

BẢN THUYẾT MINH PHẦN MỀM SÁNG TẠO

Nguyễn Gia Phong

Ngày 7 tháng 8 năm 2018

Mục lục

1 THÔNG TIN CHUNG	2
1.1 Giới thiệu sản phẩm	2
1.2 Tên tác giả, nhóm tác giả	2
2 MÔ TẢ SẢN PHẨM	2
2.1 Yêu cầu cơ sở hạ tầng để triển khai ứng dụng	2
2.2 Sản phẩm được phát triển ước tính trong khoảng thời gian	2
2.3 Phạm vi ứng dụng	2
2.4 Hướng dẫn cài đặt và sử dụng	3
2.5 Mô tả các tính năng cơ bản của sản phẩm	3
2.5.1 Tùy chỉnh	4
2.5.2 Điều khiển qua socket	6
2.5.3 Kết xuất ra JSON	8
2.6 Tự đánh giá tiềm năng ứng dụng của sản phẩm	8
2.7 Tự đánh giá những mặt tồn tại chưa được giải quyết	8
3 KẾT LUẬN	9
3.1 Hướng phát triển của sản phẩm trong tương lai	9
3.2 Nguyên vọng trong tương lai	9

1 THÔNG TIN CHUNG

1.1 Giới thiệu sản phẩm

Tên sản phẩm: Mê cung Khốc liệt (Brutal Maze)

Công cụ sử dụng (ngôn ngữ lập trình): Python

Cấu hình cài đặt: Máy tính điện tử với Windows, macOS, GNU/Linux, *BSD và một số hệ điều hành giống-Unix khác

Đăng kí bằng thi: D3

Thuộc đoàn: Hà Nội

1.2 Tên tác giả, nhóm tác giả

Họ và tên thí sinh: Nguyễn Gia Phong

Ngày tháng năm sinh: 26/05/2000

Học sinh lớp 12A1, trường THPT Phạm Hồng Thái, quận Ba Đình, Hà Nội

Số điện thoại liên lạc: 024 3835 0837

Email: vn.mcsinix@gmail.com

2 MÔ TẢ SẢN PHẨM

2.1 Yêu cầu cơ sở hạ tầng để triển khai ứng dụng

- Phần cứng: Máy tính điện tử với kết nối Internet
- Phần mềm:
 - Hệ điều hành: Microsoft Windows, Apple macOS, GNU/Linux, các *BSD hoặc một số Unix-like khác
 - Thư viện: Python, pip*, pygame[†] và appdirs[‡]

2.2 Sản phẩm được phát triển ước tính trong khoảng thời gian

Số tháng: 11 tháng (từ 10/2017 đến 08/2018)

2.3 Phạm vi ứng dụng

Mê cung Khốc liệt hướng đến hai đối tượng sử dụng chính:

Người chơi Là trò chơi hành động đơn giản với giao diện trực quan, Mê cung Khốc liệt phù hợp cho mọi lứa tuổi.

Người yêu thích lập trình Hỗ trợ TCP/IP socket cho điều từ chương trình bên ngoài, trò chơi có thể được sử dụng trong học lập trình bot.

*<https://pip.pypa.io/>

†<https://www.pygame.org/wiki/about>

‡<https://github.com/ActiveState/appdirs>

2.4 Hướng dẫn cài đặt và sử dụng

Mê cung Khốc liệt có thể chạy trên các phiên bản hiện hành của cả Python 2 và 3. Trước khi cài đặt trò chơi, người dùng cần cài Python, pip và các thư viện pygame theo các bước sau:

1. Cài đặt Python, setuptools và pip:
 - Cho Windows: Chạy file `dist/python-*.exe` được đóng gói kèm sản phẩm.
 - Cho macOS: Chạy file `dist/python-*-macosx*.pkg` được đóng gói kèm sản phẩm.
 - Cho các Unix-like khác: Sử dụng phần mềm quản lý gói của bản phân phối để cài các gói `python`, `python-pip`, `python-setuptools` (tên chính xác các gói phụ thuộc vào bản phân phối) [1].
2. Cài đặt các thư viện (nếu có đường truyền Internet có thể bỏ qua bước này và chuyển sang bước 3):
 - `appdirs`: Trong Command Prompt/PowerShell (với Windows) hoặc Terminal/Console (trong các hệ điều hành họ Unix), chạy lệnh `pip install dist/appdirs-*.whl` (với các tệp `dist/*.whl` được đóng gói trong thư mục sản phẩm).
 - `pygame`: Chạy lệnh `pip install dist/pygame-*-<nền tảng>.whl`, với *nền tảng* là hệ điều hành và CPU.
3. Cài đặt trò chơi: Sử dụng lệnh `pip install dist/brutalmaze-*.whl`. Với máy tính có kết nối Internet, `pip install brutalmaze` sẽ tự động cài đặt cả các thư viện phụ thuộc ở bước 2.

Sau khi cài đặt, trò chơi có thể được mở trong launcher của hệ điều hành với tên `brutalmaze`. Mặc định, chuột phải hoặc các phím WASD phải được sử dụng để di chuyển, chuột trái để bắn và phím cách để chém và đỡ đạn.

2.5 Mô tả các tính năng cơ bản của sản phẩm

Mê cung Khốc liệt là trò chơi hành động tiết tấu nhanh. Người chơi phải điều khiển hình tam giác bất lực chiến đấu với những hình vuông hòng thoát khỏi mê cung. Các tính năng nổi bật của trò chơi:

- Mê cung vô tận được tạo một cách ngẫu nhiên.
- Cửa sổ trò chơi có thể thu/phóng ở hầu hết mọi tỷ lệ*.
- Bot (các hình vuông) phong phú với nhiều khả năng như vô hình, làm bất động, nhân bản, ...
- Engine vật lý tương đối thực với quán tính của chuyển động và mức sát thương dựa trên khoảng cách.
- Dễ dàng tùy chỉnh qua lựa chọn dòng lệnh hoặc tệp INI.
- Hỗ trợ kết xuất bản ghi trò chơi theo định dạng JSON.
- Cho phép điều khiển qua TCP/IP socket.

*Thực tế tỉ lệ nằm ngoài khoảng 1:16 đến 16:1 sẽ gây khó khăn cho việc tạo mê cung.

2.5.1 Tùy chỉnh

Các tùy chọn đơn giản có thể được xác định qua giao diện dòng lệnh:

-h, --help in trợ giúp và thoát

-v, --version in phiên bản và thoát

--write-config [PATH] in tùy chỉnh mặc định vào tệp PATH và thoát, nếu PATH không được chỉ định, in ra stdout

-c, --config PATH nạp tùy chỉnh từ tệp PATH

-s, --size chỉ định kích thước XxY cho trò chơi

-f, --max-fps FPS chỉ định số khung hình tối đa trong một giây

--[un]mute tắt âm thanh

--music-volume VOL đặt âm lượng nhạc nền (trong khoảng 0.0 đến 1.0)

--space-music dùng nhạc nền ngoài không gian vũ trụ

--default-music dùng nhạc nền mặc định

--record-dir DIR thư mục để chứa bản ghi

--record-rate SPF số kết xuất mỗi giây

--[no-]server bật/tắt server socket

--host đặt địa chỉ server

--port đặt cổng server

-t, --timeout TIMEOUT đặt thời gian chờ trước khi server ngắt kết nối khỏi client khi không nhận được hồi đáp (tính theo giây)

--head[less] bật/tắt cửa sổ trò chơi khi chạy trong chế độ server

Tệp tùy chỉnh hỗ trợ thêm các lựa chọn về điều khiển. Tệp mặc định có dạng như sau:

[Graphics]

Độ rộng và độ dài khung hình (tính theo pixel)

Screen width: 640

Screen height: 480

FPS tối đa không nên vượt quá tốc độ nháy của màn hình

để tránh lãng phí tài nguyên

Maximum FPS: 60

```

[Sound]
# Tắt âm thanh (yes cho có, no cho không)
Muted: no
# Âm lượng nhạc nền, trong khoảng 0.0 đến 1.1
Music volume: 1.0
# Nhạc nền không gian vũ trụ lạnh lẽo, rộn rịp
Space theme: no

# Các phím điều khiển hợp lệ được liệt kê trong
# http://www.pygame.org/docs/ref/key.html
# (không tính phần K_). Tổ hợp phím không được hỗ trợ.
# Chuột trái, giữa, phải lần lượt là Mouse1 đến Mouse3.
[Control]
# Trò chơi mới
New game: F2
# Tạm dừng trò chơi
Toggle pause: p
# Tắt/bật tiếng
Toggle mute: m
# Di chuyển sang trái, phải, trên, dưới
Move left: Left
Move right: Right
Move up: Up
Move down: Down
# Di chuyển tự động
Auto move: Mouse3
# Bắn và chém (đồng thời đỡ đạn)
Long-range attack: Mouse1
Close-range attack: Space

[Record]
# Thư mục sẽ chứa bản ghi, để trống để tắt tính năng này.
Directory:
# Số kết xuất mỗi giây, nên trong khoảng từ 3 đến 60.
Frequency: 30

[Server]
# Bật server sẽ vô hiệu hoá điều khiển trực tiếp
Enable: no
# Địa chỉ server socket
Host: localhost
Port: 8089
# Thời gian chờ trước khi server ngắt kết nối khỏi client
# khi không nhận được hồi đáp (tính theo giây)
Timeout: 1.0
# Bật/tắt cửa sổ trò chơi khi chạy trong chế độ server
Headless: no

```

Các tùy chỉnh được nạp theo thứ tự sau, theo đó các giá trị sau sẽ được nạp đè lên giá trị trước:

0. Tùy chọn mặc định
1. Tập tùy chỉnh hệ thống và của người dùng*
2. Tập INI được chỉ định bởi thẻ `--config`
3. Các tùy chọn trong dòng lệnh

2.5.2 Điều khiển qua socket

Khi được khởi động với thẻ `--server` hoặc chỉ định trong tập tùy chỉnh (**Enable:** `yes` trong mục **[Server]**), Mê cung Khốc liệt sẽ mở một TCP/IPv4 socket tại địa chỉ đã chọn. Trước khi gửi dữ liệu trò chơi, server sẽ gửi độ dài l của dữ liệu dài 7 byte (ví dụ 0000531). Dữ liệu có dạng chung như sau:

<Số dòng mê cung (nh)> <Số kẻ thù (ne)> <Số đạn (nb)> <Điểm>
<nh dòng mô tả phần mê cung được hiển thị>
<Một dòng mô tả tam giác>
<ne dòng mô tả các kẻ thù (hình vuông)>
<nb dòng mô tả các viên đạn đang bay>

Mê cung Biểu diễn mê cung là bản đồ kí tự (các kí tự khác 0 đều biểu thị tường) dài nh dòng và rộng nw cột. Mỗi ô tương ứng với 100 đơn vị độ dài.

Tam giác Biểu diễn bởi 6 giá trị:

Màu sắc Một kí tự biểu thị thể lực nhân vật
Hoành độ Số nguyên trong đoạn $[0, nw \times 100]$
Tung độ Số nguyên trong đoạn $[0, nh \times 100]$
Góc Hướng ngắm bắn của nhân vật (theo độ)
Có thể tấn công 1 nếu có và 0 nếu không
Có thể hồi sức 1 nếu có và 0 nếu không

Hình vuông biểu diễn bởi 4 giá trị:

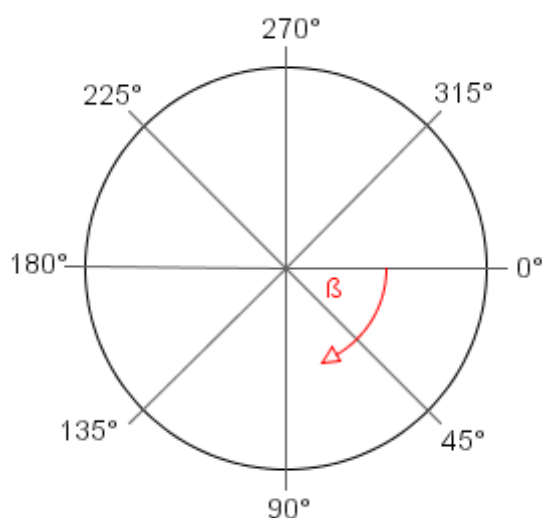
Màu sắc Một kí tự biểu thị thể lực nhân vật
Hoành độ Số nguyên trong đoạn $[0, nw \times 100]$
Tung độ Số nguyên trong đoạn $[0, nh \times 100]$
Góc Hướng ngắm bắn của nhân vật (theo độ)

Đạn biểu diễn bởi 4 giá trị:

Màu sắc Một kí tự biểu thị sức sát thương (trong $[0, 1]$)
Hoành độ Số nguyên trong đoạn $[0, nw \times 100]$
Tung độ Số nguyên trong đoạn $[0, nh \times 100]$
Góc Hướng bay (theo độ)

*Lần lượt được liệt kê trong `brutalmaze -h` ở thẻ `--config`.

Cần chú ý rằng trong trò chơi, trục hoành hướng xuống dưới kéo theo sự đảo ngược chiều dương của góc lượng giác:



Thẻ lực các nhân vật theo màu được biểu diễn bằng các kí tự như sau:

Thẻ lực	5	4	3	2	1	0
Vàng			a	b	c	0
Cam			d	e	f	0
Nâu			g	h	i	0
Lục			j	k	l	0
Lam			m	n	o	0
Tím			p	q	r	0
Đỏ			s	t	u	0
Xám	v	w	x	y	z	0

Sức sát thương của đạn cũng được kí hiệu tương tự (ứng với 1, $\frac{2}{3}$, $\frac{1}{3}$ và 0), riêng đạn xám của tam giác dùng 4 kí tự **vw**x0.

Tương ứng, server nhận tối đa 7 byte theo dạng <Hướng di chuyển> <Góc ngắm> <Cách tấn công>:

Hướng di chuyển được mã hoá theo bảng sau:

Hướng	Trái		Phải
Trên	0	1	2
	3	4	5
Dưới	6	7	8

Góc tính theo độ, làm tròn về số nguyên trong $[0, 360)$.

Cách tấn công 0 chờ, 1 bắn, 2 chém.

Như vậy, client sẽ phải thực hiện các bước sau:

1. Tạo TCP/IP socket và kết nối với server
2. Nhận độ dài l của dữ liệu trò chơi, nếu $l < 0$ thì đóng socket và thoát
3. Nhận dữ liệu dài l byte
4. Xử lý dữ liệu
5. Gửi hướng dẫn điều khiển tam giác đến server và quay lại bước 2

Client sẽ được đánh giá dựa trên số điểm đạt được và thời gian đạt số điểm đó (càng ngắn càng tốt). Để thuận tiện, server sẽ in các thông số này ra stdout.

2.5.3 Kết xuất ra JSON

Mê cung Khốc liệt cho phép kết xuất quá trình chơi (xem các tùy chọn ở 2.5.1) để thuận tiện chia sẻ chiến thuật, đặc biệt khi lập trình bot. Bản ghi JSON có thể được dễ dàng phát lại tại Github Page của trò chơi*.

2.6 Tự đánh giá tiềm năng ứng dụng của sản phẩm

Trò chơi có thể ứng dụng trong giải trí đơn thuần cũng như học hỏi, luyện tập lập trình bot trong thời gian thực. Có thể dễ nhận thấy rằng, chiến thuật tốt đòi hỏi phải xử lý các vật thể trong dòng thời gian để phán đoán cũng như lập ra bước đi tiếp theo. *Học máy* (machine learning) có thể ứng dụng trong bài toán này bởi thực tế không hề có cách giải hoàn hảo.

Mặt khác, là phần mềm tự do[†] với dung lượng mã nguồn tương đối nhỏ (khoảng 1500 dòng kể cả chú thích và dòng trống) và tài liệu tương đối đầy đủ, Mê cung Khốc liệt còn có thể dùng như ví dụ điển hình cho lập trình trò chơi đơn giản, xử lý tùy chỉnh và socket trong Python.

2.7 Tự đánh giá những mặt tồn tại chưa được giải quyết

Trò chơi còn tồn tại một số vấn đề như sau:

- Đồ hoạ còn chưa hấp dẫn.
- Thuật toán tạo mê cung chưa hoàn hảo (tồn tại vùng cô lập với bên ngoài).
- Chưa hỗ trợ tùy chỉnh qua giao diện đồ hoạ.
- Tiêu tốn nhiều tài nguyên máy do sử dụng SDL phiên bản 1 và Python.

*<https://McSinyx.github.io/brutalmaze/>

[†]Trò chơi được phát hành theo giấy phép GNU AGPL bản 3 tại <https://github.com/McSinyx/brutalmaze>.

3 KẾT LUẬN

3.1 Hướng phát triển của sản phẩm trong tương lai

Giải quyết các mặt tồn tại về đồ hoạ cũng như thuật toán tạo mê cung; mặt khác có thể viết thêm phiên bản cho điện thoại thông minh và cho web server, chú trọng tạo nền tảng cho cộng đồng tham gia chia sẻ các chiến thuật chơi.

Để cải thiện hiệu năng, giải pháp tốt nhất có lẽ là dùng phiên bản Pygame dùng SDL2, tuy nhiên tại thời điểm viết bản thuyết minh này, Pygame_sdl2 chưa hỗ trợ đầy đủ API của Pygame [2].

3.2 Nguyên vọng trong tương lai

Trò chơi được sử dụng rộng rãi và có ích cho cộng đồng trong cả phương diện giải trí cũng như học tập.

TÀI LIỆU THAM KHẢO

[1] <https://packaging.python.org/guides/installing-using-linux-tools/#installing-pip-setuptools-wheel-with-linux-package-managers>

[2] https://github.com/renpy/pygame_sdl2#current-status