

RAILS AS A REAL-TIME MULTIPLAYER GAME ENGINE

PAWEŁ STRZAŁKOWSKI
CTO @ **visuality**

RAILS WORLD 2024

WHAT IS A GAME?







BEST-SELLING GAMES EVER

- Minecraft
- Grand Theft Auto V
- Tetris
- Wii Sports
- PUBG: Battlegrounds
- Mario Kart 8 Deluxe

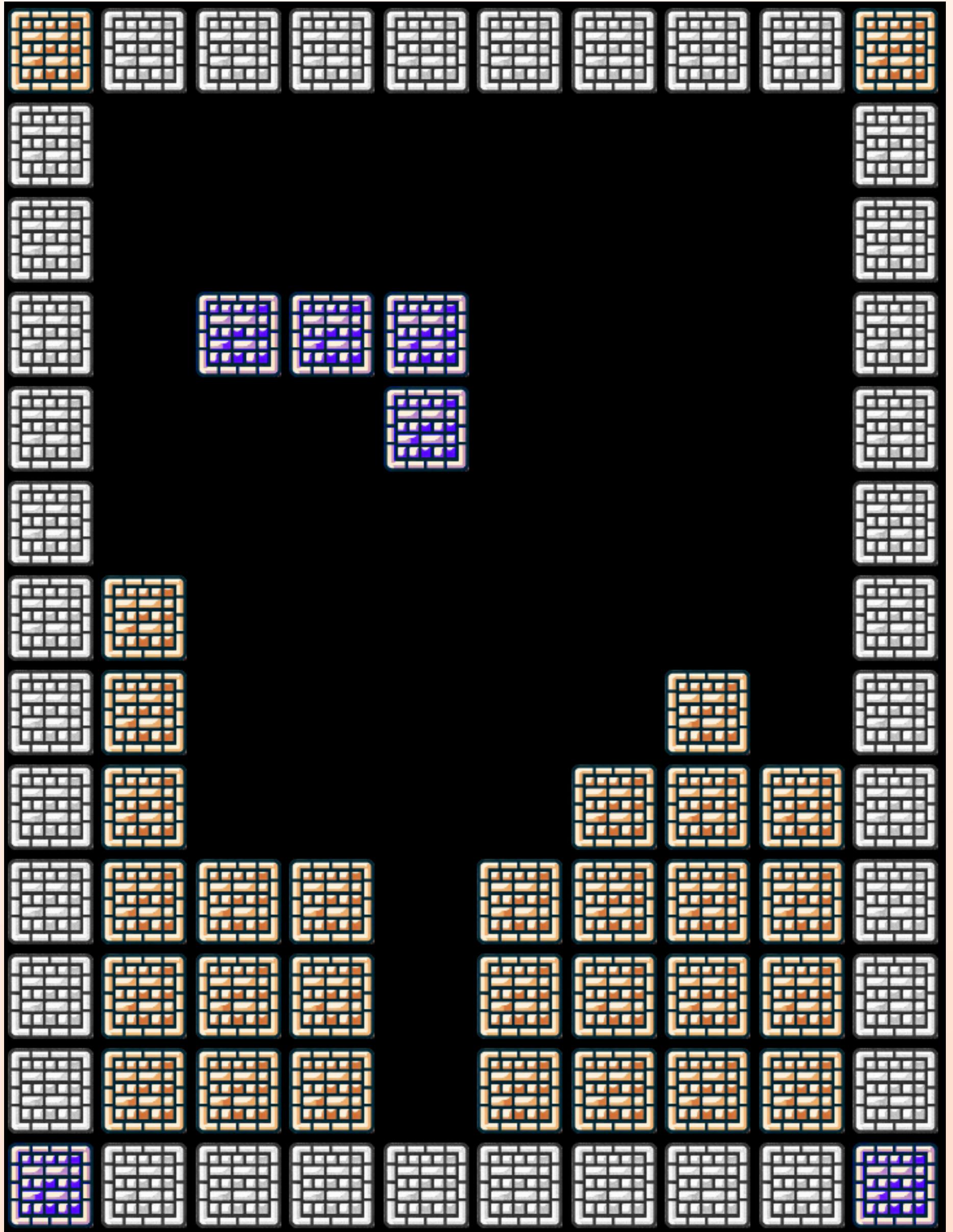
BEST-SELLING GAMES EVER

- Minecraft
- Grand Theft Auto V
- **Tetris**
- Wii Sports
- PUBG: Battlegrounds
- Mario Kart 8 Deluxe



TETRIS

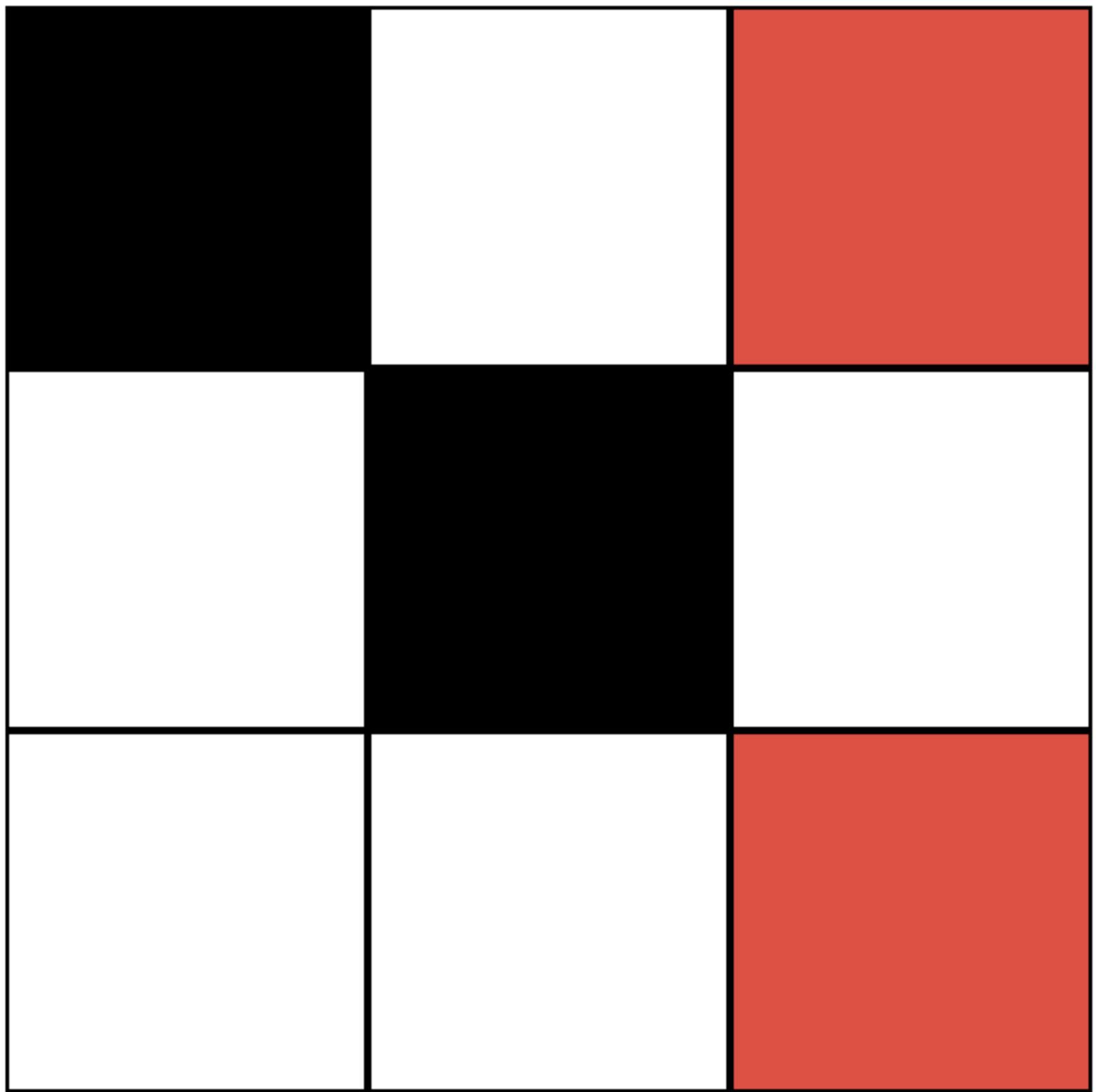
IMPLEMENTED IN RUBY ON RAILS
FOR EURUKO 2023 IN VILNIUS





TIC TAC TOE

IMPLEMENTED IN RUBY ON RAILS AS
A DOMAIN-DRIVEN DESIGN EXERCISE





THERE BE DRAGONS

IMPLEMENTED IN RUBY ON RAILS FOR
RUBY WARSAW COMMUNITY CONFERENCE

HOW TO WRITE A GAME?

ALL YOU NEED IS

\$ RAILS NEW YOUR_GAME

HOW TO MAKE A GAME OBJECT MOVE?

HOW TO MAKE A TRAIN RIDE?

```
class Train < ApplicationRecord
  def move_forward = self.position += 1
end
```

HOW TO MAKE A TRAIN RIDE?

```
class Train < ApplicationRecord
  def move_forward = self.position += 1
end
```

```
train = Train.find(17)
train.move_forward
train.save
```

HOW TO MAKE A TRAIN RIDE?

```
loop do
  train = Train.find(17)
  train.move_forward
  train.save

  sleep 0.1
end
```

HOW TO MAKE A TRAIN RIDE?

```
namespace :train do
  desc 'Goes choo choo'
  task choo_choo: :environment do
    loop do
      train = Train.find(17)
      train.move_forward
      train.save

      sleep 0.1
    end
  end
end
```

GAME LOOP

- A process which runs continuously as long as a game runs
- Processes the current state and calculates the next one

GAME LOOP

```
namespace :game do
  desc 'Play'
  task play: :environment do
    loop do
      game = Game.load
      game.progress
      game.save

      sleep GAME_TICK_INTERVAL
    end
  end
end
```

HOW TO CONTROLL A GAME?

GAME CONTROLS

```
class GamesController < ApplicationController
  def create
    # ...
  end

  def move
    # ...
  end

  def shoot
    # ...
  end
end
```

**HOW TO TRANSMIT
THE GAME STATE?**

HOW TO TRANSMIT THE GAME STATE

- **Polling** - browser asks the server for updates
- **Pushing** - server sends messages to the browser as they become available

POLLING

// WARNING!!! JAVASCRIPT AHEAD

```
async function updateGameState() {  
    const url = 'https://game.server.url/game_state';  
  
    const response = await fetch(url);  
    const gameState = await response.json();  
  
    renderGameState(gameState);  
  
    setTimeout(updateGameState, 100);  
}
```

PUSHING

app/models/game.rb

```
class Game < ApplicationRecord
  broadcasts_to ->(game) { "game_#{game.id}" }
end
```

PUSHING

```
# app/models/game.rb
class Game < ApplicationRecord
  broadcasts_to ->(game) { "game_#{game.id}" }
end

# views/games/_game.html.erb
<div id="<%=" game_#{game.id}" %>">
  <!-- just HTML things -->
</div>
```

PUSHING

```
# app/models/game.rb

class Game < ApplicationRecord
  broadcasts_to ->(game) { "game_#{game.id}" }
end
```

```
# views/games/_game.html.erb

<div id="<%= "game_#{game.id}" %>">
  <!-- just HTML things --&gt;
&lt;/div&gt;</pre>
```

```
# views/games/show.html.erb

<%= turbo_stream_from "game_#{@game.id}" %>

<%= render @game %>
```

FULL GAME EXPERIENCE

USER INPUT

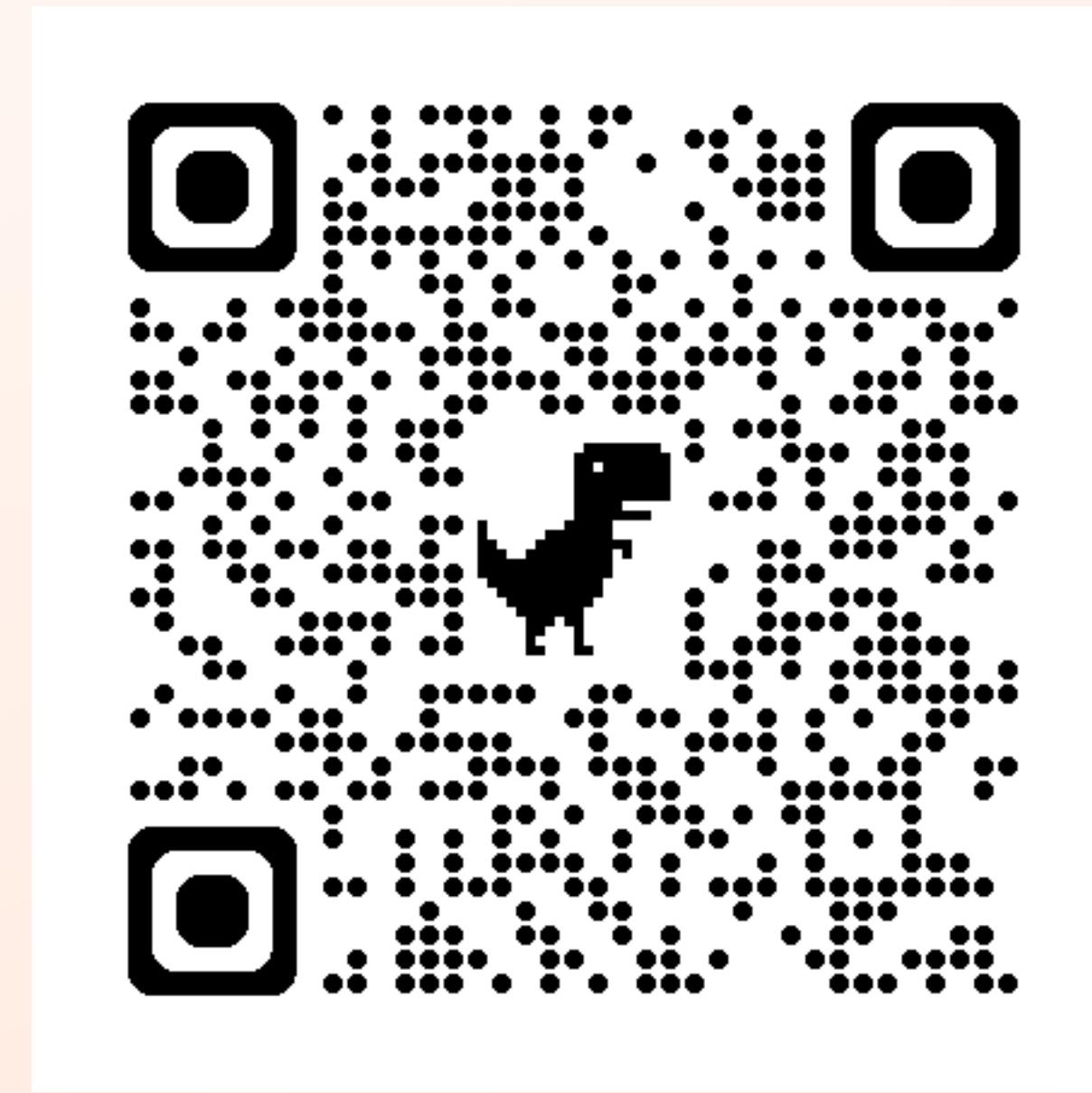
GAME LOOP

BROADCAST

FULL GAME EXPERIENCE



DO YOU WANT TO PLAY A GAME?



RUBY ON RAILS - THE GAME (DEMO)

THANK YOU!
QUESTIONS?

