of Bluetooth stacks when dealing with multiple connections. In particular, the Linux 2.4.24 Bluez stack exhibited extremely high latencies when multiple connections were attached. This appears to be an isolated bug, since tests on MacOS X and FreeBSD 5.2.1 did not show this behaviour. It does highlight the immaturity of Bluetooth stacks when operating under less common usage patterns.

4. EXPERIENCES

By varying the feedback capabilities of the system, as well as the latency and accuracy of the location system and network (as described in Section 3) we are able to report a number of experiences.

Feedback was critical to making the game playable and fun. The Bat alone (with just simple beeps as feedback) was insufficient. The *richness* of the feedback was low, and so relied heavily on the latency of notification to be meaningful. Sound servers made a great difference; more so than the ambient displays, largely because you did not have to divert your attention from the game to hear them. The combination of these into the mobile phone, where the display could be more easily consulted, made it more useful, and the improved *richness* of feedback made the players less concerned about any delay in notification of game events.

The fast-paced action games we prototyped made particular demands on the location system and network. The latency of location updates was critical to support interaction between players. Degrading this led to frustration and confusion, as the virtual world presented through the feedback channels became out of sync with the physical world.[6] The accuracy of the location system allowed for the use of gestures; this compensated greatly for the weak I/O capabilities of the mobile devices. This illustrates one of the tradeoffs between the components of the system.

5. RELATED WORK

The Global Positioning System (GPS) [13] is a highly-available outdoor location system accurate to around 30m horizontally with a latency of several seconds. Despite this relatively low accuracy and high latency, GPS has proved a popular platform for wide-area location-aware gaming. *Geocaching* [18] allows players to share the physical location of caches on the Internet and use GPS receivers to find and retrieve the contents of other players' caches. *Can You See Me Now?* [9] is a mobile mixed-reality game played on a city-wide scale. Up to twenty players on the Internet are chased across a map of a city by three performers running through the streets. *Human Pacman* [6] upgrades the classic Pacman arcade game with elements from ubiquitous computing, tangible user interfaces and augmented reality. Players wear headsets displaying their state in the game, and are tracked via GPS as they move around the Pacman grid. Physical objects

[5] Asynchronous Connection-orientated Link

[6] Savvy players would learn to shoot *behind* their target, as the virtual world lagged the physical one, rather than the normal behaviour of shooting in front.

act as "virtual cookies" that players pick up and interact with. In countries such as Japan, Singapore and Hong Kong where mobile handsets already support wide-area location information, games such as "Mogi, item hunt" [15], "Gunslingers" [14] and "Undercover" [19] are proving popular in dense urban areas.

Cricket [16] is an ultrasound and radio based indoor location system. Unlike the Active Bat, Cricket has no centralised control and uses fixed-position beacons and mobile listeners. Accuracy and latency suffer because listeners can only listen to one beacon at a time and the individual broadcasts cannot be efficiently scheduled. Balakrishnan et al describe their experiences porting Doom [3], a popular first-person shooting game.

Like the games presented here, *Pirates!* [4] merges some of the traditional social aspects of game-play by using the physical world as a component of a computer game. Players carry networked PDAs and RFID proximity sensors are used to associate physical objects with the game. Headon et al. built the *Active Floor* [11] which allows users to control games using suitable gestures e.g. jumping around.

6. CONCLUSIONS

This paper presented the design of a location-aware mobile game platform designed for fast-paced mobile games in the style of *Counter-Strike* and *Capture The Flag*. The platform was sufficiently flexible to allow experiments with different feedback mechanisms, location-accuracies and communications latencies.

These experiments revealed insight as to how weaknesses in one of these three components can be offset against strengths in the other two. In particular, fundamental tradeoffs between event latency and feedback bandwidth, as well as location-accuracy and device I/O capabilities were highlighted.

We investigated the suitability of Bluetooth as a communications network for low-latency, location-aware mobile gaming. Overall it showed promise, with sufficient bandwidth and an average application-to-application RTT of 34ms. However a number of critical bugs in some implementations were evident, particularly when handling multiple concurrent connections.

The fundamental issue of handover between Bluetooth interfaces remains to be addressed if games use a larger playing arena; we found 20m to be the practical limit in a typical office environment. Future work will also include more user studies to formalise the relationship between feedback, latency, accuracy and enjoyment.

Acknowledgements

This work was funded by AT&T Labs Research, Intel Research Cambridge, and The Royal Commission for the Exhibition of 1851. Many thanks to Michael Dales, Alastair Beresford, Marcelo Pias, Jon Crowcroft, and Richard Sharp for their valuable feedback.

7. REFERENCES

[1] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a sentient computing system. *IEEE Computer*, 34(8):50–56, August 2001.

[2] S. Baatz, M. Frank, R. Göpffarth, D. Kassatkine, P. Martini, M. Schetelig, and A. Vilavaara. Handoff support for mobility with IP over Bluetooth. In *25th Annual Conference on Local Computer Networks*, pages 143–154, November 2000.

[3] H. Balakrishnan, R. Baliga, D. Curtis, M. Goraczko, A. Miu, N. B. Priyantha, A. Smith, K. Steele, S. Teller, and K. Wang. Lessons from developing and deploying the Cricket indoor location system, November 2003.

[4] S. Björk, J. Falk, R. Hansson, and P. Ljungstrand. Pirates! Using the physical world as a game board. In *Interact, IFIP TC.13 Conference on Human-Computer Interaction*, July 2001.

[5] Bluetooth SIG. *Bluetooth Core Specification v1.1*, April 2003. `http://www.bluetooth.org/`.

[6] A. D. Cheok, S. W. Fong, K. H. Goh, X. Yang, W. Liu, and F. Farzbiz. Human Pacman: a sensing-based mobile entertainment system with ubiquitous computing and tangible interaction. In *2nd Workshop on Network and System Support for Games*, pages 106–117. ACM Press, 2003.

[7] G. M. Djuknic and R. E. Richton. Geolocation and assisted GPS. *IEEE Computer*, 34(2):123–125, 2001.

[8] Federal Communications Commission. Enhanced 911. `http://www.fcc.gov/911/enhanced/`.

[9] M. Flintham, S. Benford, R. Anastasi, T. Hemmings, A. Crabtree, C. Greenhalgh, N. Tandavanitj, M. Adams, and J. Row-Farr. Where on-line meets on the streets: experiences with mobile mixed reality games. In *Conference on human factors in computing systems*, pages 569–576. ACM Press, 2003.

[10] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999.

[11] R. Headon and R. Curwen. Movement awareness for ubiquitous game control. *Personal and Ubiquitous Computing*, 6:407–415, December 2002.

[12] T. Henderson and S. Bhatti. Modelling user behaviour in networked games. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 212–220. ACM Press, 2001.

[13] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer-Verlag, 5th edition, March 2001.

[14] Mikoishi Studios. Gunslingers. `http://guns.mikoishi.com/gunsSingTel/`.

[15] Newt Games. Mogi item hunt. `http://www.mogimogi.com/`.

[16] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *6th conference on mobile computing and networking*, pages 32–43. ACM Press, 2000.

[17] H. Ritter, T. Voigt, M. Tian, and J. Schiller. Experiences using a dual wireless technology infrastructure to support ad-hoc multiplayer games. In *Proceedings of the 2nd workshop on Network and system support for games*, pages 101–105. ACM Press, 2003.

[18] R. Webb. Recreational Geocaching: The South-East Queensland Experience. In *A Spatial Odyssey, Australian Surveyors Congress*, September 2001.

[19] YDreams Team. Undercover. `http://hk.playundercover.com/`.