



**Oops! My Chatbot  
becomes a Mobile Game**



蒼時  
弦也

@elct9620

WEB DEVELOPER  
GAME DEVELOPER

5 x {  } · tw

Can we build a **Mobile Game** on Chatbot?



## 銅賞



### 新紀元：重生

得獎者 玄武工作室

「新紀元：重生」是一款結合科幻、戰鬥、對白、機器人等元素的對話式生存遊戲，玩家可以在遊戲最初期於不同的選擇中，獲得不同的機器人來駕駛，並且扮演主角「約翰」在遊戲裡努力生存，與邪惡勢力抗爭；而也必須在有限的資源下遊走世界，獲得更多經驗、更多資源、提升等級、提升機器人的戰鬥力，以抵抗更頑強、更厲害的敵人。經過好幾次的世界大戰之後，地球已成為極其荒廢破敗的慘況，自然環境受到嚴重破壞、世界各地也缺乏物資，簡直民不聊生，此時名為「吉利帝國」的極權組織高舉新興政府之名義開始崛起，一舉消滅世界許多大國，大力搜刮收編各種人力物力，坐擁龐大資源，造成各方逐漸出現反抗勢力。自小就身為戰爭孤兒的主角「約翰」，痛恨吉利帝國的行徑，誓言要對抗他們到底。在一次反抗軍與帝國的猛烈混戰中，約翰一邊搜刮物資，一邊潛入敵軍基地設施，卻意外發現大型戰鬥兵器，開啟了他與帝國死命對抗的命運，唯有想盡辦法在這個詭譎的局勢中繼續生存，才有機會擊潰恐怖的惡勢力。

[查看作品](#)

## 佳作賞 共6名

## 評審獎 得獎名單

## 金賞



Crystal小晶

得獎者 影子-shadow★

由Crystal公司的頂尖科技打造給人類的全新伴侶「小晶」，在你的啟動下甦醒了她。為了幫助她尋找缺少的部分讓她成為完美，你開始與「小晶」踏上了未知的旅程.....

[查看作品](#)

## 銀賞



# 美女夢工場

PRINCESS MAKER 5

[www.ttime.com.tw](http://www.ttime.com.tw)

T-TIME GAINAX

サイバーポート

How **fast** the Ruby on Rails can be?

But it also becomes a "FRAME" to limit us

Model  
View  
Controller

Model

~~View~~

~~Controller~~

You are Ruby programmer, not Rails programmer!





Action  
Sender  
Event  
Handler  
Reward  
etc.



新紀元：重生-08:14

先到卡菲城鎮稍作休息，應該位於 C-8



新紀元：重生-08:17

村長: 小鎮東邊受到襲擊，需要支援  
(D-8)



新紀元：重生-08:18

小鎮東南邊也傳出嚴重砲擊 (D-9)



← Handler



新紀元：重生-08:18



15:53

冒險



新紀元：重生-15:53

```
# frozen_string_literal: true

class HelpHandler < BaseHandler
  def perform
    send t('chat.help')
  end
end
```

新紀元：重生-08:14

先到卡菲城鎮稍作休息，應該位於 C-8

新紀元：重生-08:17

村長: 小鎮東邊受到襲擊，需要支援  
(D-8)

新紀元：重生-08:18

小鎮東南邊也傳出嚴重砲擊 (D-9)

08:18

冒險

新紀元：重生-08:18



Sender

15:53

冒險

新紀元：重生-15:53

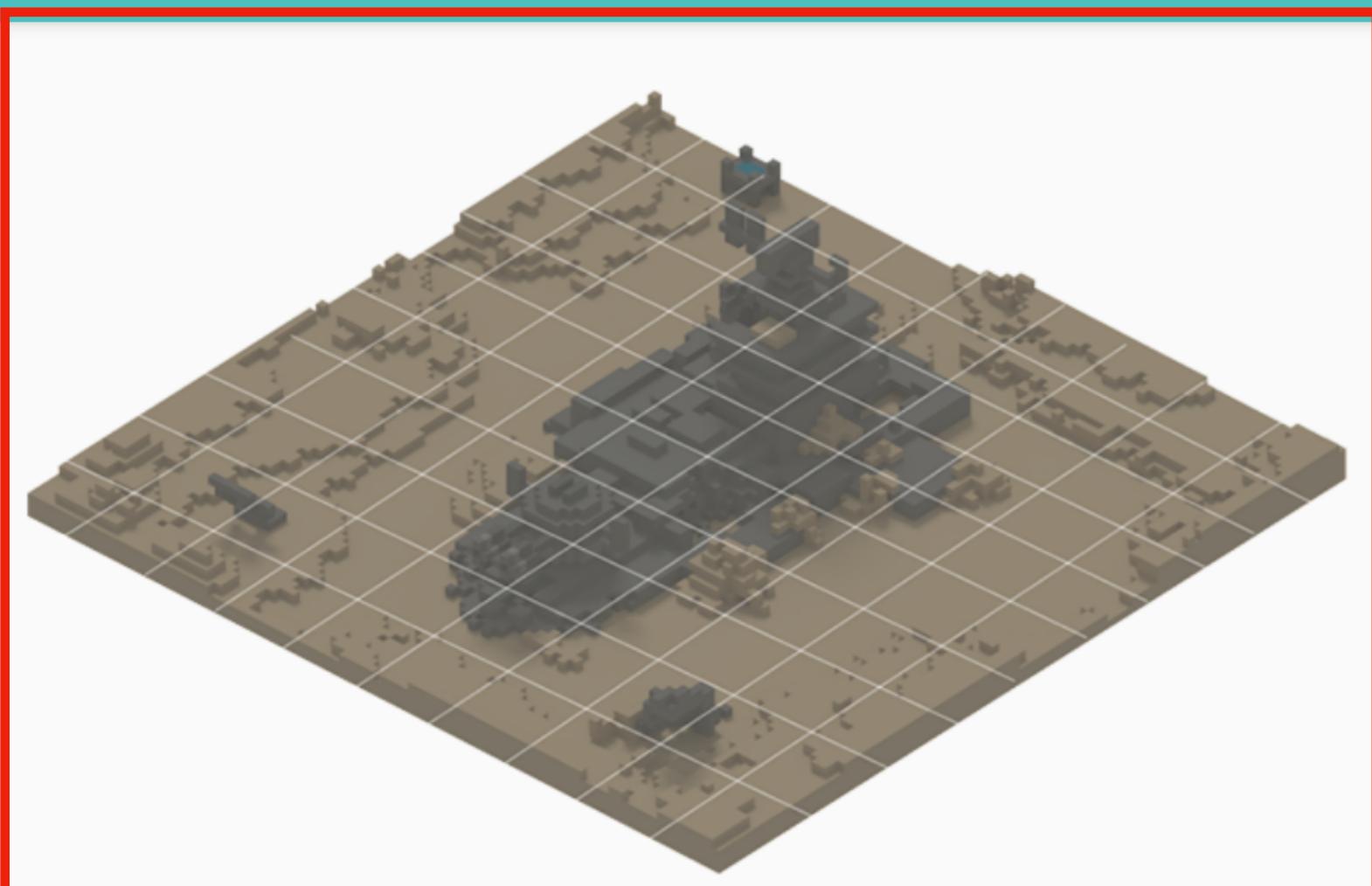
```
# frozen_string_literal: true

module Bahamut
  class TextSender < Bahamut::BaseSender
    def initialize(recipient, message)
      @recipient = recipient
      @message = message
    end

    def body
      build(type: 'text', text: @message)
    end
  end
end
```



新紀元：重生



進入了高安全係數區(F4)

向北

向西

向東

向南

Event

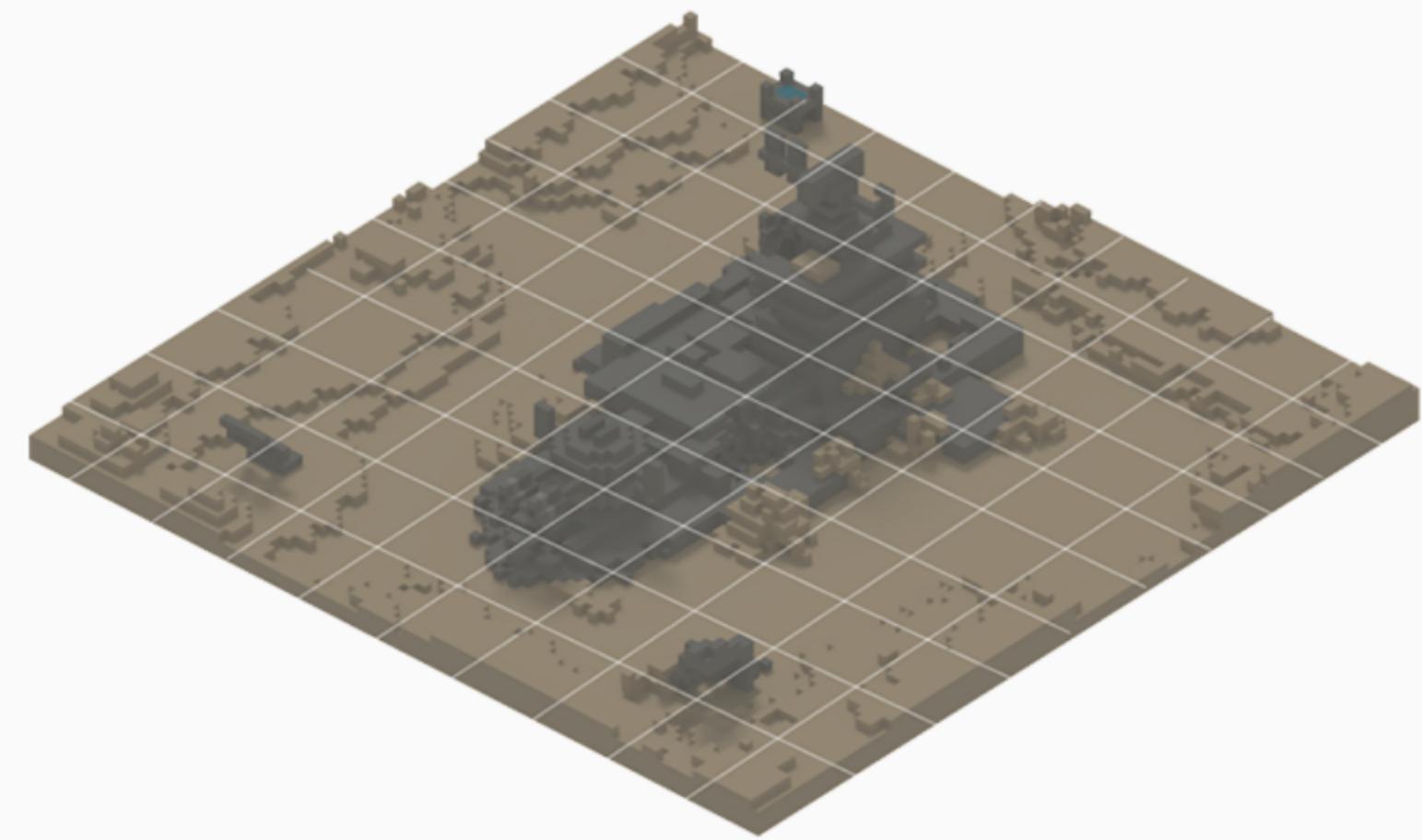
```
# frozen_string_literal: true

class ErrorEvent < BaseEvent
  def initialize(message)
    @message = message
  end

  def to_event
    {
      text: {
        message: @message,
        color: '#7A929B'
      },
      button: {
        hidden: true
      }
    }
  end
end
```



新紀元：重生



進入了高安全係數區(F4)

向北

向西

向東

向南

Action

```
# frozen_string_literal: true

class MapAction < BaseAction
  before_action :find_map
  skip_before_action :find_map, if: -> { params[:action] == :back }

  delegate :current_robot, :current_map, :current_mission,
  to: :current_player

# ...

def back
  bot_event MapEvent.new(current_player.current_map)
end

private

# ...
end
```

Why use the Rails even it **not useful** in this case?

We are Ruby programmer, not only Rails programmer!

The flexible came from well design, not framework

THANKS