



Robotron: 2084 inspired Game  
Written in PyGame with MVC architecture

John Montgomery  
Candidate Number: 5199  
Centre Name: Kimberley College Sixth Form  
Centre Number: 15125  
Qualification: AQA 7517 (A-Level Computer Science)

Supervisor: B. Harris

2020-2022

# Contents

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>Analysis</b>	<b>4</b>
2.1	What is MVC? . . . . .	4
2.2	The Game . . . . .	5
2.3	Limitations . . . . .	5
2.4	Objectives . . . . .	5
2.5	Design and Inspiration . . . . .	6
<b>3</b>	<b>Documented Design</b>	<b>8</b>
3.1	MVC in practice . . . . .	8
3.2	Database . . . . .	8
3.3	The API . . . . .	8
3.4	The Server Setup . . . . .	10
3.5	Security . . . . .	10
<b>4</b>	<b>Technical Solution</b>	<b>12</b>
4.1	Example . . . . .	12
<b>5</b>	<b>Testing - TODO</b>	<b>13</b>
<b>6</b>	<b>Evaluation - TODO</b>	<b>14</b>
<b>7</b>	<b>Appendix &amp; Bibliography</b>	<b>15</b>
7.1	Appendix . . . . .	15
7.2	Listings . . . . .	15
7.2.1	Website Code . . . . .	16

# List of Figures

2.1	A diagram showing the MVC architecture . . . . .	4
2.2	Screen from original game - <a href="https://arcadeblogger.com/2020/06/27/the-development-of-robotron/">https://arcadeblogger.com/2020/06/27/the-development-of-robotron/</a>	
2.3	Advertising Material - <a href="https://arcadeblogger.com/2020/06/27/the-development-of-robotron/">https://arcadeblogger.com/2020/06/27/the-development-of-robotron/</a>	
3.1	Class diagram . . . . .	9
3.2	Class diagram . . . . .	9
3.3	Class diagram . . . . .	10
3.4	Class diagram of characters . . . . .	10
3.5	Flowchart of MVC . . . . .	11
3.6	How tokens are generated . . . . .	11

# Chapter 1

## Abstract

In essence the project is an implementation of Robotron in PyGame, using Model-View-Controller, with a Flask based high scores board. The main content of the code is in the game itself, with flask acting only as an API. This allows for shared usage of the route by a static web page, and by the PyGame code itself. The webpage is simply served off as static, where JS is able to communicate with the API to retrieve the information needed. The database used is Postgres.

## Chapter 2

# Analysis

### 2.1 What is MVC?

Model-View-Controller plays a large part in the project, the diagram [Figure 2.1] shows the main way that MVC works. It isolates the components of the game into 3 main components. The View, which is the screen, or what the user will see. The controller, which is where the user interacts with the game, in this case it is the interaction with the keyboard. The model, which is the part the user never interacts with, and stores the state of the game and current information about it.

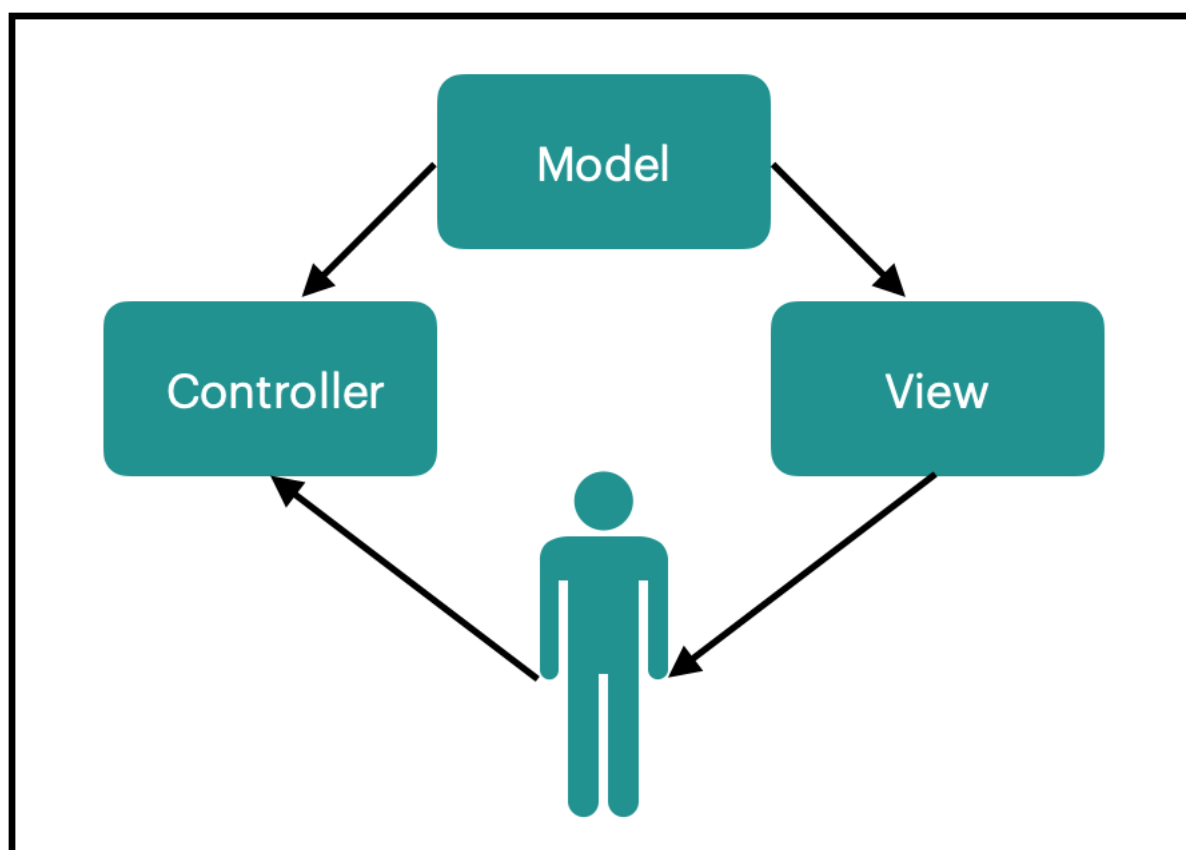


Figure 2.1: A diagram showing the MVC architecture

There are many benefits to this set up, for example, it will easily allow me to swap out what controller is used. If desired, it is much simpler to replace the keyboard as the human interface, and replace it with a game controller. Even more useful may be the ability to remove

the controller and view entirely, allowing for a streamlined game which an AI could learn how to play. This flexibility, along with ease of programming is what drew me to use MVC for the game.

Another important information is the way information travels between the 3 sections. This is done with events, and an event manager is responsible for maintaining the sending and receiving of events through the system. A similarly important section is the States, and state machine, which controls the current 'state' the game is in, that is to say what level is being played, or what screens should be shown, such as a loading or help screen.

## 2.2 The Game

"Robotron: 2084" was released in 1982 by Williams Electronics. It was revolutionary as a dual stick shooter, was high energy and loved by many. This is important to capture into the game, where I want it to have a similar feeling to the original game, with some modern twists.

The game is about a species of 'Robotrons' created by humans in the year 2084, after realising their failings and created an advanced species. The goal is to save the humans (Mommies, Daddies and Mikeys), whilst fighting the robots, which have many kinds. The most basic are electrodes, which are static obstacles that kill on contact, but can be shot by players. The other basic enemy is the grunt, which is simply a basic soldier, which kills on contact, but moves towards the player. There are some other robots that will be talked about and implemented later, but the details about them are less important.

## 2.3 Limitations

The dual stick shooter nature means the player uses one joystick to move, and one joystick to shoot. This is difficult to implement well with a keyboard, but a simple setup which I am using is having WASD to move, and IJKL to shoot. Holding 2 keys diagonally at the same time will result it movement in an angle, allowing for shooting in 8 directions, and moving in 8 too.

Robotron is a fast fast game, I had to slow it down slightly in order to make it more playable on my laptop, and so it does feel somewhat different to the original. However by slowing it as I have I have made it a much smoother game to play.

## 2.4 Objectives

1. Create basic playing ability
  - (a) Player can move in 8 directions
  - (b) Player can shoot in 8 directions
  - (c) Players animation is correct for direction of travel
2. Create basic enemies
  - (a) Enemy is spawned in random position
  - (b) Enemy can move
  - (c) Enemy is animated
  - (d) Enemy kills players
3. Create Loading Screens
  - (a) Fuzzy loading screen
  - (b) 'All test' screen

- (c) Home Screen
- 4. Create levels and transitions
  - (a) Player moves between levels
  - (b) Level transitions
  - (c) Player is invincible on load
- 5. Create the API
- 6. Create login system
  - (a) Basic API sign up works
  - (b) GUI interactions with PyGame
- 7. High Scores
  - (a) Top 10
  - (b) Player Search
- 8. Create sounds with Game
- 9. Create scoring and score counter
- 10. Create a life counter
- 11. Automate testing on API and basic functions in PyGame

## 2.5 Design and Inspiration

The design for all the game is heavily taken from the original game. I used many places to research this, but below is a selection of screenshots and videos which were used in the creation of the game.

- <https://www.youtube.com/watch?v=ccltMtkFBSI>
- <https://www.youtube.com/watch?v=aOVA2Axxfdk>

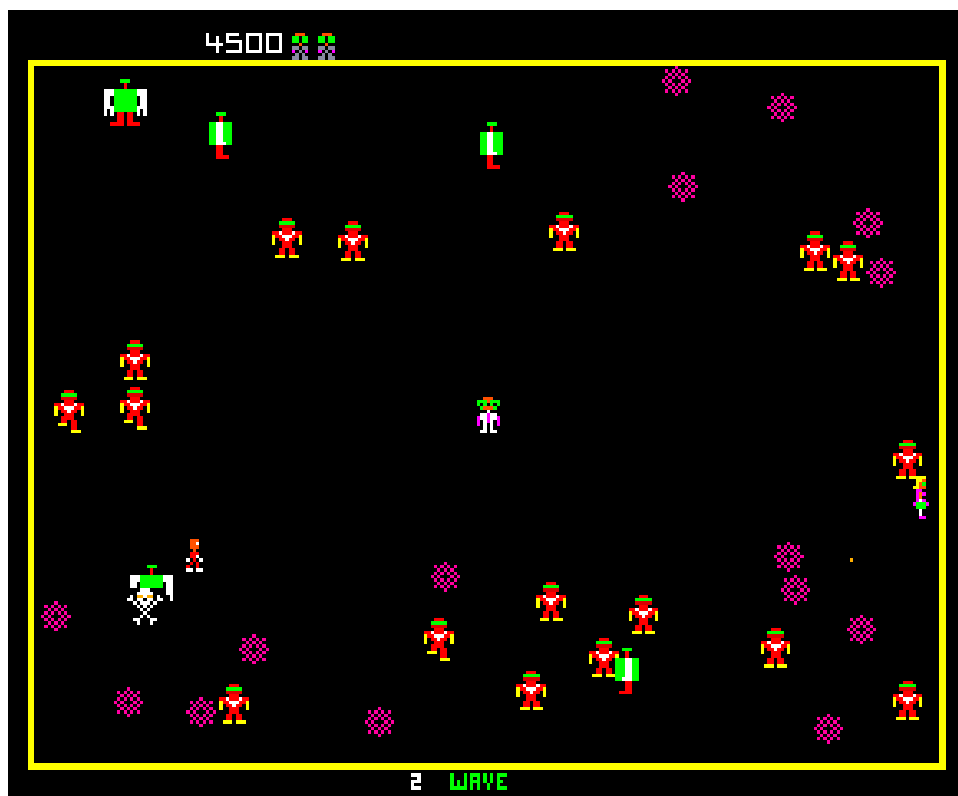


Figure 2.2: Screen from original game - <https://arcadeblogger.com/2020/06/27/the-development-of-robotron/>

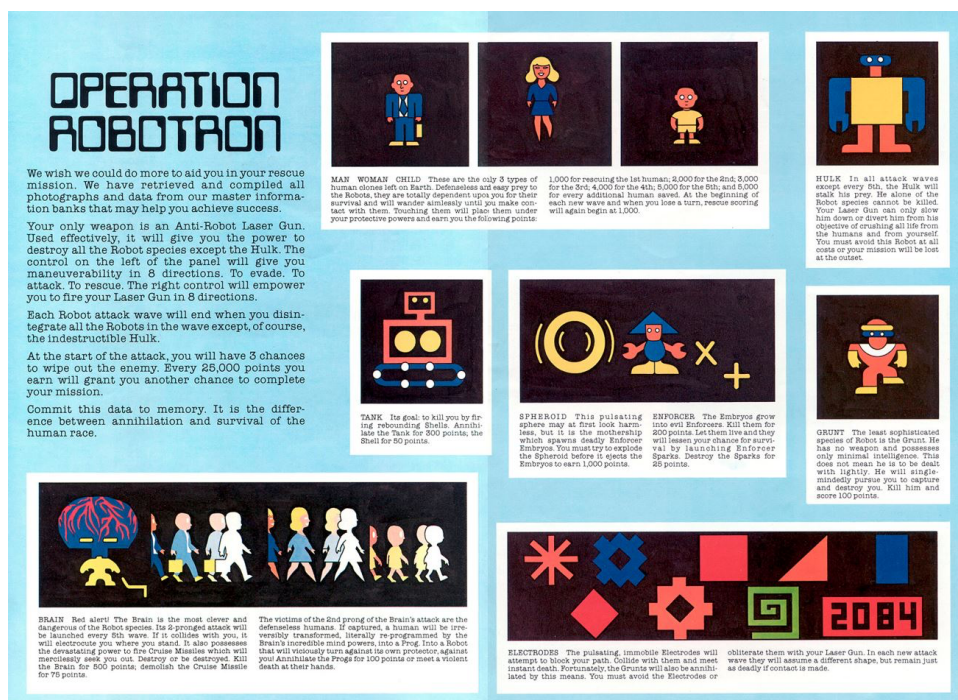


Figure 2.3: Advertising Material - <https://arcadeblogger.com/2020/06/27/the-development-of-robotron/>



## Chapter 3

# Documented Design

The main design aspect is the MVC architecture and how it forms the basis of the game. Fig 1, from the analysis section, gave a very brief, high level and non technical view of MVC. In this section I will go into more detail about my own implementation, and how it works in greater detail. This section also details the database on the web side, the API, the technical setup of the servers, the data structures and HCI designs.

### 3.1 MVC in practice

In the analysis section I gave a very high level overview of MVC, this part will detail further into my design on its implementation in python. The first main, basic components of MVC are of course, the model, the view, and the controller. Figure 3.3 shows the 3 classes diagrams for each of the implementations of these in python.

On top of these key features, there's also a range of other important cogs in the system. One of the most important, to allow for the communication between the M, V and C are Events, and an event manager. A Sample of events, and the event manager is given in Fig fig:events.

The other key class is the state machine. Each state is not given its own class, rather there is a constant number which is attributed to a given state. The states are used for the larger changes in the program and events are for the smaller interactions, and ticks.

In order to run through a basic idea of what happens when the program is run, I have created a step by step flowchart. This flowchart [Fig 9] is a gross oversimplification, but works as a high level description of what it is my code is doing when executed.

### 3.2 Database

This section will show the database design and set up, and explain some of the SQL used in the program. Fig 10 shows the database diagram.

[TODO - Database diagram]

There are 3 tables, scores, users and tokens. The scores database has 2 fields which store the users ID and their Score for a given game. The Users table stores the users info, such as emails, password hashes, etc, and then the tokens database is used to store validated tokens (with time limits) which are used to validate the GUI and avoids needing to login to the the program every time the game is run. Fig 11 shows the process of creating the tokens.

### 3.3 The API

The leaderboard contains only 6 routes, as these were all that are necessary, the details for the routes are detailed in the table below.

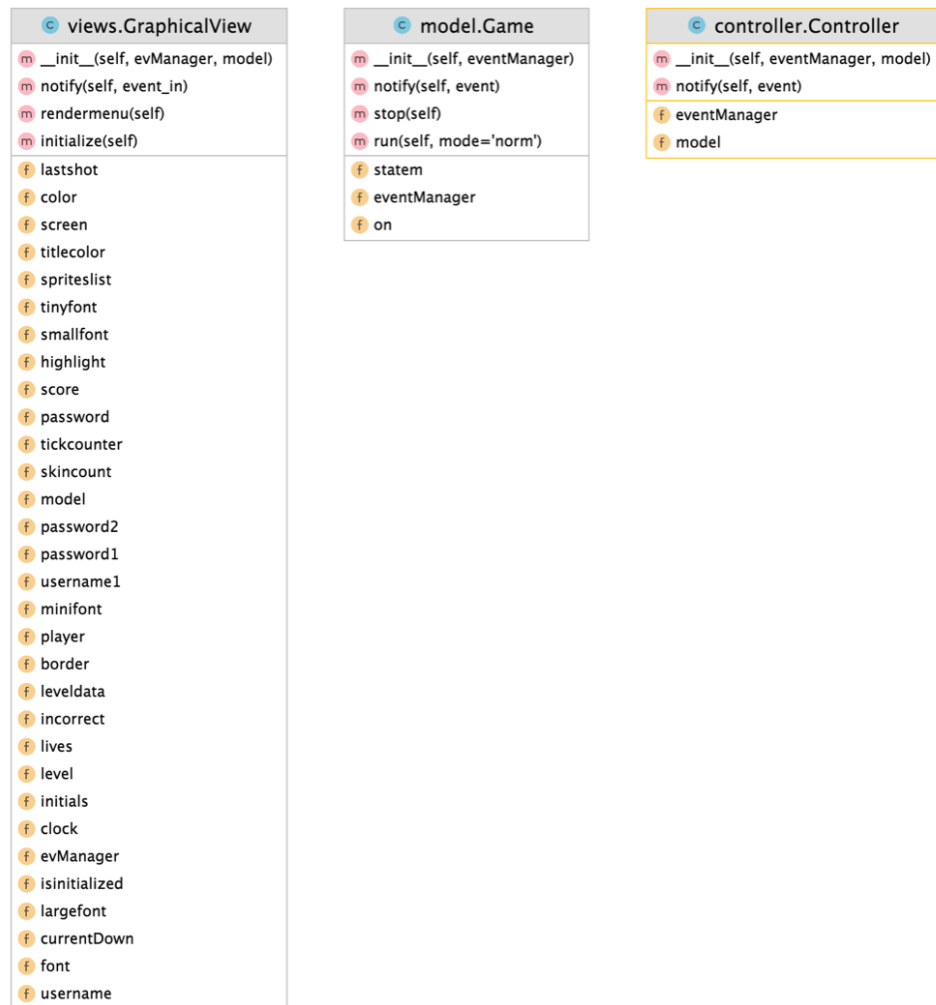


Figure 3.1: Class diagram

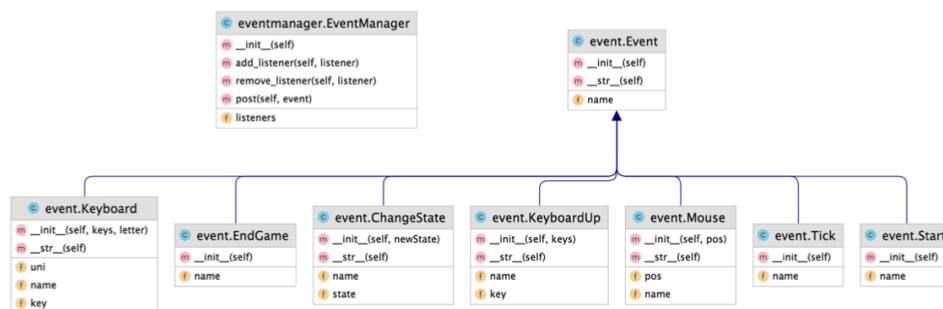


Figure 3.2: Class diagram

ROUTE	METHOD	DESCRIPTION
/leaderboard	GET	Returns JSON of top 10 users (initials + scores) in Database
/user/userid	GET	Returns JSON of top score
/username/userid	GET	Returns ID of given username
/login	POST	Logs in a user, sends token, or logs user in with token
/addscore	POST	Adds a score, given score and a token
/adduser	POST	Adds a user to the database

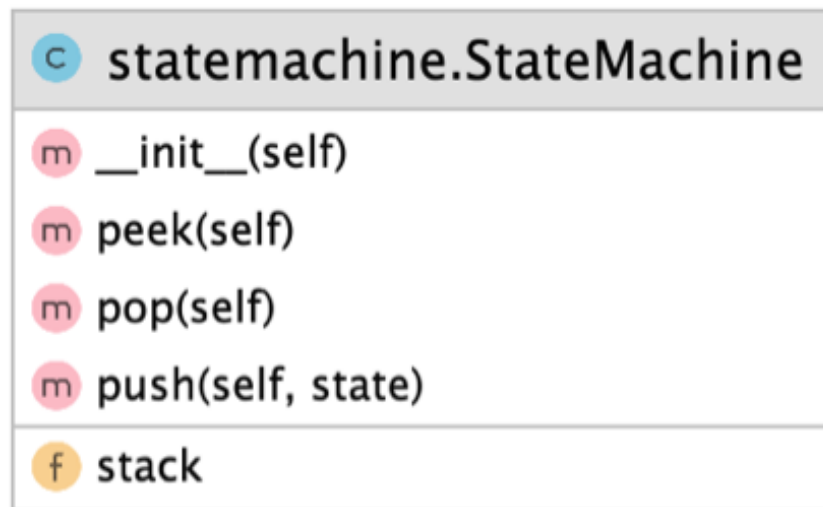


Figure 3.3: Class diagram

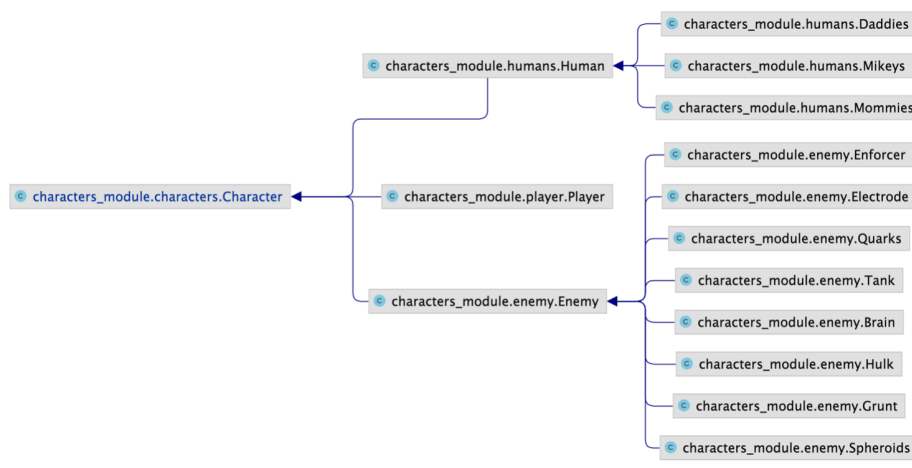


Figure 3.4: Class diagram of characters

### 3.4 The Server Setup

Fig 12 shows the set up the server is in. All using AWS, there is an RDS Postgres database, and EC2 instance (this is the server running the actual flask) and then an S3 bucket to handle sending the static files. It may also be possible to use NGINX or Apache to serve and handle the API. This system may end up being better, so my current architecture could change.

### 3.5 Security

Because the database and client handles personal details like email and passwords, there needs to be a thought to security. First off, there is an enforcement of passwords and a strong policy. Users passwords will need to be 8 characters, with 1 special, and my plan is to check them against a list of common passwords (rocky.txt) using hashes. For this I will probably use MD5,

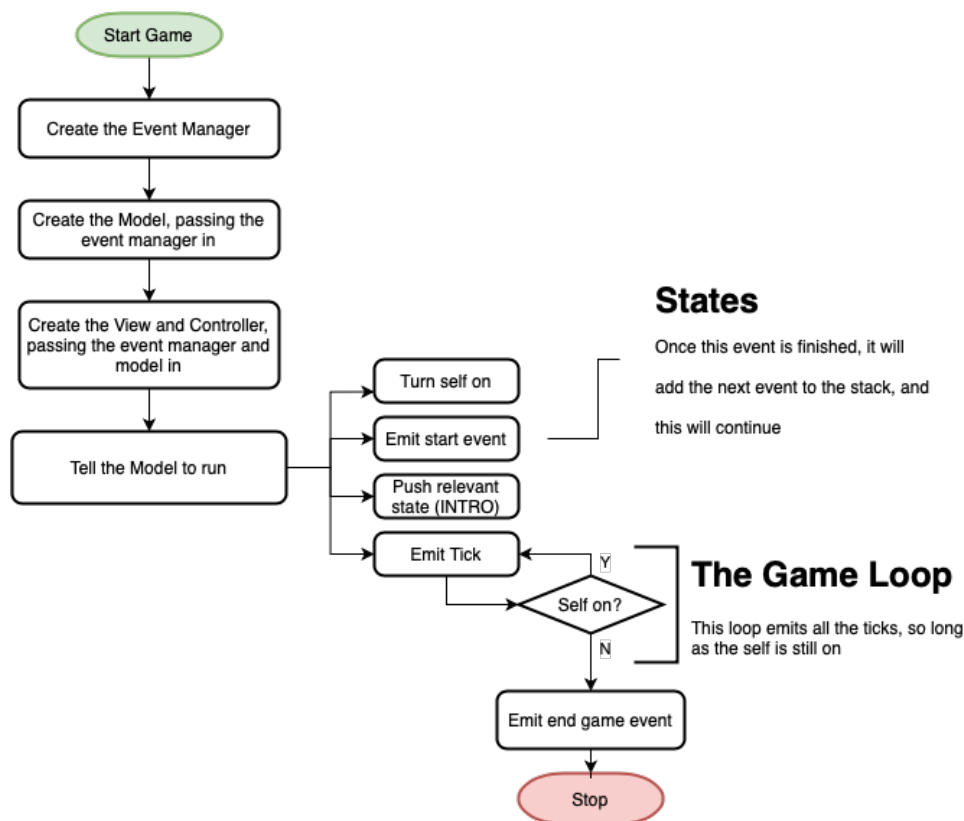


Figure 3.5: Flowchart of MVC

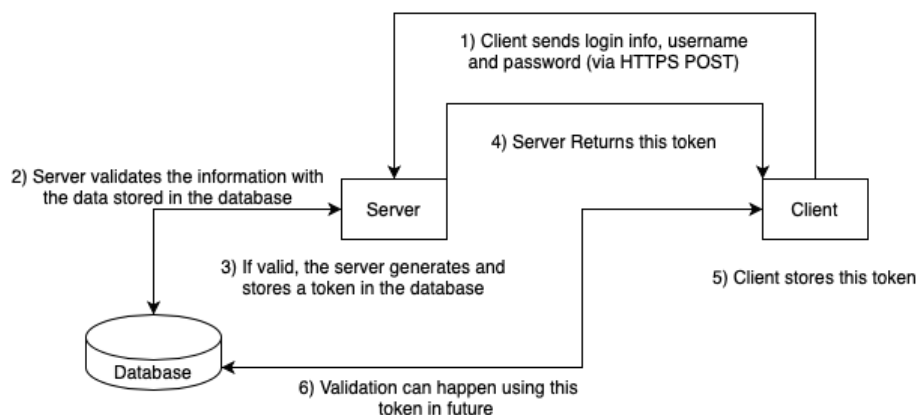


Figure 3.6: How tokens are generated

or something even faster. However it is important to avoid these fast algorithms when hashing passwords for storage. As such, passwords will undergo key derivation through bcrypt, an algorithm which not only salts, but performs many rounds of hashing. I could implement a similar algorithm using the basic functions like SHA, but rolling your own crypto is never good, so its going to be done with bcrypt, as this is essentially the best option available, and more than secure enough.

To help further security, HTTPS is being used for all the sending and receiving of data, this avoids man in the middle attacks of the data as it gets sent over the internet.

## Chapter 4

# Technical Solution

Im working on a way to easily do this. I am using LaTeX as this will allow me to not have to change the code on my report manually, as it will link to the files. However i am still learning. This file will also update on my github, so if you go to <https://github.com/john-montgomery2003/Robotron2084> you will be able to find a pdf of this document which is more up to date.

### 4.1 Example

## Chapter 5

# Testing - TODO

## Chapter 6

# Evaluation - TODO

## Chapter 7

# Appendix & Bibliography

### 7.1 Appendix

Name	Server/Web/Game/Dev	Use
Flask	Server	Handles the API and web on server side
SQLAlchemy	Server	Used to connect to the Postgres database
BCrypt	Server	Key derivation
Waitress	Server	WSGI server
PyGame	Game	Graphics and input handling
S3	Server	AWS static file hosting / serving
EC2	Server	AWS server to run flask app
Hetzner	Server	Alternative option to run flask and serve files
PyCharm	Dev	My IDE choice

### 7.2 Listings



# Listings

"Website Code/app.py" . . . . .	16
"Website Code/templates/index.html" . . . . .	20
"Website Code/templates/error.html" . . . . .	24

## 7.2.1 Website Code

app.py

```

1 from waitress import serve
2 # Flask is used to handle the web requests
3 from flask import Flask, jsonify, request, render_template
4
5 # Sql alchemy handles all SQL interactions, but rather than using and overly
   relying on the ORM,
6 # Ill use raw SQL commands. The SQL server is running on RDS (AWS) with
   PostgreSQL
7 from sqlalchemy import create_engine
8 from sqlalchemy.orm import scoped_session, sessionmaker
9
10 # Allow CORS - so it will work from both the webserver and python
11 from flask_cors import CORS
12
13 # This is used to hash passwords and validate them - could of used a different
   tool, or built it myself, but
14 # But this is prebuilt and purpose designed
15 import bcrypt
16 import secrets
17 # This starts the App
18 app = Flask(__name__)
19 # Allow the cors to work
20 CORS(app)
21 # Gets the database URL, creates the connection
22
23 engine = create_engine(
24     'sqlite:///test.db',
25     connect_args={'check_same_thread': False}
26 )
27
28 db = scoped_session(sessionmaker(bind=engine))
29
30
31 db.execute('''
32 CREATE TABLE IF NOT EXISTS leaderboard (
33     id INTEGER UNIQUE PRIMARY KEY AUTOINCREMENT,
34     initials VARCHAR(255),
35     username VARCHAR(255) UNIQUE,
36     password VARCHAR(255)
37 )''')
38 db.commit()
39 db.execute('''
40 CREATE TABLE IF NOT EXISTS scores(

```

```

41     id INT,
42     scores INT
43 )
44 '''
45 db.commit()
46 db.execute('''
47 CREATE TABLE IF NOT EXISTS tokens (
48     id INT,
49     token VARCHAR
50 )
51 '''
52 db.commit()
53
54 @app.route('/test', methods=['GET'])
55 def test():
56     return render_template('error.html')
57
58 @app.route('/', methods=['GET'])
59 def index():
60     leaders = db.execute('''SELECT leaderboard.initials, scores
61 FROM leaderboard
62 LEFT JOIN scores
63 ON leaderboard.id = scores.id
64 ORDER BY scores DESC
65 LIMIT 10;''')
66     # ...so we convert it into a dictionary
67     a, d = [], {}
68     for lead in leaders:
69         for column, value in lead.items():
70             d = {**d, **{column: value}}
71         a.append(d)
72     return render_template('index.html', a=a)
73
74
75 @app.errorhandler(500)
76 def page_not_found(e):
77     # note that we set the 404 status explicitly
78     return render_template('error.html')
79
80
81 @app.route('/robo/leaderboard', methods=['GET'])
82 def leader():
83     """
84     This route fetches the top 10 results from the server, allowing the page to
85     display the leaderbaord
86     :return:
87     """
88     # This returns a Result Proxy object...
89     leaders = db.execute('''SELECT leaderboard.initials, scores
90 FROM leaderboard
91 LEFT JOIN scores
92 ON leaderboard.id = scores.id
93 ORDER BY scores DESC
94 LIMIT 10;''')
95     # ...so we convert it into a dictionary
96     a, d = [], {}
97     for lead in leaders:
98         for column, value in lead.items():
99             d = {**d, **{column: value}}
100         a.append(d)
101     return jsonify(a)
102

```

```

103
104 @app.route('/robo/user/<string:userid>', methods=['GET'])
105 def user(userid):
106     """
107     This returns a users high score, given their ID - this means that the API
108     will have to fetch the ID first
109     Could it have used the username? probably.
110     :param userid:
111     :return:
112     """
113     score = list(db.execute(f'''SELECT score
114 FROM leaderboard
115 LEFT JOIN scores
116 ON leaderboard.id = scores.id
117 WHERE leaderboard.id = {userid}
118 ORDER BY scores DESC
119 LIMIT 1;'''))[0][0]
120     return jsonify({'score': score})
121
122 @app.route('/robo/userid/<string:username>', methods=['GET'])
123 def useridget(username):
124     """
125     This is used to get the id of a user, from their username (which has to be
126     unique)
127     Returns a 0 if the username is not unique
128     :param username:
129     :return:
130     """
131     userid = list(db.execute(f"""SELECT leaderboard.id
132 FROM leaderboard
133 WHERE leaderboard.username = '{username}'
134 LIMIT 1;"""))
135     try:
136         print(userid)
137         return jsonify({'id': userid[0][0]})
138     except IndexError:
139         return jsonify({'id': 0})
140
141
142 @app.route('/login', methods=['POST'])
143 def login():
144     """
145     Used to login to the game, returns a token which is used to verify the
146     user.
147     :return:
148     """
149     userid = request.values.get('userid')
150     password = request.values.get('password')
151     print(userid)
152     hashed = list(db.execute(f'''SELECT password
153 FROM leaderboard
154 WHERE leaderboard.id = {userid}
155 LIMIT 1;'''))[0][0]
156     valid = bcrypt.checkpw(password.encode(), hashed.encode())
157     if not valid:
158         return jsonify({'message': 'password fail'})
159     else:
160         try:
161             token = list(db.execute(f'''SELECT token
162 FROM tokens

```

```

163         WHERE id = {userid}
164         LIMIT 1;'''))[0][0]
165         return jsonify({'token': token})
166     except:
167         token = secrets.token_urlsafe(30)
168         db.execute(f"""INSERT INTO tokens (id, token)
169                     VALUES ('{userid}', '{token}');""")
170         db.commit()
171         return jsonify({'token': token})
172
173
174 @app.route('/robo/addscore', methods=['POST'])
175 def add():
176     """
177     Used to add scores to the database, uses a post request. Must provide a
178     password to add a score.
179     This might be slightly annoying, but adding in functionality for tokens and
180     storing them in python
181     feels like a lot of work, maybe I will, but I probably wont invest my time
182     there, I could always cache the
183     password inputted in the python code instead.
184     :return:
185     """
186     userid = request.values.get('userid')
187     score = int(request.values.get('score'))
188     token = request.values.get('token')
189
190     tokenDB = list(db.execute(f'''SELECT token
191                                FROM tokens
192                                WHERE id = {userid}
193                                LIMIT 1;'''))[0][0]
194     valid = token == tokenDB
195
196     if not valid:
197         return jsonify({'message': 'password fail'})
198     try:
199         db.execute(f'''INSERT INTO scores (id, scores)
200                     VALUES ({userid},{score});''')
201         db.commit()
202
203         return jsonify({'message': 'success'})
204     except:
205         return jsonify({'message': 'fail'})
206
207
208 @app.route('/robo/adduser', methods=['POST'])
209 def adduser():
210     """
211     This is the API used to add a user to the database, users provide a
212     username, initials and their password.
213     Password validation will be done client side, need to keep this app as
214     lightweight as possible.
215     :return:
216     """
217     username = request.values.get('username')
218     initials = request.values.get('initials')
219     password = request.values.get('password')
220
221     tostore = bcrypt.hashpw(password.encode(), bcrypt.gensalt()).decode()
222
223     db.execute(f"""INSERT INTO leaderboard (initials, username, password)
224                 VALUES ('{initials}', '{username}', '{tostore}');""")
225     db.commit()

```

```

221
222     return jsonify({'message': 'success'})
223
224
225 if __name__ == '__main__':
226     serve(app, host="0.0.0.0", port=80)

```

index.html

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0,
7 shrink-to-fit=no">
8     <title>Robotron</title>
9     <meta name="theme-color" content="rgb(194,1,0)">
10    <meta name="description" content="Robotron leaderboard for robotron by John
11 Montgomery - a pygame game.">
12    <link rel="icon" type="image/png" sizes="360x360" href="../static/img/
13 Screenshot%202020-12-13%20at%2020.56.23.png">
14    <link rel="icon" type="image/png" sizes="360x360" href="../static/img/
15 Screenshot%202020-12-13%20at%2020.56.23.png">
16    <link rel="icon" type="image/png" sizes="360x360" href="../static/img/
17 Screenshot%202020-12-13%20at%2020.56.23.png">
18    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter
19 -bootstrap/4.5.2/css/bootstrap.min.css">
20    <link rel="manifest" href="manifest.json">
21    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/aos
22 /2.2.0/aos.css">
23    <link rel="stylesheet" href="../static/css/styles.css">
24 </head>
25
26 <body style="background: rgb(0,0,0);max-height: 100vh">
27     <div data-aos="zoom-out" data-aos-duration="2000" style="margin-right: 1%;
28 margin-bottom: 0;margin-left: 1%;height: 98vh;width: 98%;margin-top: 1vh;
29 border: 3px dotted #9f095c;">
30         <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin
31 -left: 0px;height: 100%;width: 100%;border: 3px dotted #9f095c;">
32             <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;
33 margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #970b60;">
34                 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0
35 px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #970b60;">
36                     <div style="margin-top: 0px;margin-right: 0px;margin-bottom
37 : 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #900c64;"
38 >
39                         <div style="margin-top: 0px;margin-right: 0px;margin-
40 bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted
41 #900c64;">
42                             <div style="margin-top: 0px;margin-right: 0px;
43 margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
44 dotted #880e68;">
45                                 <div style="margin-top: 0px;margin-right: 0px;
46 margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
47 dotted #880e68;">
48                                     <div style="margin-top: 0px;margin-right: 0
49 px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
50 dotted #81106b;">
51                                         <div style="margin-top: 0px;margin-
52 right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;
53 border: 3px dotted #81106b;">

```

```

31         <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #7a126f;">
32             <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #7a126f;">
33                 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #721473;">
34                     <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #721473;">
35                         <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #6b1577;">
36                             <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #6b1577;">
37                                 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #63177b;">
38                                     <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #63177b;">
39                                         <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #5b197e;">
40                                             <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #5b197e;">
41                                                 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #541b82;">
42                                                     <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #541b82;">
43                                                         <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #4d1c86;">
44                                                             <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #4d1c86;">
45                                                                 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #451e8a;">
46                                                                     <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #451e8a;">
47                                                                         <h2 style="color: rgb(69,31,138);font-family: Conv_robotron-2084;text-align: center;margin-top: 9px;">robotron heroes</h2>
48                                                                             <div class="container" style="padding-right: 50px;padding-left: 50px;margin-top: 50px;">
49                                                                                 <div class="row" style="margin-right: -15px;">
50                                                                                     <div class="col">
51

```

```

    <h3 style="font-family:
Conv_robotron-2084;color: rgb(255,51,38);">1 > {{ a[0].initials }} - {{ a
[0].scores }}
```

```

Conv_robotron-2084;color: rgb(255,51,38);">
75
        <h3 style="font-family:
Conv_robotron-2084;color: rgb(255,51,38);">4 > {{ a[3].initials }} - {{ a
[3].scores }}</h3>
76
    </div>
77
    <div class="col">
78
        <h3 style="font-family:
Conv_robotron-2084;color: rgb(255,51,38);">9 > {{ a[8].initials }} - {{ a
[8].scores }}</h3>
79
    </div>
80
</div>
81
<div class="row" style="margin-right: -15px
; ">
82
    <div class="col">
83
        <h3 style="font-family:
Conv_robotron-2084;color: rgb(255,51,38);">5 > {{ a[4].initials }} - {{ a
[4].scores }}</h3>
84
    </div>
85
    <div class="col">
86
        <h3 style="font-family:
Conv_robotron-2084;color: rgb(255,51,38);">10 > {{ a[9].initials }} - {{ a
[9].scores }}</h3>
87
    </div>
88
</div>
89
</div>
90
<div class="row" style="margin-top: 10%; ">
91
    <div class="col">
92
        <h1></h1>
93
        <h2 style="color: rgb(69,31,138);font-
family: Conv_robotron-2084;text-align: center;margin-top: 9px;">play the
game</h2>
94
        <p style="font-family: Conv_robotron
-2084;color: rgb(254,51,38);text-align: center;margin-top: 16px;font-size:
16px;">Get the game -&nbsp;<a href="#">Github</a></p>
95
        <p style="font-family: Conv_robotron
-2084;color: rgb(254,51,38);text-align: center;margin-top: 16px;font-size:
16px;">Original game info -&nbsp;<a href="#">here</a></p>
96
    </div>
97
</div>
98

```



```

99         <h2 style="color: rgb(113,113,113);font-family:
100         Conv_robotron-2084;text-align: center;margin-top: 50px;font-size: 12px;">&
101         nbsp;by John Montgomery</h2>
102
103         </div>
104
105         </div>
106
107         </div>
108
109         </div>
110
111         </div>
112
113         </div>
114
115         </div>
116
117         </div>
118
119         </div>
120
121         </div>
122
123         </div>
124
125
126         <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min
127         .js"></script>
128         <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap
129         /4.5.2/js/bootstrap.bundle.min.js"></script>
130         <script src="https://cdnjs.cloudflare.com/ajax/libs/aos/2.2.0/aos.js"></
131         script>
132         <script src="../../static/js/script.min.js"></script>
133     </body>
134 </html>

```

error.html

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0,
7     shrink-to-fit=no">
8     <title>Robotron</title>
9     <meta name="theme-color" content="rgb(194,1,0)">
10    <meta name="description" content="Robotron leaderboard for robotron by John
11    Montgomery - a pygame game.">
12    <link rel="icon" type="image/png" sizes="360x360" href="../../static/img/
13    Screenshot%202020-12-13%20at%2020.56.23.png">

```

```

11 <link rel="icon" type="image/png" sizes="360x360" href="../../static/img/
Screenshot%202020-12-13%20at%2020.56.23.png">
12 <link rel="icon" type="image/png" sizes="360x360" href="../../static/img/
Screenshot%202020-12-13%20at%2020.56.23.png">
13 <link rel="icon" type="image/png" sizes="360x360" href="../../static/img/
Screenshot%202020-12-13%20at%2020.56.23.png">
14 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter
-bootstrap/4.5.2/css/bootstrap.min.css">
15 <link rel="manifest" href="manifest.json">
16 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/aos
/2.2.0/aos.css">
17 <link rel="stylesheet" href="../../static/css/styles.css">
18 </head>
19
20 <body style="background: rgb(0,0,0);">
21 <div data-aos="zoom-out" data-aos-duration="2000" style="margin-right: 1%;
margin-bottom: 0;margin-left: 1%;height: 98vh;width: 98%;margin-top: 1vh;
border: 3px dotted #9f095c;">
22 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin
-left: 0px;height: 100%;width: 100%;border: 3px dotted #9f095c;">
23 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;
margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #970b60;">
24 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0
px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #970b60;">
25 <div style="margin-top: 0px;margin-right: 0px;margin-bottom
: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #900c64;"
>
26 <div style="margin-top: 0px;margin-right: 0px;margin-
bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted
#900c64;">
27 <div style="margin-top: 0px;margin-right: 0px;
margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
dotted #880e68;">
28 <div style="margin-top: 0px;margin-right: 0px;
margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
dotted #880e68;">
29 <div style="margin-top: 0px;margin-right: 0
px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
dotted #81106b;">
30 <div style="margin-top: 0px;margin-
right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;
border: 3px dotted #81106b;">
31 <div style="margin-top: 0px;margin-
right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;
border: 3px dotted #7a126f;">
32 <div style="margin-top: 0px;
margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width:
100%;border: 3px dotted #7a126f;">
33 <div style="margin-top: 0px
;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width:
100%;border: 3px dotted #721473;">
34 <div style="margin-top:
0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;
width: 100%;border: 3px dotted #721473;">
35 <div style="margin-
top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;
width: 100%;border: 3px dotted #6b1577;">
36 <div style="
margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height
: 100%;width: 100%;border: 3px dotted #6b1577;">
37 <div style=
"margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;
height: 100%;width: 100%;border: 3px dotted #63177b;">

```

```

38                                     <div
style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px
;height: 100%;width: 100%;border: 3px dotted #63177b;">
39                                     <
div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left:
0px;height: 100%;width: 100%;border: 3px dotted #5b197e;">
40
<div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-
left: 0px;height: 100%;width: 100%;border: 3px dotted #5b197e;">
41
    <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin
-left: 0px;height: 100%;width: 100%;border: 3px dotted #541b82;">
42
        <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;
margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #541b82;">
43
            <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0
px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #4d1c86;">
44
                <div style="margin-top: 0px;margin-right: 0px;margin-bottom
: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #4d1c86;"
>
45
                    <div style="margin-top: 0px;margin-right: 0px;margin-
bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted
#451e8a;">
46
                        <div style="margin-top: 0px;margin-right: 0px;
margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
dotted #451e8a;">
47
                            <h1 style="color: rgb(69,31,138);font-family:
Conv_robotron-2084;text-align: center;margin-top: 9px;">UH OH</h1>
48
                                <h2 style="color: rgb(69,31,138);font-family:
Conv_robotron-2084;text-align: center;margin-top: 9px;">Something went wrong
:(</h2>
49
                                    <h2 style="color: rgb(113,113,113);font-family:
Conv_robotron-2084;text-align: center;margin-top: 50px;font-size: 12px;">&
nbsp;by John Montgomery</h2>
50
                                        </div>
51
                                            </div>
52
                                                </div>
53
                                                    </div>
54
                                                        </div>
55
                                                            </div>
56
                                                                </div>
57
                                                                    </div>
58
                                                                        </div>
59
                                                                            </div>
div>
60
                                                                </div>
61
                                                                    </div>
62
                                                                        </div>
63
                                                                            </div>

```

```
64         </div>
65     </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min
79 .js"></script>
80 <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap
81 /4.5.2/js/bootstrap.bundle.min.js"></script>
82 <script src="https://cdnjs.cloudflare.com/ajax/libs/aos/2.2.0/aos.js"></
83 script>
84 <script src="../../static/js/script.min.js"></script>
85 </body>
86 </html>
```