High-Performance Pen + Touch Modality Interactions: A Real-Time Strategy Game eSports Context

William Hamilton, Andruid Kerne, Thomas Robbins
Interface Ecology Lab | Department of Computer Science & Engineering
Texas A&M University
{bill, andruid}@ecologylab.net, motasheron@hotmail.com

ABSTRACT

We used the situated context of real-time strategy (RTS) games to address the design and evaluation of new pen + touch interaction techniques. RTS play is a popular genre of Electronic Sports (*eSports*), games played and spectated at an extremely high level. Interaction techniques are critical for eSports players, because they so directly impact performance.

Through this process, new techniques and implications for pen + touch and bi-manual interaction emerged. We enhance non-dominant hand (NDH) interaction with *edge-constrained* affordances, anchored to physical features of interactive surfaces, effectively increasing target width. We develop *bi-manual overloading*, an approach to reduce the total number of occurrences of NDH retargeting. The novel *isosceles lasso select* technique facilitates selection of complex object subsets. *Pen-in-hand interaction*, dominant hand touch interaction performed with the pen stowed in the palm, also emerged as an efficient and expressive interaction paradigm.

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

General terms: Design, Human Factors

Keywords: bi-manual interaction; game design; eSports

INTRODUCTION

eSports is a popular cultural form of playing and spectating digital games. eSports play constitutes one of the world's highest performance HCI contexts. Skilled players practice in order to be able to perform rapid, precise, complex, and sustained interactions to compete. However, despite a shift among casual game players to new interaction modalities such as the Nintendo Wii and Microsoft Kinect, most eSports players continue to favor the keyboard and mouse.

In this research, we develop *Pen-in-Hand Command* (PiHC), pen + multi-touch modality interaction for the real-time strategy (RTS) games genre of eSports. We used the game mechanics and requirements of RTS games to provoke the design of embodied interaction. For detecting pen + touch, we used a Wacom Cintiq augmented with the minimum effort multi-touch interaction of ZeroTouch [18].

In the PiHC design process, we incorporated our experiences observing and playing RTS games, plus principles of direct manipulation, bi-manual interaction, and mode selection. Beyond prior techniques, such as marking menus, we developed new techniques for selection, multi-finger mode selection, and 3D manipulation. To enable real play, these techniques were composed to form a complete interface to an existing open source RTS game.

Through the design process, we developed techniques for increasing the efficacy of interaction with the non-dominant hand. One such is *edge-constrained multi-touch interaction*. We hypothesize that by leveraging physical characteristics of surfaces, such as their edges, we can anchor touch interactions to be performed quickly by the NDH. We also introduce *bi-manual overloading*, an approach to alleviating the difficulties of NDH targeting. Once a target has been acquired with the NDH, the interface should directly enable multiple actions, avoiding extraneous target re-acquisition.

We observed the emergence of the *pen-in-hand* interaction modality, in which the dominant hand alternatively interacts with fingers and stylus. The kinematic chain model postulates that, when performing bi-manual interactions, humans contextualize the dominant hand's action with the other [9]. This research extends the kinematic chain, applying its principles to pen-in-hand direct manipulation's division of labor.

To evaluate PiHC, we recruited RTS players and conducted a user study. These players are highly skilled at playing RTS games with mouse + keyboard. They are attuned to the game interface. They are skeptical of new interaction modalities. We observed participants play against one another. We use data to draw implications for design to support expert, high-performance pen + touch interfaces and new interaction modalities in general.

We begin this paper with interaction techniques and eSports background. We then present PiHC's interaction techniques, evaluation, and results. We synthesize implications for pen + touch interaction, and design for situated expert contexts.

RELATED WORK: INTERACTION TECHNIQUES

The design of PiHC draws from prior research in bi-manual interaction. We further address more specific work involving

UIST'12, October 7-10, 2012, Cambridge, Massachusetts, USA.

pen + touch and other new interaction modalities.

Bi-manual Interaction

Yves Guiard developed one of the first models of human bi-manual interaction: the *kinematic chain theory* [9]. The model addresses the relationships between the right and left hand for right handed people. This gets generalized as the dominant hand (DH) and non-dominant hand (NDH). The model has three components: the NDH defines the spatial context for action performed by the DH; the NDH initiates action before the DH; and the NDH performs coarser action than the DH. This model has been applied and investigated in many bi-manual interaction studies [5, 11, 20, 12, 13]. We apply kinematic chain principles in the design of PiHC. We also formulate a new hypothesis: *the kinematic chain extends further into the DH, where the fingers alternatively define the spatial reference, while the stylus is stowed in the palm, then hold and invoke the stylus, for precision action*.

Other research into bi-manual interaction has investigated the performance of the NDH performing Fitt's law tasks. Kabbash et al. confirm Todor and Doane's findings, that for the NDH, performance of Fitt's law tasks is significantly hindered more by target width than the DH [25, 14]. However, they also find that NDH is less affected by target distance than the DH. From these findings, we develop the design principle of physically constrained touch, in which the effective widths of targets for the NDH are increased by using a physical constraint such as the edge of a surface. Prior work has investigated constraining pointing targets to the physical edges of surfaces using mice and on small surfaces[1, 7]. We also develop approaches for minimizing situations in which the NDH must target new interface components.

Wobbrock et al. found that non-expert multi-touch users had trouble differentiating between gestures employing the same motion, like a swipe or tap, but a different number of fingers [27]. Other studies [6], as well as the command set built into Mac OS X, have found that using number of fingers for command differentiation is effective. In bi-manual overloading, we use number of fingers as a quick mode selection technique that minimizes retargeting with the NDH. Our study concurs: with minimal training, number of touching fingers is an effective technique for mode selection.

Pen + Touch Interaction

Pen + Touch interaction is a particular form of bi-manual interaction that seeks to leverage the advantages of both modalities while avoiding the pitfalls. Pen performs well for tasks that require precise path definition, such as inking, while touch is fast and direct, but imprecise. Several designs and studies have investigated the pen + touch modality [5, 3, 12, 13]. Pen + Touch interaction designs are often based on the metaphor of paper-based desk work; the NDH is used to manipulate artifacts, while the DH performs precise interaction using the stylus.

Prior work has investigated the application of pen + touch either in un-situated contexts, such as maze navigation [3], or in those involving electronic drafting tables [3, 12, 13]. The latter directly mimics paper-based desk work. As a RTS game, PiHC is situated in a new high-performance context, which we hypothesize will benefit from the direct manipulation afforded by touch and the high-precision of stylus-based interaction. Without direct correspondence to pen and paper, interaction design must be more abstractly derived. By working in this new context, we aim to develop generalized implications regarding the pen + touch modality.

Performance

Pen, multi-touch, and mouse + keyboard are fundamentally different interaction modalities. It would seem that direct performance comparisons are difficult, as few quantitative studies have achieved this. Odell et al. compared command selection performance across bi-manual marking menus, hotkeys, and toolbars [20]. While they found that hotkeys performed well, no significant differences in completion time were found versus a bi-manual marking menu. PiHC uses fluid in-context techniques, such as marking menus [16], to efficiently perform RTS game mechanics previously executed via hotkeys.

Kabbash et al. conducted an early study investigating the application of Fitt's law to the mouse, stylus, and trackball input modalities, with both the DH and NDH. Results indicated that stylus out-performed mouse on movement time. However, error rates were found to be slightly lower for the mouse [14]. Kurtenbach and Buxton find that for directional interactions, such as marking menus, stylus-based interaction outperforms the mouse [16]. More recent work suggests that for shape tracing tasks, people perform more accurately and quickly with a stylus than a mouse [28]. However, no statistically significant results were found. Brandl et al.'s study investigating performance of a maze task compared pen + touch, touch + touch, and pen + pen [3]. They found that the pen-based modalities outperformed the touch + touch condition. The design of PiHC focuses on leveraging stroke-based interactions, which in light of the prior work, seem to benefit from the stylus modality.

ESPORTS CONTEXT

We use the situated context of eSports as a resource for the design of interaction. *eSports* is the high-level play and spectating of digital games. RTS games are a popular eSports genre. Other eSports genres include first person shooter and arcade style fighting games. The community surrounding eSports is comprised of professional and amateur players, spectators, commentators, and game designers [4]. As with football, professional players are especially skilled. They generate income from tournament winnings, coaching fees, and streaming ads. More players are amateur, practicing skills at home, without pay, for fun and challenge.

eSports games between skilled players are broadcast live from major tournaments. These events are spectated by large co-located and online audiences. The Global StarCraft League (GSL) finals at Blizzcon 2011, Blizzard Entertainment's gaming convention, engaged 25,000 co-located and 300,000 online viewers. They watched two of the world's best StarCraft players compete for a $46,000 prize. We note that only 16,000 attended SIGGRAPH 2011, the ACM's largest annual event, with little live online participation.

Given the complexity of RTS games, it is necessary to play them in order to understand interaction and strategy. Some of the authors have extensive experience playing RTS games, such as Blizzard Entertainment's StarCraft 2 and Riot Games' League of Legends [24, 17]. To better understand high-performance interaction in RTS games, we watched extensive online videos of professional tournament play and explanations of how to perform at a high level [21]. Successful RTS players expertly invoke *game mechanics*, repeated actions that constitute a game's essence [22]. We draw from our experiences playing RTS games, and our observations of experts to form mental models [19] of the game mechanics of RTS eSports. We use this understanding to motivate the design of interaction techniques with new modalities.

eSports play is one of the most extreme interaction contexts that can be readily observed. As with football, players of varying degrees of skill are abundant and may be recruited for studies. We hope that our new interaction modality designs for eSports will create more engaging player and spectator experiences. Pen + Touch is a compelling interaction modality in this context, because it affords fast and precise interaction, the pre-requisites for RTS eSports. Before we present the design of PiHC, we discuss the basic interaction standards for RTS games and investigate eSport player's perceptions of new interaction modalities.

RTS Mechanics and Interfaces

At a low level, RTS game mechanics focus on managing *units*, game entities the player produces and controls. Players may command hundreds of units simultaneously, and make minute adjustments. High-level RTS players rely on reflexes and practice to efficiently perform the suite of mechanics.

RTS game interfaces are usually built around a main view of the game world with limited scope, but high-detail. The main game view is accompanied by a head-up display (HUD) with buttons and a low-detail overview, known as a "mini-map". The common interaction scheme in RTS games is to use the mouse to select units. The mouse is also used to either invoke commands for selected units.

The keyboard serves two primary purposes: activating commands for currently selected units, and accessing saved groups of units. While the mouse can be used to activate HUD buttons, hot-keys are faster, providing a competitive edge.

Using this scheme, players manage the game in macro using the keyboard, with minimal mouse input. This leaves the mouse free for micro-management: performing precise selections and commanding units. Through practice, players become extremely quick at executing game mechanics.

Perceptions of New vs Old Interaction Modalites

Prior play of RTS eSports is dominated by the mouse + keyboard modality. Players are quite skeptical of new interaction modalities; so much so that in 2011, Blizzard Entertainment's April Fool's joke was that they were releasing a version of StarCraft 2 playable with the Microsoft Kinect.

Before running our study, we surveyed 11 RTS e-Sports players to get a sense of how they perceived the potential of new interfaces for RTS eSports. These participants were recruited at a local eSports event. None participated in the later study. Eight of the players were were ranked between StarCraft's Master (Top 2%) and Gold Leagues (Top 20%); the other two did not play ranked games. When asked about Pen + Touch interaction for eSports, they expressd doubt. According to one Master League player: "A mouse and keyboard are very precise and fast. Pen and touch based input in my experience is slow, clunky, imprecise." Other players reported concerns about not having hotkeys, not being able to effectively left or right click, and slow or imprecise input sensors. One player even reported: "Everything in these games is about APM [actions per minute]; no pen or touch based inputs will ever be able to replace it."

Of the ten players surveyed, 80% reported that they thought Pen + Touch interaction in RTS games would hinder their ability to play the game at a high-performance level. However, one of the Masters League players reported that he thought it would *help* his ability to play. At the same time, he reported concern that "easier execution could make the game too easy, which could narrow the skill range." eSport players tend to be very proud of their ability to play at a high-performance level. This is to be expected, as their ability is hard-earned, developed over weeks if not years of practice. Another participant was distressed that pen + touch input would, "drastically change the instinctual feeling of basic game mechanics. I would be relearning EVERYTHING."

Some players thought that interacting with new interaction modalities would provide perceivable benefits. "It would allow you to input more complex actions." "It would encourage less rigid hand movements," and, "could lend itself to creative input schemes". Interestingly, one of the Masters League players reported that he would be excited about, "the difficulties I would have to go through." This indicates that for some high-level players, the challenge of learning a new interaction modality would be rewarding in itself.

SPRING & ZERO-K

Given the RTS eSports context, we present the building blocks of our apparatus. PiHC is built on top of the open source Spring RTS game engine [23], which provides the graphics, unit mechanics, and basic interface features required to create RTS games. We modified Spring to accept TUIO touch inputs [15]. These inputs are then passed through the engine to the game interface. PiHC is more specifically based on Zero-K [29], an open source game built using the Spring engine. We use the units and game mechanics defined in the Zero-K project, along with several existing interface components. We then built the touch and pen based inputs, by modifying Zero-K, adding new interface components, and using the touch and cursor events dispatched by the Spring engine.

PEN + TOUCH APPARATUS

For input, we use a 21" Wacom Cintiq tablet augmented with a ZeroTouch multi-touch sensor (Figure 1), enabling the simultaneous, distinct sensing of pen and touch interactions [18]. The Cintiq's pen sensing resolution is 5080 LPI, with 2048 pressure levels, and tilt sensing $\pm 60°$, at 145 Hz. ZeroTouch's finger sensing resolution is 600 DPI, with a refresh rate of 91 Hz. This sensor can detect up to 28 fingers. The resulting multi-modal resolution is substantially better than that afforded by Anoto (100-677 DPI and 256 pressure levels) + Diamond Touch (82 DPI) [3] and that of the custom

Table 1: Comparison of RTS game interaction techniques across modalities: prior mouse + keyboard vs. PiHC.

Game Mechanic	Prior Mouse + Keyboard	Pen + Touch in PiHC
Unit Selection	Box selection and single unit clicking.	Extended lasso and single unit tapping.
Command Selection and Invocation	Keyboard hot keys, default command right click, queuing with the shift key, and single-point click invocation.	Number of fingers command selection panel, queuing using scrubbed activation, formation drawing and tap invocation.
Macro	Keyboard factory selection through control groups and keyboard hotkeys for build commands.	In-context menu through direct and proxy selection.
View Manipulation and Selection	Screen edge panning, minimap dragging, control group double-tap, and saved came postion hotkeys.	Direct manipulation multi-touch, bimanual configurable viewpoints, and double tap or two finger press on control group button.
Control Groups	Number hotkeys, shift to add, and cntl to redefine.	Control group buttons. One finger to select/add, two fingers to see in game view and activate default command.

pen (“minimum pressure”) on Microsoft Surface 1 (43 DPI for both modalities). The Cintiq was mounted with its display surface tilted ˜30° from horizontal, at the edge of a desk.

INTERACTION TECHNIQUES

In the following sections, we motivate the design of techniques for pen + multi-touch modality interaction in PiHC. We draw considerations from prior interaction design work, from eSports RTS games, and our own experiences conducting formative user studies and play testing.

The mouse + keyboard modality affords quick performance of discreet actions. RTS games have been designed for these affordances in the extreme. While designing PiHC, we focused on maximizing the performance of pen + touch interaction techniques, while leveraging the fluidity and directness afforded by the modality. This fluidity enabled us to change the nature of several RTS mechanics, giving players more direct control of the game environment.

We designed for expert use. Some interaction techniques may not be considered easily recognizable or intuitive when first encountered; they were designed to be performed quickly after learning. This is the case for most RTS game interaction. Optimal RTS interaction is non-obvious. Most players must practice considerably to become competitive.

We develop new interaction techniques addressing five RTS game mechanics: unit selection, commands, macro, view manipulation, and control groups. For each mechanic, we discuss its importance in the RTS games context, present the traditional mouse and keyboard interaction, and discuss how it is performed by skilled players. We follow with the motivation and design of corresponding pen + touch interaction techniques. Table 1 compares interaction techniques across modalities for the RTS game mechanics.

Unit Selection

Selection is arguably the mechanic that requires the most precision in RTS games. In order to issue commands to units, they must first be selected; subsequent commands are issued to those units. Selection becomes difficult when units are moving or when they are densely packed.

Mouse In the majority of RTS games, selection is performed using box selection. Box selection is performed by specifying the diagonal corners of a rectangle, selecting the units circumscribed by this box. The first corner is placed at the first click location; the second corner is dynamically specified by dragging then releasing the mouse.

The area defined by the box is difficult to accurately define quickly. High level RTS players are known to ‘spam’ (invoke repeatedly at a very high rate, with little intended effect) boxing while playing in order to practice/hype themselves up. However, even with practice, selection remains difficult to perform, given that game elements are not axis aligned.

Pen In PiHC, we developed the *isosceles lasso select* technique that better suits many selection scenarios encountered during play. Isosceles lasso select is based on prior lasso selection techniques [26]. However, while the path of the lasso is being drawn, its area is extended by projecting a perpendicular vector from the midpoint of the line segment connecting the start and end of the lasso’s path (See Figure 2). The magnitude of the perpendicular vector, d, is one half the length of the connecting segment. Two line segments from the ex-

Figure 1: Cintiq augmented with ZeroTouch sensor.

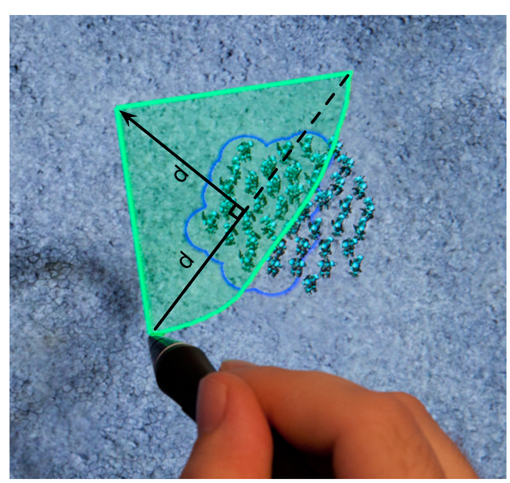

Figure 2: Isosceles lasso select technique.

tension point create an angle that subtends the current path of the lasso, forming an extension area that is an *isosceles triangle*. The directionality of the extension is defined by the user's most recent drawn path curvature. If the lasso end is directed toward the start point, the extension is removed.

Isosceles lasso select enables definition of a selection area with a precise and a non precise edge, reducing the distance of the pen stroke. Consider the scenario of selecting half of a clump of units (Figure 2). Using the leading edge of the lasso, a group may be split (along any direction). The extension selects which half to select by the curvature of the stroke. This contrasts with a normal lasso, which would have to be extended much farther to fully encompass the entirety of the selection. We note that a mouse may be used to perform this interaction. However, a stylus is more direct; it has been shown to more quickly and accurately define paths [28].

Command Selection and Invocation

After unit selection, command selection is among the most commonly invoked mechanics in RTS games. Units are issued one of many commands, including move, attack, repair, guard, reclaim, build, and harvest. Players quickly select and issue commands frequently throughout play, in order to micromanage or *micro* the actions of their units.

Keyboard + Mouse In mouse + keyboard play, commands are either selected specifically by using hot keys, or are automatically invoked as contextual default commands.

With mouse, a default command is issued by right clicking on a game world region. The command issued depends on what units are selected and what is targeted when the default command is invoked. If the cursor is over open ground, the default command is simply *"move"*. If over a unit, it is either *"guard"* or *"attack"*, depending on whether or not the unit is friendly. Non-default commands are issued using keyboard shortcuts, by pressing a key and left clicking the target.

Commands may also be queued, meaning that after finishing the first command issued, the selected unit will execute additional commands. This is useful for minimizing the number of tasks that must be micromanaged by the player. Queuing is enabled by pressing shift while issuing commands.

Pen + Touch We designed an edge-constrained multi-finger command selection panel for quickly issuing common commands (See Figure 3). This panel is positioned in the bottom left corner of the surface; it is approximately 19 x 6.5 cm. Players place a specific number of fingers in this panel to specify what action the stylus will perform upon marking. With no fingers in the panel, the pen performs isosceles lasso select. With a single finger in the command selection panel, the pen issues default commands to the selected units. With two fingers in the panel, the pen issues the *"fight"* command. Fight is frequently used in place of regular move to avoid losing units to unexpected enemy units. Finally, if three fingers are used, the stylus invokes a marking menu for selecting structures for construction units to build.

Formative user studies showed that 4 fingers were too difficult to use in the command selection panel. Spacing 4 fingers apart from one another, such that the multi-touch sensor can sense each individual finger, is difficult to reliably perform. This issue is common to many multi-touch sensors. It could be overcome with better blob / area recognition.

Commands are queued when one finger in the command selection panel is to the right of of the queuing activation panel division (Figure 4). This is performed by sliding all of the fingers to the right. We call this interaction technique *scrubbing*, where already placed fingers must simply be slid in an axis aligned direction in order to perform a gesture. This obviates the need to hit a specific target with the NDH.

For less-used commands, we placed an array of buttons above the command selection panel (Figure 3, top). To select a button's command, the player pushes it with the NDH. For buttons that only change state, this is sufficient. For commands that could be invoked multiple times, invocation is afforded by stylus tap or stroke on a target in the game world. These buttons are designed with the technique of Hinckley's springboard: holding the button queues the command [10]. When a finger presses a command button and releases before the stylus marks a target, then the command is issued right away.

When issuing commands with the stylus, instead of only defining a single target point, a stroke may be used, along which selected units will each be given an individual instance of the command (Figure 5). This gives the player more control of unit spacing and formation, a critical tactic. A mouse-based version of this functionality was previously developed by a Spring RTS developer; it is not commonly found in RTS games. The new interaction leverages path definition with the pen to enable more fluid and precise control of units.

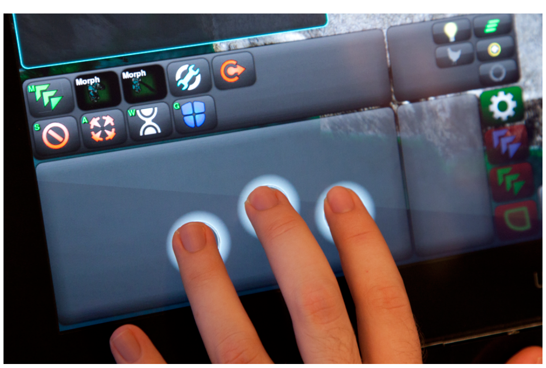

Figure 3: Edge-constrained multi-finger command selection panel. The player chooses how the pen performs by selecting a command by number of fingers.

Figure 4: Command queueing is activated when one finger is to the right of queuing activation panel division.

Control Groups

Control groups are a fundamental mechanic of RTS games. Instead of invoking the direct selection mechanic, control groups let the player save specific unit selections. A group may then be quickly selected, in order to issue commands to it. This saves the player the interaction time required to manipulate the game view to the units and select them directly.

Keyboard Each number key may be assigned to a control group. When a control group's key is pressed, all associated units are selected. If the key is pressed twice, the camera is moved to focus on that group. At this point the user may issue commands to its units. Units are added to control groups using the shift and ctrl modifier keys.

Control groups are used when 'performing macro' (described later). They are also used to control multiple groups of units on the front line. A player may assign all of one type of unit to a control group separate from the rest of his army, to easily invoke those units' special abilities.

Pen + Touch In PiHC, we developed a technique for defining, accessing, and directing control groups. We present a vertical array of buttons, associated with each control group. In order to add units to a control group, the player touches the control group button with one finger, then selects those units using isosceles lasso select. To reset a control group, the player taps its button with the stylus.

When two fingers are used to touch a control group button, the camera automatically moves to focus on it. Interaction performed with the pen, while two fingers are held down, invokes the default command for that group of units. The control group buttons may also be scrubbed, as described for selecting commands. This supports rapidly changing between different views of the game world, and fast control of several control groups at once. Again these control groups are edge-constrained to the left side of the Cintiq tablet, using its physical properties to facilitate NDH targeting.

Macro

In RTS games, 'macro' refers to tasks that a player performs in order not only to control front line units, but also to construct and manage units and buildings in his base. Thus, macro is not macro-management in a strict sense, but really simultaneous micro-management of many elements. Efficient macro is one of the most difficult interactions for new players to master, because it extensively relies on hot-keys and is difficult to observe other players performing.

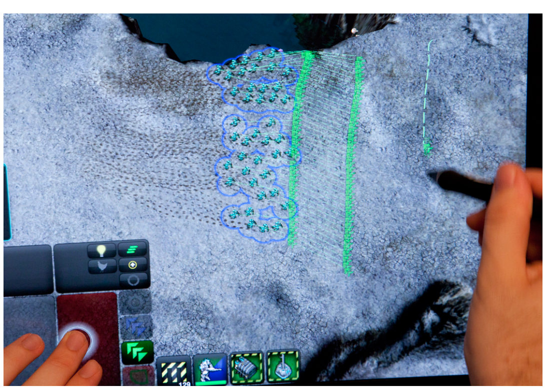

Figure 5: Commands can be placed along a path using the pen. This enables defining unit formations, giving players a higher degree of control over unit placement.

Keyboard One of the most frequently invoked macro tasks is issuing unit construction commands to factories (buildings that produce units). This must be done by first selecting the factory, and then issuing the construction command by either clicking its button, or pressing the hotkey to select the factory's control group. Skilled players almost never use the mouse to perform this task; it is performed via hotkeys because of the speed they afford.

Here is an example macro key sequence in Blizzard's StarCraft. Press '4', to select control group 4, commonly used in StarCraft to select a set of construction buildings, such as a barracks. Once selected, the player builds a 'marine' by pressing the 'A' key. This interaction, once mastered, is quick to execute. It is easy to interweave with other interactions, such as micro of the front line. The keys are pressed without having to change the main view of the screen.

Touch + Pen To support performing macro in PiHC, we developed an in-context touch menu for issuing building commands to factories. To activate the menu, the player touches the building in the game world. The menu appears above and to the right of the touch, avoiding occlusion (Figure 6). The menu interaction is designed so that the NDH brings up the menu, leaving the DH free to interact with the menu using either the stylus or fingers. Thus, menu interaction follows the kinematic chain model: the NDH defines a context for precise DH interaction.

To support the performance of macro while the camera is focused on any location in the game world, the in-context menu may be activated by touching a building's icon in an array at the bottom of the screen (Figure 7). This array includes icons for each of the player's factories. Like graphical hotkeys, this lets the player interweave macro with other mechanics, and saves the player the interaction time required to change the game view to the building's location.

To support issuing various commands to a selected factory, we *overload* the in-context menu with several sub-modes. To interact with all buildings of the same type, a second finger is placed next to the first. Now, the player can issue commands to all factories of this type using the in-context menu. While using the in-context menu, the stylus has several functions.

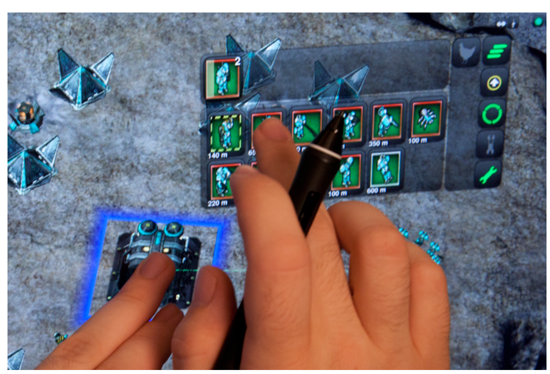

Figure 6: To issue build commands to factories, the NDH is used to activate the in-context build menu.

Figure 7: Macro may be performed at any view location by activating the in-context menu using the factory icons edge-constrained along the bottom of the display.

It can be used to press the menu's buttons. As the Cintiq can track the stylus hovering over the screen, the stylus can bring up tool-tips for items in the in-context menu (a very basic form of scaffolding). While the in-context menu is activated, the pen can also issue movement commands to the selected factories, enabling setting *rally points* (locations for finished units to move to). By overloading the in-context menu, we aim to reduce the number of targets the NDH must seek.

View Manipulation/Selection

Manipulation of the game world view (camera) is a primary action in RTS games. The position of the camera directly determines what the player can see transpiring in the game world, and at what level of detail. Camera positioning also affects how quickly a player can react to game events.

Mouse + Keyboard The most common camera-positioning technique in RTS games is to move the mouse to an edge of the screen. The game world then gradually pans in the direction of that edge. This interaction has been decried by expert players [21]. They argue that the interaction is slow when you want to move the camera view to a distant location in the game world. It meanwhile prevents the mouse cursor from be used to perform other important interactions. Further, such screen panning can be difficult to perform accurately; it often results in over- or under-shooting a target.

Camera position can also be manipulated in ways more accepted by expert players. The camera is quickly moved to look at a control group by double tapping that control group's assigned key. The player sets predefined camera positions by assigning hotkeys. Another option is to manipulate the camera view by left clicking on the mini-map; this moves the camera to focus on the clicked point. The camera view is also moved by dragging on the mini-map. This last approach is commonly used among skilled players, because its directness affords fast and accurate map manipulation.

Touch In PiHC, the player manipulates the camera directly using touch in the game world. The view is panned using a single finger, zoomed/rotated/panned using two fingers (Figure 8), or pitched by using a three-finger pinch gesture. This interaction may be performed with either hand, and is much more direct than any of the approaches previously implemented in mouse + keyboard RTS games. We note that these gestures may be performed easily using pen-in-hand interaction, with the fingers of the DH, while the stylus is stowed in the palm. This interaction technique emerged early in our design process, through informal use.

Players save and recall camera positions using the viewpoints control. A series of buttons can be touched to target the camera on preset locations. Each button is set by sustained press with the NDH while manipulating the camera view with the DH (Figure 8). When the button is released, it remembers the current camera position. As its icon, the button retains a small representation of its viewpoint (Figure 8, upper left).

Quickly changing the camera viewpoint to observe multiple game world locations is a practice commonly performed in RTS games to maintain awareness. Camera viewpoint buttons can be scrubbed to quickly navigate to preset views. They are edge-constrained in the display's upper left hand corner, so that the NDH can hang off the top edge of the screen, with the thumb touching the control. Again, this edge constraint of the tilted display makes the control easier to target with the NDH, while visually focusing on the main view.

EVALUATION

We conducted an evaluation in which RTS players came to our lab to play PiHC. We recruited 10 participants that reported having previously playing RTS games. All participants were male, university students. Four pairs of participants were friends. All participants were right handed. They were compensated with pizza.

Before the study, we asked participants to fill out a questionnaire that asked about their experience with RTS games and new interaction modalities. After completing the questionnaire, we had them read several pages explaining basics of Zero-K, including how the game economy works, and descriptions of unit types. We conducted a tutorial demo of the pen + touch interface. After this tutorial, we gave participants 15 minutes to try out and learn the interface in a game without enemy units. The proctor answered players' questions during the tutorial.

After the tutorial, players were asked to play against one an-

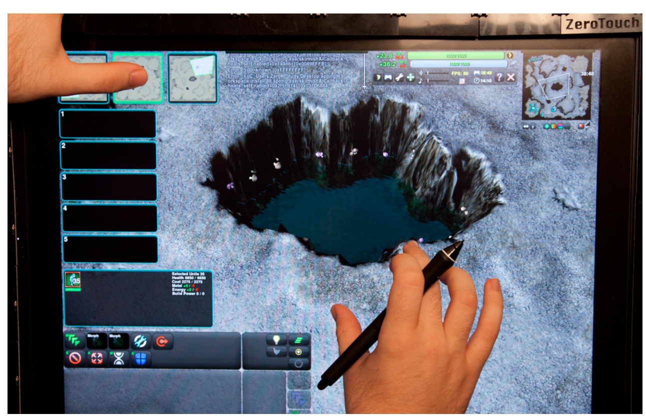

Figure 8: Different game views are saved using the edge-constrained configurable viewpoints control in the display's upper left. A player holds down a viewpoint button with the NDH, manipulating its saved viewpoint with the DH.

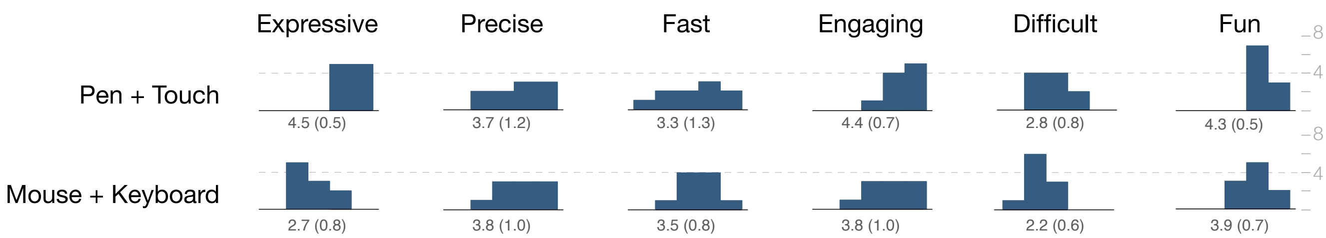

Figure 9: Participant experiences survey distributions. Means and Standard Deviation are given below each distribution.

other for 4 games. We video recorded each game. A high-resolution screen-capture of the game interface and an overhead camera view of the player's hands interacting with the sensor were simultaneously recorded. The independent variable was interaction modality: a player used either mouse + keyboard, or pen + touch.

During the first game, each participant played with pen + touch. In the middle two games, players alternated modality conditions. In the final game, both participants played using mouse + keyboard. We decided not to counter-balance the condition order. Because these players already had extensive experience interacting with mouse + keyboard, we thought it best to engage them with the new interaction modality after the tutorial, to create continuity of kinesthetic memory. After playing, participants completed a post-questionnaire.

RESULTS AND DISCUSSION

Despite the findings of our previous survey, the participants were excited about the Pen + Touch interaction modality. We conducted a post-questionnaire survey of participants' experiences of the precision, speed, engagement, difficulty, expressiveness, and fun of the pen + touch, and mouse + keyboard conditions, with a value of 5 for strong agreement and 1 for strong disagreement. The average values reported are shown in Figure 9. A Wilcoxon signed-rank test reported significant differences between the conditions for fun, engagement, and expressiveness ($p < 0.05$). We note that for difficulty, speed, and precision there was more variance in reported ratings: different participants reported strong preference for one condition or the other for these characteristics.

Participants were also asked which condition they preferred in terms of map control, unit control, game awareness, unit selection, and factory control. The results are shown in Figure 10. Unit selection with mouse + keyboard was favored strongly. However, participants' qualitative responses indicated that there was a significant issue with the present implementation of single unit selection. We observed that when players tapped a unit, the tip would travel several pixels between stylus down and up events. This accidentally activated isosceles lasso select, which often failed to select the unit in the small selection area. Many players reported that for group selection, they found that isosceles lasso select was more effective for selecting groups and subgroups of units.

All but one of our participants reported that they felt they could become as good at playing RTS games with pen + touch as they are with mouse + keyboard. Two participants, one of whom was a platinum level StarCraft 2 player, reported that they were already as good with the pen + touch modality by the end of the study. In the following sections, we report on specific qualitative findings concerning the practicability of different interaction components and players' overall engagement with the game.

Engagement

Players reported being more engaged with the game in the Pen + Touch condition. This was found in the Likert scale questions, and also across qualitative data. One participant reported that the experience felt, "more engaging on a visceral level," and that, "somehow the level of control felt more 'real' ". Others reported similar feelings of realness and more physical identification with the game world.

We suspect this is due to the direct nature of the Pen + Touch modality. Several participants reported that the manipulation of the map helped them stay closer to the action. One participant mentioned that he thought he would be more likely to watch someone else playing a RTS game using the Pen + Touch modality, because he would be more engaged watching the physical interactions of another play.

Thus, we see that interaction with new modalities has the potential to transform the experience of engagement in eSports. Part of the spectacle of eSports is being able to observe the skill of high-level players [4]. Because the current mouse + keyboard interactions for RTS eSports are so indirect, the extraordinary skill that high-level players possess can be hard to observe and fully appreciate. The directness of physical interaction in the Pen + Touch modality makes it much easier to connect physical movement and performance. Thus, this aspect of the interaction stands to dramatically augment player and spectator eSports game experiences.

Practicability

Our interaction techniques address efficiency and expression for expert users. Expert use is characterized by extended practice and experience. Our study only lasted for several hours. Meanwhile, our participants reported that they play

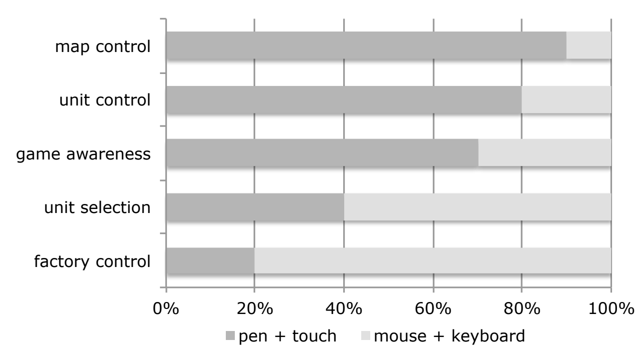

Figure 10: Task performance condition preference.

RTS games with mouse + keyboard for an average of 11.5 hours per week. That constitutes a significant experience level gap. Nonetheless, all but one of our participants felt that with practice, they would become at least as good at playing RTS games using a pen + touch interface.

It is interesting to note that, when asked what they would specifically have to practice to improve their performance, many participants reported that they would practice the in-context build menus for factories. This was the least preferred interaction component (Figure 10). However, no participants reported that they felt they could not practice this component to develop comparable skill.

We observed that few participants used the control group and viewpoints interface components. Typically, these are more advanced skills for RTS players. They are optional and, when practiced, improve overall performance. Several participants noted that they would practice these interactions. Again, none reported that they felt they could not improve their performance by practicing these skills.

IMPLICATIONS FOR DESIGN

Our design process, techniques, and evaluation develop implications for designing pen + touch, and more generally, bi-manual interactions. We first discuss our understanding of pen-in-hand interaction, an emerging paradigm of DH interaction with the Pen + Touch modality. Our experience designing NDH techniques has also led us to a new understanding of how to optimize interaction to minimize the impact of inherent NDH motoric impediments. Finally, we discuss our experiences conducting evaluations with eSports players and present lesson's learned about working with these experts.

Pen-in-Hand Interaction

We first observed DH *pen-in-hand interaction*, touch interaction performed with the pen stowed, early on in informal testing sessions. While we originally intended the NDH to perform map manipulations, as prescribed by Guiard's kinematic chain model, we found ourselves and others manipulating the map using the DH. This is done with the pen stowed between the fingers (see Figure 8). Hinckley et al. observed that this type of interaction is commonly performed when working on desks with pen and paper [13].

We found that Pen-in-Hand interaction increases the capacity of the DH to perform complex interactions. Across the 40 game play sessions of the study, we observed zero instances of a player dropping their stylus. We noticed various stowed positions for the pen. Sometimes, the pen was only moved out of the way for a single finger to pan the map or press a button. At other times, the player would fully stow the pen to perform more complex touch gestures, such as the two or three finger map manipulations.

This data shows us that Pen-in-Hand interaction is particularly useful for direct manipulation. The kinematic chain model is extended further to describe the relationship between the fingers of the DH and an implement held in that hand. Our study indicates that the fingers of the DH effectively define the spatial reference of the held implement, which may be stowed during interaction. It appears that the fingers and stylus have equal priority for interaction order, and may be easily interwoven during interaction.

Edge-Constrained Affordances & Bi-manual Overloading

It has been shown that targeting is more difficult to perform with the NDH [14]. We strove to significantly reduce this problem in two ways. First, we used physical characteristics of the display to *edge-constrain* NDH interactions. Second, we used *bi-manual overloading* to reduce the total number of times that interaction required NDH retargeting.

Designers should place on-screen NDH affordances near a surface's edges and corners. Edge-constraining interactive controls effectively increases target width. The edge of the screen is used as a tactile feedback mechanism for the acquisition of the region. The hand can also grasp or be rested on the edge of the screen to alleviate physical strain from continuous support of the arm. New display technologies, such as curved displays, may offer additional physical properties and tactile feedback to exploit in interaction design.

Another approach to alleviating the difficulty of targeting with the NDH is to simply minimize the number of instances in which new targets must be acquired. We achieve this using the bi-manual overloading design paradigm, which holistically suggests to reduce the number of times that the NDH must retarget. When specifying a mode using the DH, by touching a target, it is best to *overload* the target by providing a number of interactions, likely to be performed together in sequence, that may be performed without retargeting.

An example of bi-manual overloading in PiHC is the in-context build menu. Once a factory has been targeted with the NDH, there are a number of actions that may be performed without moving the NDH. The player may issue build commands to the factory, select all factories of that type by placing a second NDH finger down, and place a rally point using the pen on the game world. All of these options are relevant to the context of the factory and don't require retargeting with the NDH. One of our participants reported that they found this interaction "very smooth and quick".

We note that despite the findings of Wobbrock et al., number of fingers is a very direct way to implement bi-manual overloading. Our experience is that placing a specific number of fingers is a sufficiently coarse action that it may be performed robustly by the NDH, while demanding minimal visual attention. Another approach to bi-manual overloading is the use of *scrubbing*, movement of a finger or fingers of the NDH between multiple axis-aligned interface components without lifting the NDH. Scrubbing avoids retargeting the NDH. The fingers are simply moved in a pre-specified direction. This performs particularly well when feedback can be visualized within the current visual focal point, as when the control group and viewpoint controls are scrubbed. We note that by constraining to an axis-aligned direction, scrubbing requires significantly reduced interactive bandwidth, as prescribed by Balakrishnan and MacKenszie [2]. We utilize scrubbing extensively in the design of the command selection, viewpoint selection, and control group NDH interaction. One participant particularly noted that, "my eyes never had to be averted from the place I was originally looking at".

Situated Evaluation With eSports Players

Situated evaluation with eSports players has many benefits. The foremost is the recruitment of enthusiastic expert participants. All our participants had existing knowledge of how to play RTS games at a relatively high level of performance.

However, we encountered several unforeseen difficulties in our study. The skill levels of the recruited eSports players varied. Even when recruiting participants who were friends and knew they would be playing against each other, almost every pairing of participants resulted in unbalanced skill levels. This makes empirical data from head-to-head matches, such as wins and losses, difficult to interpret, given that the conditions are confounded with the players' prior skill levels. Further, it also makes participation less fun. Few players wanted to stomp or be stomped by their friends.

We found that, given this difference in skill level, another phenomenon emerged during game play. The stronger player would assure their victory early in the game, and would then hold back, not attacking to finish the game. While toying with their opponent, the player would spend time exploring the interface. While this resulted in more observable interaction performance, it also meant that the players were not pushing themselves to perform at the highest level possible.

In future studies, we plan to engage participants in a tournament style study to engage them in more heated play for longer periods of time. We did find that having participants play in pairs sparked interesting discussion and helped players learn more quickly. We may also attempt to have participants play cooperatively against AI opponents.

CONCLUSION

Throughout the design of PiHC, we have encountered many interaction challenges that have pushed us to evolve our understanding of the pen + touch modality. The RTS eSports context has requirements that motived higher performance design. From these requirements comes new designs and implications for new interaction modalities. The isosceles lasso select technique stands as a novel hybrid approach for quick and precise selection. We also developed approaches to enhance NDH interaction using the physical characteristics of surfaces, by edge-constraining affordances, and by overloading NDH interactions. We were excited to see the emergence of the pen-in-hand modality as efficient and expressive DH interaction approach.

Our findings can be generalized in terms of the theory of affordances. Physical properties of objects are perceived, and thus become actionable [8]. We are working with a new form factor for pen + touch. The surface is neither table, nor wall, neither enormous, nor tiny. It is mounted mechanically, and so, unlike a phone or tablet, does not require one hand to hold it. The new physical properties of the surface redefine what can be actionable. The size, proximity to the user, and 30° tilt of this pen + touch surface afford reaching around, and grasping the edges. The constraint of the edge orients the NDH, facilitating the design of new surface-situated interaction techniques involving touch with NDH fingers.

Future research will further investigate edge-constrained NDH interaction, isosceles lasso select, bi-manual overloading, and other affordances and techniques. Their emergence through design for eSports promotes the value of investigating new techniques in this and other situated contexts.

REFERENCES

1. Appert, C., Chapuis, O., Beaudouin-Lafon, M. Evaluation of pointing performance on screen edges. *Proc. AVI* (2008), 119–126.
2. Balakrishnan, R., MacKenzie, I. S. Performance differences in the fingers, wrist, and forearm in computer input control. *Proc. CHI* (1997).
3. Brandl, P., Forlines, C., Wigdor, D., Haller, M., Shen, C. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. *Proc. AVI* (2008).
4. Cheung, G., Huang, J. Starcraft from the stands: Understanding the game spectator. *Proc. CHI* (2011).
5. Cutler, L. D., Fröhlich, B., Hanrahan, P. Two-handed direct manipulation on the responsive workbench. *Proc. I3D* (1997).
6. Damaraju, S., Kerne, A. Comparing multi-touch interaction techniques for manipulation of an abstract parameter space. *Proc. ICMI* (2011).
7. Froehlich, J., Wobbrock, J. O., Kane, S. K. Barrier pointing: using physical edges to assist target acquisition on mobile device touch screens. *Proc. SIGACCESS* (2007), 19–26.
8. Gibson, J. J. *The ecological approach to visual perception*. Psychology Press, 1986.
9. Guiard, Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *J. Motor Behavior 19* (1987).
10. Hinckley, K., Guimbretiere, F., Baudisch, P., Sarin, R., Agrawala, M., Cutrell, E. The springboard: Multiple modes in one spring-loaded control. *Proc. CHI* (2006).
11. Hinckley, K., Pausch, R., Proffitt, D., Patten, J., Kassell, N. Cooperative bimanual action. *Proc. CHI* (1997), 27–34.
12. Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., Buxton, B. Manual deskterity: An exploration of simultaneous pen + touch direct input. *CHI EA* (2010).
13. Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., Buxton, B. Pen + touch = new tools. *Proc. UIST* (2010).
14. Kabbash, P., MacKenzie, I. S., Buxton, W. Human performance using computer input devices in the preferred and non-preferred hands. *Proc. CHI* (1993), 474–481.
15. Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E. TUIO: A protocol for table-top tangible user interfaces. *Proc. 6th Int'l Gesture Workshop* (2005).
16. Kurtenbach, G., Buxton, W. The limits of expert performance using hierarchic marking menus. *Proc. CHI* (1993), 482–487.
17. *League of Legends*. [PC]. Riot Games, 2009.
18. Moeller, J., Kerne, A. ZeroTouch: An optical multi-touch and free-air interaction architecture. *Proc. CHI* (2012), Full paper in press.
19. Norman, D. A. *The Design of Everyday Things*. Basic Books, September 2002.
20. Odell, D. L., Davis, R. C., Smith, A., Wright, P. K. Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. *Proc. GI* (2004).
21. Plott, S. Day[9] Daily #242 Secrets of Hotkeys, APM, and Mouse Movement, February 2011.
22. Salen, K., Zimmerman, E. *Rules of Play: Game Design Fundamentals*. MIT Press, Cambridge, MA, USA, 2004.
23. *Spring Engine*. [open source], 2004. http://springrts.com/.
24. *StarCraft II*. [OSX]. Blizzard Entertainment, 2010.
25. Todor, J. I., Doane, T. Handedness and hemispheric asymmetry in the control of movements. *J Mot Behav 10*, 4 (Dec. 1978), 295–300.
26. Wills, G. J. Selection: 524,288 ways to say "this is interesting". *Proc. INFOVIS* (1996).
27. Wobbrock, J. O., Morris, M. R., Wilson, A. D. User-defined gestures for surface computing. *Proc. CHI* (2009).
28. Zabramski, S. Careless touch: a comparative evaluation of mouse, pen, and touch input in shape tracing task. *Proc. OzCHI* (2011), 329–332.
29. *Zero-K*. [open source], 2010. http://zero-k.info/.