Psybergram

Psybergram.com is a tool used for developing client-server, peer-to-peer, and standalone apps. The apps are written in a new programming language called Parthonyte, as well as a text markup language called Parthotags. Parthonyte is implemented in Java, so apps run on Windows, Mac, and Linux. Server-side code is written in Java. Mobile apps are supported for Android and iOS.

Mandate

The mandate of Psybergram is to use its revenue stream to support a nonprofit organization called Psyberhood.org. This organization creates and maintains a tool used for teaching math and an online community of consumer/survivors. Subscription fees of community members are waived, as well as fees of math tool users (who are clients of other nonprofit organizations).

Business Model

Client-server apps, which are hosted by third party Java hosting services, and peer-to-peer apps (no hosting required), are monetized by charging subscription fees of $60 per regular user per year. Subscription fees for basic users are $20 per year. Guest users of those apps pay no fees and are restricted to monochrome mode (black and white display). Quarterly, a portion of the subscription fee for each regular user equal to $12 (or $4 for basic users) is split up amongst the app authors. Apps which are more heavily used receive a larger share of that portion of the subscription fees.

Fee Allocation

- Apps are ranked by amount of accumulated user-minutes. Any minute in which a given user generates at least 5 user events (each event is basically a mouse click or keystroke) counts as a user-minute.
- Multiple tiers exist. The number of shares per app in each tier is equal to Fib(i), where i = tier no. starting at 1 for the lowest tier and Fib(i) = i-th term in the Fibonacci series: 1, 2, 3, 5, 8, 13,...
- Assume Fib(0) = 1, then Fib(i) = Fib(i-1) + Fib(i-2)
- Iterating from the highest ranking to the lowest ranking, start with a share of 1, after the 2nd iteration, the top ranking app becomes 2 (and the 2 lower ones are each 1). Keep iterating, so that the number of 1's is up to 2 times the number of 2's, which is up to 2 times the number of 3's, which is up to 2 times the number of 5's, which is up to 2 times the number of 8's, and so on.
- Quarterly, perform the above process, to determine the fee allocation amount for each app.
- Note: bots are strictly forbidden, so all user-minutes are genuine.

Sample Table

Example: in the case where 60 apps exist, then 5 tiers exist.

T = Tier No.
F = Fib = Weight
N = App Count
P = Product = F x N

[T, F, N, P]
[1, 1, 30, 30]
[2, 2, 16, 32]
[3, 3, 8, 24]
[4, 5, 4, 20]
[5, 8, 2, 16]

Totals of N and P:
[60, 122]

Monospaced Mode

Only monospaced mode is supported. All characters in a given panel are the same size, and adjacent cells in a given panel may be merged to form a subpanel. Panels and subpanels can contain a widget, graphic, or a block of text. Different panels containing text need not share the same font size. Subscripts and superscripts are offset vertically by half the height of a character cell. Lines of text can optionally be separated by a gap equal to half of a character cell. The canvas property of a panel supports drawing graphics such as lines, circles, and rectangles. Panels with canvases do not contain a character grid.

Basic Users

Basic users save money but can only access 3 apps in full color mode per quarter. All other apps run in monochrome mode. In any given month, 2 of the apps in full color mode experience block-photo mode, in which photos are partially obscured by multiple randomly sized and positioned black rectangles.

About Us

I am Mike Hahn, the founder of Psybergram.com. I was previously employed at Brooklyn Computer Systems as a Delphi Programmer and a Technical Writer (I worked there between 1996 and 2013). At the end of 2014 I quit my job as a volunteer tutor at Fred Victor on Tuesday afternoons, where for 5 years I taught math, computers, and literacy, and became a volunteer math/computer tutor at West Neighbourhood House. I quit that job in mid-2019. I have a part-time job working for a perfume store. My hobbies are reading and I often go for walks. I don't read books very often, but on March 19, 2021 I started reading a biography of Steve Jobs which my brother gave me. I read the CBC news website, news/tech articles on my Flipboard app, and science/tech articles (under Google) on my phone. I visit my brother about once a month.

Steps

1. Develop foundation of Parthonyte code execution - ***done!***
2. Develop rest of Parthonyte code execution: WCNMIL
    1. Wrap up core foundation features
    2. Classes and objects
    3. Non-scalar data types
    4. Modules
    5. Inheritance + Interfaces (hedrons)
    6. Library
3. Release Parthonyte as console-based compiler on GitHub
4. Begin recruiting contributors
5. Write Parthotags design specs
6. Develop Parthotags
7. Integrate Parthonyte with Parthotags
8. PYRE: Parthonyte Runtime Environment (open source)
9. Develop Parthonyte code editor
10. Expand code editor to Parthonyte SDK
11. Create forums for developers and end-users
12. Develop desktop app store
13. Implement peer-to-peer synchronization
14. Monetize client-server and peer-to-peer apps
15. Implement Java Converter
16. Support paid apps
17. Launch website
18. Purchase Google AdWords advertising
19. Develop Math Tutor App (closed source)
20. Implement support for mobile devices:
    1. Hire Android/Swift programmer if sufficient funds
    2. Port PYRE to Android
    3. Develop mobile app store
    4. Convert PYRE for Android to Swift
    5. Develop PYRE for iOS from raw Swift codebase
    6. Develop optional code editor
21. Lower priority features:
    1. Implement Keyboard Aid (bells and whistles of editor)
    2. Develop WYSIWYG Parthotags screen editor
22. Partner with West Neighbourhood House or Parkdale Project Read
23. Use partner(s) to enhance functionality of Math Tutor App
24. Partner with Progress Place
25. Develop online community of consumer/survivors

Revenue

Assume 10,000 regular users and 20,000 basic users exist. Assume an additional 3000 converted users of the Math Tutor App exist, and one third of them are regular users. Revenue generated by users of the Math Tutor App is not shared with other app authors (whenever at least 20 percent of their user-minutes are spent on this app). Then annual revenue equals 10,000(12) + 20,000(4) + 1000(60) + 2000(20) = 120,000 + 80,000 + 60,000 + 40,000 = $300,000.

Monochrome Mode

Foreground and background colors are limited to black and white, respectively. Panels and rectangular widgets have black borders, one pixel in width. The pixel positions of bottom and right borders are incremented by one, so they overlap with the borders of adjacent panels/widgets. Photos are partially obscured by multiple randomly sized and positioned black rectangles.

Java Converter

Converts Parthonyte code to server-side Java code. Parthonyte variable declarations are handled by using variable names beginning with an uppercase letter preceded by a type character: b for boolean, i for int, j for long, c for char, f for double, and s for string. Object variables followed by comment beginning with the class name.

Edit Code on a Phone

Using nested menus, an onscreen "keyboard" has 3 to 5 rows of buttons/text. The top row includes icons for commonly used global functions, and the bottom row includes Space, Enter, Open/Close Parenthesis, and optionally Up/Down Arrow. The middle 1 to 3 rows display menu options: words or special characters, from left to right, then top to bottom. The Up/Down Arrow is used when the available menu options don't fit on one screen. A regular alphanumeric keyboard is displayed when entering identifier declarations. Scrolling vertically or horizontally is accomplished by swiping, and dynamic word wrap (default is enabled) ensures that the rightmost character of a line is always visible.

Default indentation: 2 spaces per level. Every program statement ends with a semicolon. Except for the first line, all lines of a multiline program statement are indented 1 space. Typing close parenthesis automatically inserts semicolon when needed. All text (except comments) is displayed as either upper or lower case, depending on the case-display switch, but that text is stored as lower case in the actual source file. By convention, identifiers of constant values terminated with underscore.

Glossary

Psybergram	Project name
Psyberhood	Nonprofit arm of Psybergram
Parthonyte	New programming language, similar to Python
Parthotags	Text markup language, similar to HTML
PYRE	Parthonyte Runtime Environment

.PTHY	Parthonyte source file
.PTHX	Parthonyte compiled unit
.POTG	Parthotags file
.POTJ	Parthotags/JSON file

Parthonyte

Parthonyte (implemented in Java) is an open source Python dialect in which all operators precede their operands, and parentheses are used for all grouping (except string literals, which are delimited with double quotes, also statements are separated by semicolons). Parthonyte source files have a .PTHY extension. Parthotags is a text markup language, with a .POTG extension. Parthonyte boasts an ultra-simple Lisp-like syntax unlike all other languages.

Special Characters

Core:

- () grouping
- - word separator
- ; end of stmt.
- : dot operator
- " string delimiter
- \ escape char.

Operators:

- + - * / %
- = < >
- & | ^ ~ ! ?

Other:

- # comment
- {} block comment
- _ used in identifiers
- $ string prefix char.

Differences from Python

- Parentheses, not whitespace
- Operators come before their operands
- Integration with Parthotags
- Information hiding: public/private
- Single, not multiple inheritance
- Adds interfaces: "hedron" defs.
- Drops iterators and generators
- Adds lambdas
- Adds quote and list-compile functions, treating code as data
- Adds cons, car and cdr functionality

Parthotags

Parthotags is a simplified markup language used to replace HTML. Mock JSON files using Parthotags syntax have a .POTJ extension, and include no commas. Instead of myid: val, use [myid: val]. Instead of [1, 2, 3], use [arr: [: 1][: 2][: 3]]. Arbitrary Parthotags code can be embedded in the Parthonyte echo statement. Parthotags syntax, where asterisk (*) means occurs zero or more times, is defined as follows:

Tags:

- [tag]
- [tag (fld val)*: body]
- [tag (fld val)*| body |tag]

Body:

- text
- [(fld val)*: text]*

Parthonyte call:

- [expr: <expr>]
- [exec: <stmt>...]
- [pthy: <path>

Note: for fld = style, corresponding val = (fld val)*

Benefits of Parthonyte

Parthonyte is simpler than any other object-oriented programming language, and integrates nicely with Parthotags, which is simpler than HTML. Operators come before their operands, also statements are semicolon-separated, so they never start with a parenthesis. Parthonyte is written in Java, so apps can be written in one or both of the Java and Parthonyte programming languages. Parthotags is simpler than HTML, based on nested rows and columns. A row cell is divided into multiple variable-width column cells, and a column cell is divided into multiple variable-height row cells.

Keyboard Aid

The close delim switch of the Parthonyte code editor enables the automatic insertion of a closing parenthesis, brace, or double quote whenever the open delimiter is inserted. The optional keyboard-aid feature enables hyphens, open parentheses, and close parentheses to be entered by typing semicolons, commas, and periods, respectively. When enabled, keyboard-aid can be temporarily suppressed by using the Ctrl key in conjunction with typing semicolons, commas, and periods (no character substitution takes place).

By convention, hyphens are used to separate words in multi-word identifiers, but semicolons are easier to type than hyphens. Similarly, commas and periods are easier to type than parentheses. Typing semicolon converts previous hyphen to a semicolon, and previous semicolon to a hyphen (use the Ctrl key to override this behaviour). Typing semicolon after close parenthesis simply inserts semicolon. Typing space after hyphen at end of identifier converts hyphen to underscore.

More...

Please click on More to access miscellaneous Parthonyte documentation. Only the first 2 paragraphs of the More web page contain up-to-date info, the rest of that web page is obsolete.

Psyberhood

Psyberhood.org is a nonprofit organization. Its sister website, Psybergram.com, makes money from charging subscription fees to users of apps written in a new programming language called Parthonyte. The apps are client-server, peer-to-peer, or standalone. Standalone apps can be accessed without a subscription.

Primary Mandate

The primary mandate of Psyberhood is to provide a tool, the House Peer-to-Peer app, which is used for teaching math. Organizations which employ volunteer tutors who provide free support to learners are encouraged to use the tool, and those users receive free subscriptions. If necessary, funding is sourced from grants and donations to cover any shortfall in operating revenue and expenses.

Secondary Mandate

In case Psybergram is profitable, the secondary mandate of Psyberhood is to develop and nurture online communities of consumer/survivors and other categories of people with disabilities. Organizations which work with consumer/survivors such as Progress Place will be encouraged to make use of these communities, and their users receive free subscriptions to Psybergram.

Math Tutor App

The House Peer-to-Peer app is a tool used for teaching math. The top half of the screen is used by the tutor, and the bottom half of the screen is used by the learner. Both users can highlight/copy text on the other user's halfscreen and paste it on their own screen. Both halfscreens can scroll vertically, so they can contain dozens of lines if needed. Chat window: always on top, yellow background, variable width font. Audio is provided by cellphones on speaker phone.

Math Commands

- Use the arrow keys to move the cursor.
- Type underscore(s) to underline the numerator of a fraction.
- Use the special character command (Ctrl+K) to insert special characters such as pi, square root, sum, and integral.
- Use Ctrl+Tab/Shift+Tab to display/undo the next step in the math problem being solved.
- Type question mark (?) to explain the current step or to break the current step down into lower-level steps.
- Click on Help after typing question mark to access the help system.

Miscellaneous Commands

- Use asterisk and slash for multiply and divide.
- Fractions or matrices enclosed in brackets use tall brackets.
- Mixed numbers (example: three and a half) employ a vertical offset of half a character cell for the whole number.
- Smart down/up arrow: press it after inserting a character moves the cursor beneath/above that character.
- Functions such as lines and parabolas can be plotted interactively on a graph.

- The default-to-upper-case setting assumes that all letters entered are upper case (use the shift key to enter a lower case letter), so Caps Lock is redundant.

Expression Language

Mathematical expressions are encoded (internally) using a Parthonyte-style expression. Each step in the math problem being solved manipulates this expression. Even if the user enters steps in a different order than the default ordering, the simplification logic can handle that. The user can type Ctrl+Tab/Shift+Tab to redo/undo her previous step, as well as to redo/undo the computer's previous step.

Superscripts

Superscripts and subscripts are handled by employing a vertical offset of half a line per level of superscripting or subscripting. The caret symbol (^) is used as a superscript prefix, double-caret (^^) is used as a subscript prefix, and backslash (\) is used as an escape character (terminate super/subscript with a semicolon). Carets and double-carets cannot be mixed (exception: one level of superscript can be combined with one level of subscript).

Advanced Commands

These next 2 paragraphs may be ignored (they are written in computerese). Use Shift+Arrow Key to highlight a rectangular block. Press Insert to insert a row or column of spaces before a highlighted block (insert blank line if no highlight). Press Shift+Insert/Delete to insert/delete an entire row/column when a block is highlighted. The next paragraph discusses commands handling multiple indent levels.
Press Enter at end of a line of text: insert blank line, back up on that line to line up with beginning of text on previous line. Press Enter on blank line to back up to line up with beginning of text on a previous line, or insert blank line if already at beginning of line. Press Tab to move forward to line up with beginning of first or next word on a previous line. Press Home to move to beginning of text on current line, press it again to toggle between beginning of line and beginning of text. The user doesn't have to memorize these commands: type question mark at any time to access the help system.

Contact Info

Mike Hahn
Founder
Psybergram.com
2495 Dundas St. West
Ste. 515
Toronto, ON M6P 1X4

Phone: 416-533-4417
Email: hahnbytes (AT) gmail (DOT) com
Web: treenimation.net/hahnbytes/
Country: Canada