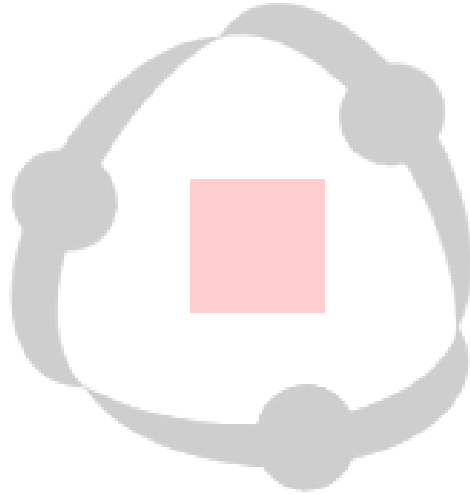


PitA-Board

YEAR END PROGRESS REPORT 2001



Henry Kyle Bygott
Henry.Bygott@colorado.edu
Undergraduate Research Apprentice
L³D

1/03/01

SCOPE

The PitA-Board (Participate In The Action Board) is a human-computer-interface with primary goals of increasing simultaneous interaction and group collaboration. It does this in a system utilizing simulations and dynamic information spaces to provide transparent access to problem solving resources in order to increase user understanding of the problem being dealt with.

INTRODUCTION

The PitA-Board is designed to be a human-computer interface that is much more transparent and easy to use than a typical keyboard and mouse (or even touch-screen) system. It is an improvement to the action space in the Envisionment and Discovery Collaboratory (EDC). The PitA-Board is, at its core, a chessboard sensor net designed by DGT Projects. This sensor has the ability to detect the type of any chess piece (plus 3 unreleased pieces – a total of 15) placed in any one of the squares (64 per board). Using these sensor nets and the tags from within the chess pieces, the PitA-Board was designed using the Smalltalk programming language in Squeak.

Using the PitA-Board allows information to be seemingly imbedded into specific pieces. In the earlier EDC design, a user had to understand interface issues in the system to use the system effectively. In order to place a house, the user has to click draw, click house, and click where the house is to go. With the PitA-Board system all one has to do to place a house is pick up the house model and place the house wherever it is needed. Although the house model actually has no ability to store information, it does have one very important characteristic- a resonant frequency identified by the system as #8. This allows the PitA-Board to determine the piece's identity as a house, and take appropriate action.

The PitA-Board also allows for simultaneous interaction much better than the earlier version of the EDC. The previous EDC utilized a touch screen that only allowed one user to interact at a single time, if two tried to use the screen simultaneously it would place the touched point somewhere between the two users. The PitA-Board scans one square at a time, identifying pieces in that square. This means that any number of users can be using the PitA-Board at once, even using same pieces in certain cases. This is a great stride forward in simultaneous interactions.

PROGRESS SUMMARY

Since the start of my work on the project in early June 2001, the PitA-Board has continually accelerated in its advancement and has made great strides. A two-board system and a 4-board system have each been created. Many new interaction techniques have been created, notably the "administrator" piece, the "tool" piece, and the directional piece. New functions have been implemented in the menu system, the warning system and the fade system. The addition of bus stops into the simulation added a unique type of "stamp" piece, with certain constrictions on bus stop placement.

Recently a new group was formed consisting of Hal Eden, Tomohiro Oda, Eric Scharff, Jack Elston and myself, with additional input from Stefan Carmien. This subgroup is each taking a part of the PitA-Board for investigation. My role involves investigating the use of PDA's and wireless interactions, and currently I am involved in the abstracting the "nuts and bolts" of PitA-Board usage away from the simulation using it. Although this changes little visibly, it is a large and important change in the operation and usability of the system. Following is a more in depth look at some of this progress.

PROGRESS SPECIFICS

- Investigation of close proximity boards
- Stacking Boards to increase resolution
- Theoretical board design and Prototype
- Bus interface programming and interface with Hal's code
- Time-out, re-sync and statistics features
- Initial uses of the 3D Genisys Printer
- The tool piece
- On-Board Menu
- Color-coded bus-routes, stops and buses
- Multi-Route stops
- Orienting pieces – Directional Buses
- Fade feature
- Changing Houses
- Logo Design
- Preliminary Investigation of PDA usage
- Creation of PitaController
- Creation of Test System

Investigation of close-proximity boards

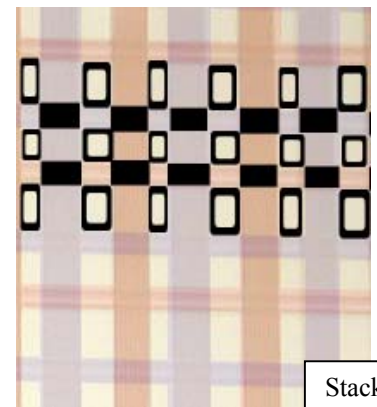
In testing methods of having two of the DGT Electronic Boards in extremely close proximity (to simulate one large board), it was found that folding the boards back under themselves was most effective and the least prone to glitches. Glitches in this case are when the opposite board erroneously detects a piece placement. Glitches only occur when a piece is placed on the column of a board closest to the other board, always occur on the opposite board, and always normalize by the next poll of the board. Although the boards would incorrectly sense pieces on their folded edge, using metal tape solved the problem. This changed the resonance properties of that spot, and rendered the board unable to detect any pieces on it.

An interesting note is that glitches seem to be highly dependent on the type of piece used. When testing methods for attaining a stable close proximity between boards, a #10 (Black Bishop) tag was used. It was later discovered that pawn pieces (#1 and #7) are very prone to glitches. This makes the selection of purposes for certain pieces an interesting question.

Stacking Boards to increase resolution

This idea, although initially very interesting, is a disappointing one. Due to the dead-space on a board, although stacking boards does essentially quadruple the resolution, it halves the detection area. If a piece is placed on one board's dead spot, the piece will be in an unknown location throughout the created cross, its location determined only by the other board. This makes the grid used non-uniform, and the idea was abandoned.

In the image at right, red represents one board's dead spots, while blue represents the other. It is apparent how the alignment of these spots creates the solid black spots representing a completely dead board, and the crosses in which position is indeterminable.



Stacked boards

Theoretical Board design and Prototype

A prototype 2-board system was designed for investigation into the construction techniques and usability of a folding PitA-Board system. It was the base system used for reference and critiquing during the design of the final model, created by Hal.

Bus interface programming and interface with Hal's code

Interfacing with the chessboards in bus format is slightly different from in single-board mode. Namely, bus interfacing requires polling, whereas single boards can just update as changed occur. The basic communication engine of the Squeak program for the chessboards was completely rewritten to allow the changes. Once basic communication was worked out, some prime methods were rewritten to allow for multiple boards. The most important was the re-working of the "spaceToLocation" method. This is a method that when given a space index number determines the coordinate for proper graphic placement by other methods. A redefinition of this method and some basic protocol allowed multiple boards to be used.

Time-out, Re-sync and Statistics features

Time-out and Re-sync were added to increase the robustness of the code, which is designed for anybody to operate effectively, regardless of computer skills. Previously the entire program would freeze if a board were unplugged, waiting on a next byte that would never arrive. A time-out feature was added to avoid this, set to 80ms as suggested by dgtbrd13.h documentation. Thus the program will continue if a board fails.

The re-sync function is designed to keep lost data to a minimum. Under the polling design, a board will send changes, but if that message is lost the board has no knowledge of the loss and the changes are lost at the time of the next poll. Re-sync tells the board to repeat the *last* set of changes, in an attempt to minimize loss of data. With these features, boards can be disconnected, or even unplugged for any amount of time, and will still continue to function properly when reconnected.

The first "administrator" feature was added in the statistics feature. This is not needed for the average user, but provides useful information to a knowledgeable person. It is a small box in the lower right corner that displays total polls, polls per second, number of time-outs, number of re-syncs attempted and number of re-sync fails for each board.

The 3D Genisys Printer

The 3D printer in the ITLL was utilized to create real pieces, and eliminate the need for chess-pieces with paper wrapped around them. Models are fairly simple to create in Solid Works and then can be printed directly using the Genisys printer. The use is comparatively expensive at \$7.50/hour. Hollowing out pieces and using cheaper material such as Styrofoam to fill the models is a very important technique. So far two models were created to test the use of the printer: a simple house model, and a more complex bus model. The bus has features ranging down in size to .01", which the printer handled well. Drawbacks to the printer include the texture left by its method of printing- small horizontal stripes across the entire model and occasional loss of small (especially hole or indent) features due to the printer created support structures for overhangs above.

The Tool piece

With a change in implementation allowing houses, stores, and all objects in general to be their own class, it became possible for them to easily store any private data needed. To demonstrate this capability a "tool" piece was created. This piece allows editing of objects. Dependant upon what type of object is clicked the tool piece takes varying action, doing nothing to stores but allowing house type to be selected (see "Changing Houses") or a bus stop's route to be identified (see "color coded routes and stops"). It generally interacts through an on-board menu (see below). This is the first piece to invoke different behavior dependant upon what type of object it is used upon.

On-Board Menu

The previous system of menus, with pop-up dialogs requiring keyboard input was completely deleted, replaced by a menu system that actually appears on the chessboard itself. The menu items could then be selected using the same tool piece that invoked the menu. If the

tool piece is clicked anywhere else on the board the menu treats it as a “cancel” and the menu disappears without any action taken.

Color coded routes and stops

To lessen confusion between separate bus routes, code was added so that each route had a new color. Bus stops were then implemented so that the flag of the stop matched the color of the route it belonged to. The Bus code itself was then changed to allow the bus to match the color of its route. This will help keep confusion to a minimum where multiple routes are concerned. The tool piece was augmented to allow selection of which route a stop belongs to.

Multi-Route stops

With the ability to select what route a stop belongs to, the next obvious step was to have stops able to serve multiple routes. This feature was added through the tool piece. When a bus stop is edited it finds all near roads and gives a selection of each individual route or any combination of two routes. The simple combination of two routes was selected for the time being over any combination of multiple routes in order to keep the menu graphical. A further implementation may allow for many-route stops by having a checklist for routes as opposed to a pure graphical menu.

Orienting Pieces

The add bus code was augmented to allow experimentation with a piece containing two piece tags, one in each end. The PitA-Board, through examining the positions of each of the two tags, can identify not only where this piece is, but also which *direction* it is facing. If the PitA-Board detects one of the linked pieces, but cannot find the partner piece in an adjacent square, it displays a blinking red and blue circle to indicate the problem. Once the matching piece is located, the board takes appropriate action based on the piece’s location. A piece with this property was utilized to add a bus going a specific direction on a route.

Fade Feature

The fade feature was implemented such that when the tool piece is clicked on a bus stop, the route redraws on top of everything else, and all of the other routes (and their included buses and stops) fade into the background. All routes are still visible; however the route being edited is much more apparent. This feature allows the PitA-Board to make it readily apparent what the editor is affecting, and is quite useful in simplifying complicated scenarios.

Changing Houses

In order to make the house population feature (selection of how many people occupy a house by the tool piece) less abstract, the previous system was replaced by a change in graphics. Upon editing a house, the menu system now displays a choice of four types of residences, ranging from a small home all the way up to a towering apartment complex. When one of these is selected the actual house graphic changes to represent the new type of residence. This system is much more intuitive than the previous number based system and also displays the PitA-Board’s (or technically squeak’s) ability to alter any object’s graphic at a whim.

Logo Design

A simple logo was created for the PitA-Board, with three stylized people circled around a square, representing the PitA-Board. This logo is shown at right. Although the PitA-Board is the center of attention and the obvious focal point, it is clear that the outstretched and linked arms of the people that the interaction and the action around the table is of importance as well. This design is not necessarily complete and final, input and revisions are welcomed as a continuing group design project in the ROLE grant spirit.



Preliminary Investigation of PDA usage

A short investigation was made into using PDA's with the PitA-Board. A Compaq I-Paq was used to test Squeak's usability on the PDA. It was not very useful, however this is not a large problem, since most coding is likely to take place on a computer and then transferred and only run on the PDA. A new I-Paq is on order, which has integrated Bluetooth technology. This will be the PDA used to primarily investigate this application. The PDA will initially replace the tool piece, but will take on greater and more complex roles as time goes on. In the bus simulation, for instance, it could be used as a general query piece investigating everything from neighborhood populations to traffic at intersections.

Creation of PitaController

The meta-design aspects of the PitA-Board began to be taken into account with the creation of the PitaController. The PitaController is a new class designed to abstract out the messy tasks of dealing with the serial port and low-level board interaction from the simulation. A simulation creates a PitaController instance and registers itself with the controller as a listener. The simulation then knows nothing of the actual board interaction, it is just informed by the controller when pieces are added or removed. This greatly simplifies the work on the part of the simulation and will make it many times easier to integrate new simulations into the PitA-Board system.

For the PitaController's part, it automates many tasks. When it is created it automatically searches COM1 for any PitA-Boards present. It then uses a series of touches, ala SmartBoard, to detect the relative position of the boards. Currently this can only detect positions; but soon boards in any (even flipped) orientations will be possible. The controller then polls the boards and sends any piece movement information to the simulation. When it does this it converts the piece's position from a local index on the individual board to a global index over the entire system. This further reduces the simulation's workload and complexity and increases the generality of the system, allowing the PitaController to deal with extra boards not just those that are hard coded in.

Creation of Test System

A convenient new addition to the PitA system is the testing system. If a new PitaController can find no boards attached to the system it queries the user for options, one of which is to use the testing system. If this is selected a graphical representation of the PitA-Board is created, with small icons representing each of the 15 available pieces. If the simulation chooses it can lock this display to its own graphical output, allowing a transparent overlay to the already displayed graphics. The test system then allows for the piece icons to be moved and dropped onto the board wherever desired, and sends the appropriate piece movement information to the simulation. The simulation has no idea that it is being triggered by a test system as opposed to the actual PitA-Board. This allows easy testing of simulations without needing a PitA-Board on hand at all times. Since the system automatically triggers after a simple query box, it is incredibly simple to use with any simulation as well.

FUTURE WORK PLAN

The coming months will mark great strides in the PitA-Board system, with the PitA-Board team moving in many directions and each member focusing on a specific issue. For my part, I will further the abstraction and generalization of the PitaController system, allowing simulation writers to do their jobs easier and the PitA-Board system to become more versatile in its board configurations. Other simulations that have been written will be converted and merged with the new system, including the Walking Distance Simulation. Finally, at long last, PDA use will start to be more heavily investigated with using a Compaq iPaq running Squeak and communicating wirelessly with the PitA-Board system. Following is a more in depth discussion of these items.

Increase Abstraction and Generalizations

The creation of the PitaController was an important first step in the meta-design of the PitA-Board system. Even this system, however, has certain “hacks” that presume such things as board orientations. Creation of a system that determines board positions and orientations based on a series of touches (such as the SmartBoard calibration) is important so that any PitA-Board system can be created in any shape by connecting many individual or smaller board systems in series. Further abstractions are also necessary in order to ease the workload on the simulation designer. Certain interactions such as the stamp, the tool, the path and the directional piece could be another level of abstraction. Currently the simulation itself must deal with these pieces, however this is repetitive and likely unnecessary. Abstractions such as this will be investigated to ease the simulation designer’s work and encourage many simulations to be designed.

In the long run the ideal would be to create a system so easy to deal with that people little to no programming experience can create PitA-Board interaction systems. The PitA-Board could be created by using a series of single board pieces and linking them together without needed to worry about what way is up and which side is left. The PitA-Controller would do all the conversions and calibrate such that everything works transparently.

Integrate Walking Distance Simulation

Hal created a simulation based on the Bus simulation entitled the Walking Distance simulation. This must be converted to the new PitaController based system and integrated with the new generalized PitA-Board system. This seemingly simple task is complicated by the fact that Hal and I worked in parallel on many similar tasks, going in slightly different directions for a variety of reasons. Squeak incorporates a number of tools for simplifying this, however, and it should be completed with reasonable effort.

Integrate PDA Technology

A task of great importance and my main focus after reaching a suitable level of abstraction with the PitaController will be PDA use. PDAs could be incredibly useful as personalized reflection spaces. The current group reflection space may display something such as “The increased bus volume will increase night time noise levels in neighborhoods”, not necessarily pertinent to everybody, such as a farmer living 25 miles from the bus route. A PDA might be able to tell *me personally* “The bus route goes right outside your house. Night time driving noise may be a problem”. This type of personalized information can be incredibly useful. In the EDC a survey about bus usage was completed on the single reflection space, one person at a time. The applications of PDA technology to this task is quite obvious, everybody could complete the survey simultaneously and in private, thanks to their PDA. The PDA could become a personalized tool replacing both the “tool” and the “query” pieces.

The most basic initial steps into investigating this technology have been taken; all of the major work is yet to be completed however. A Compaq iPaq with integrated Bluetooth technology is on order to be used for this purpose. Even discounting the gains in using the PDAs with the PitA-Board, just the knowledge gained in using the Bluetooth system will be very useful in any number of future endeavors. This is a very interesting and important task.