Twitch Plays TwiPi: An analysis and comparison with Twitch Plays Pokémon

Kedar More
Department of Mechanical Engineering
University of Colorado, Boulder
Boulder, USA
kedar.more@colorado.edu

***Abstract*— Social experiments on crowd-controlled gaming have been done on video games which are devoid of any physical obstructions/limitations. The best example is *Twitch Plays Pokémon* where players from all over the world played Pokémon Red by inputting commands (A, B, up, down, left, right, start, select) on chat while watching the live stream. This paper shows a detailed analysis of varying number of players (1, 2, 4, …) on physical wheeled robot TwiPi (Twitch Raspberry Pi Bot) and its comparison with video game (Pokémon Red). The infinite monkey theorem states that a monkey hitting keys at random on a typewriter keyboard for an infinite amount of time will almost surely type any given text, such as the complete works of William Shakespeare. This situation is quite like the theorem. The only and the most important difference is that instead of random hitting of keys, multiple intelligent entities are trying to solve a problem with a specific goal.**

Keywords—Twitch, Pokémon, Crowd-base Gameplay

I. INTRODUCTION

Twitch is a gamer's live streaming platform where this crowd-controlled game experiment was first tested on a Nintendo Gameboy Color Pokémon Red. This game was started on an emulator and live streamed on twitch.tv so that players from all over the world were able to see the character.

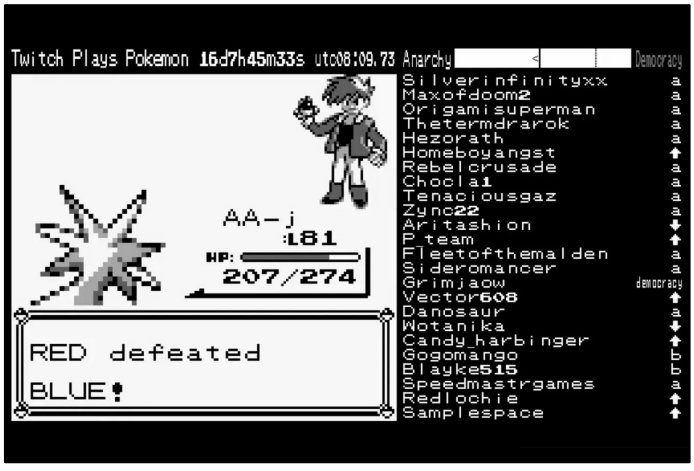


Fig. 1. Players inputting commands and those being excecuted

In the game the character must travel across the map in search of Pokémons and collect badges from in-game bosses. After collecting 8 badges the character can compete Elite 4 and become the Pokémon champion by defeating them. Among these tasks there are a few errands that the character runs to open new places on the map. There are a few tasks where the player has to decide between two routes e.g. you can reach Fuchsia City (green) from Saffron City (red) by going south of Lavender Town (blue) or by the bike path (yellow) near Celadon City. Each player has different approach to control the character and to choose between alternatives. This is where the conflict among the many players kick in. I have included some obstacles in the local game to demonstrate and compare the effect of this conflict.

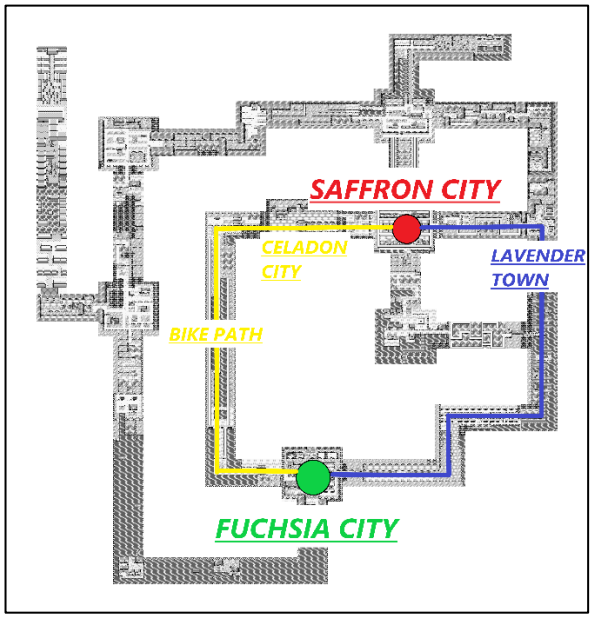


Fig. 2. Map of Pokémon Red with two route options

In TwiPi there are some physical hindrances such as ground friction and motor backlash. This change is addressed in this paper by using a physical robot as a character. Players completing the task remain the same as they are playing real-time. Here the players have to complete tasks with varying difficulty so that analysis of varying amount of time and number of commands inputs used to complete different tasks could be made. Also, in Twitch Plays Pokémon the commands were executed 30 to 40 seconds after they were input but, in this

experiment, there will be significantly short difference between input and execution.

II. Literature Review

A. A Crude Analysis of Twitch Plays Pokémon [1]

Twitch Plays Pokémon Red has two types of input methods namely Anarchy and Democracy. Anarchy executes all the commands which are input by all the players while democracy take a poll of 20 secs and executes the most voted command. This paper predicts the time taken to complete the game and the probability of winning for N players. This analysis is done with an assumption that there are no non-command texts from the players. The other assumption is that there are only two types of commands namely good and bad. In reality, there can be difference of opinion among the players where both approaches are correct.

B. Twitch Plays Pokémon: A Case Study in Big G Games Dennis [2]

As per the Infinite Monkey Theorem a goal can be achieved given enough time and effort. This paper analyses a similar phenomenon encountered in a social experiment called Twitch Plays Pokémon. Instead of monkeys there were actual people playing the game at the same time. This paper differentiates the players into categories namely explorers, achievers, socializers and killers or griefers and analyses different effects caused due to this socio-psychological experiment. It concludes that randomness caused by many intelligent entities not only achieves a goal but also delivers insightful and rich meaning behind this collective effort. This gave rise to many lore pertaining the Helix and Dome fossils (in-game items).

C. The dynamics of collective social behavior in a crowd controlled game [3]

Despite many efforts, the behavior of a crowd is not fully understood. The advent of modern communication means has made it an even more challenging problem, as crowd dynamics could be driven by both human-to-human and human-technology interactions. Here, the study of dynamics of a crowd controlled game (Twitch Plays Pokémon) is done in which nearly a million players participated during more than two weeks. Unlike other online games, in this event all the players controlled exactly the same character and thus it represents an exceptional example of a collective mind working to achieve a certain goal. A temporal evolution of the system dynamics was dissected along the two distinct phases that characterized the game. Having a fraction of players who do not follow the crowd's average behavior is key to succeed in the game. Most od theses phenomenon can be explained using Markow n^{th} order models. An analysis of the voting system between Anarchy and Democracy is done comprehensively. It is shown that the introduction of this system clearly polarized the community, splitting it in two. Finally, discussion on one of the peculiarities of these groups in the light of the social identity theory, which appears to describe well some of the observed dynamics is done.

III. Description of TwiPi

Designing of the robot for the experiment includes 4 parts:

1) *Physical body of the TwiPi*
2) *Processor to take inputs from the players and give commands to body*
3) *Server to connect the players to the robot*
4) *Integration of hardware and software*

A. Physical Body of TwiPi

The chassis used is Yahboom Raspberry Pi 4WD Robotic Car Kit [4]. The dimensions of the final assembly are 24cm*18.5cm*10cm. TwiPi is a mid-sized robot with 4 wheels. Material of the chassis is aluminum with holes for attachments of motors, expansion board and Raspberry Pi and the battery.

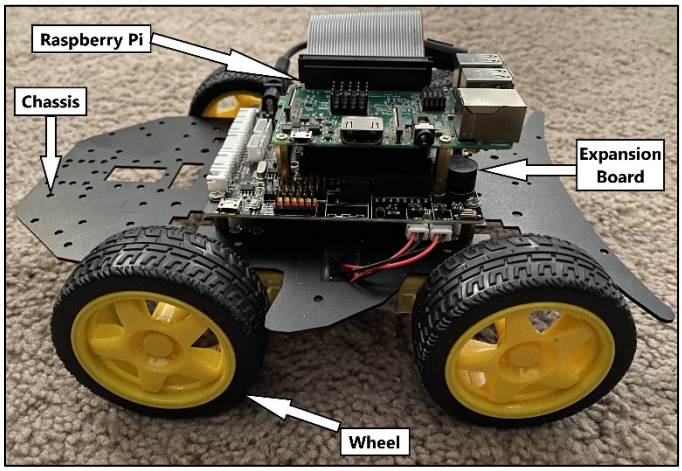


Fig. 3. Parts of Robot

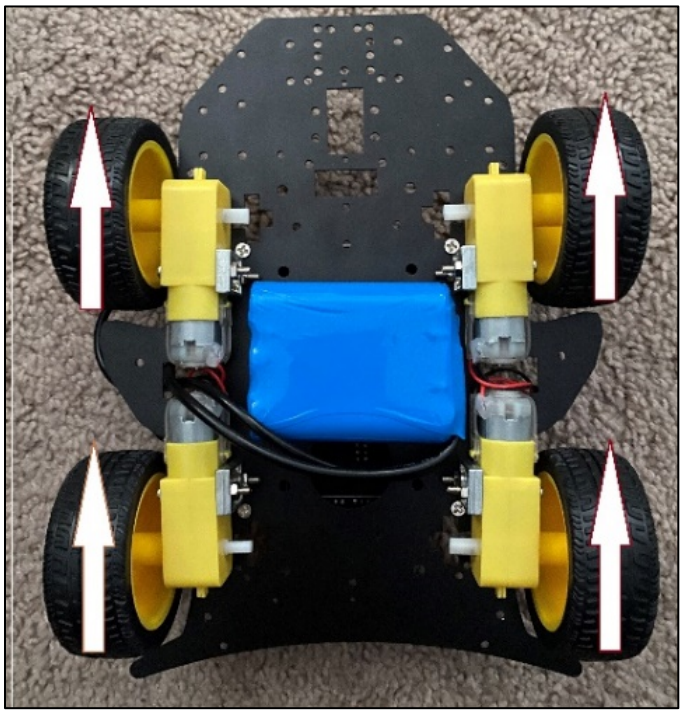


Fig. 4. Example of a Motion (Forward)

B. *Processor*

The Raspberry Pi is like a connection between the body and server. The server runs on the Pi and listens to the messages on the chat continuously.

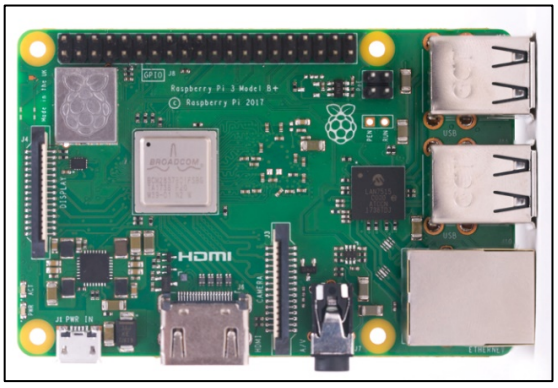


Fig. 5. Raspberry Pi

The expansion board in the kit has in-built motor drivers which take power input from external source and logic of direction and speed from Raspberry Pi to run the motors.

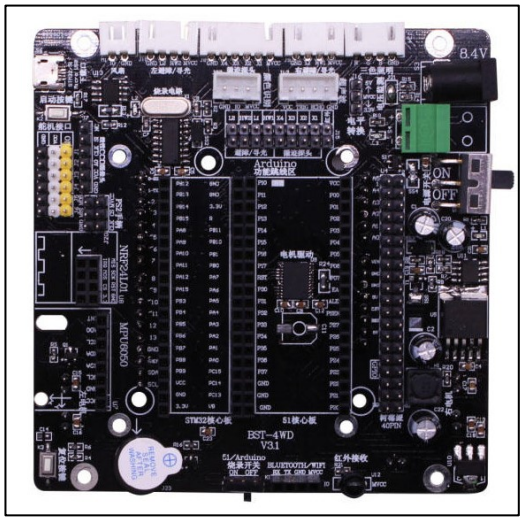


Fig. 6. Expansion Board

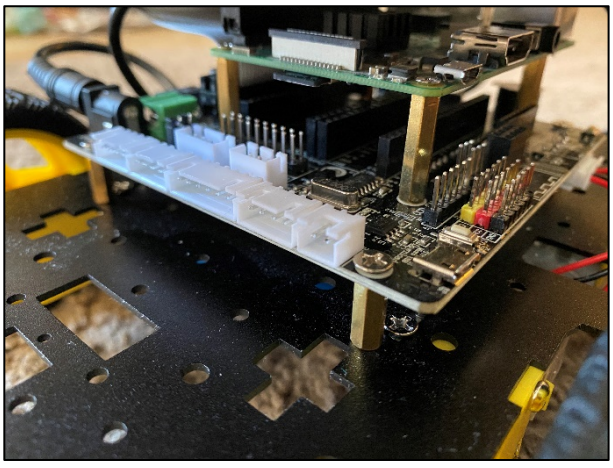


Fig. 7. Connection of Expansion Board to Processor (Raspberry Pi)

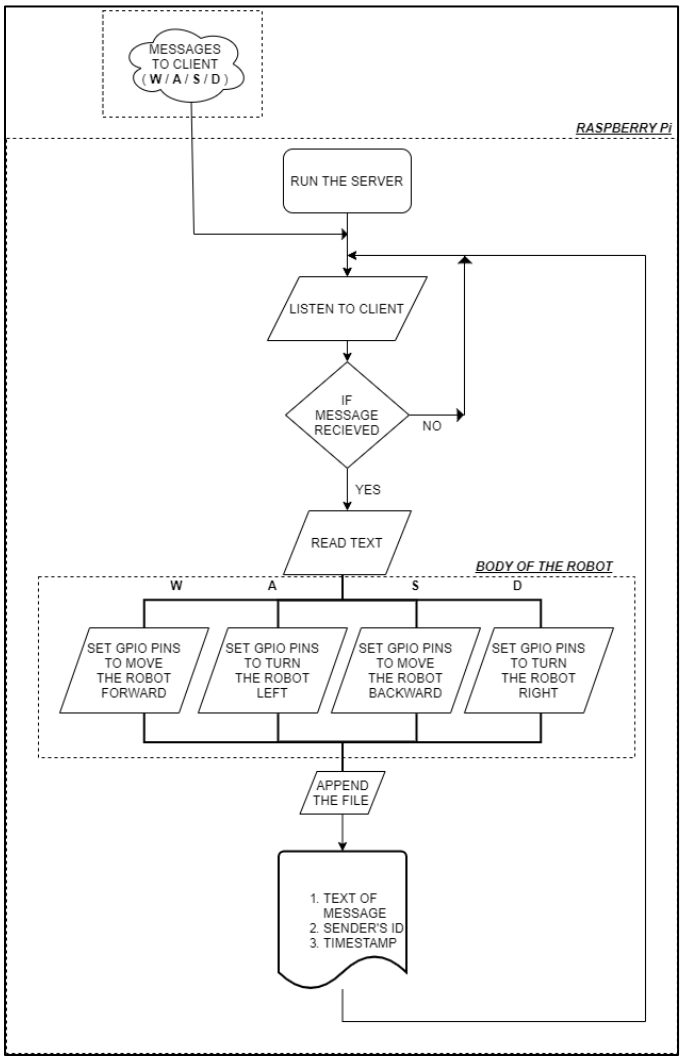


Fig. 8. Working flowchart

C. *Server*

1) *Local Django Server*

The commands are input by pressing the arrow buttons and the actual commands are received by the RPi. The RPi executes the command i.e. run the motors at the same time they are input. There is no Server Lag as everything takes place over the same Local Network.

A log file (.csv) is generated containing the raw data of command texts, Local IP address of the player and timestamp.

e.g.

w	10.0.0.40	3.24305796623
ww	10.0.0.225	3.4123980999
s	10.0.0.165	3.78674411774
s	10.0.0.225	4.33634495735
sa	10.0.0.40	4.43501591682
a	10.0.0.225	5.39895892143
ad	10.0.0.165	5.55254602432

w	10.0.0.40	5.77115797997
w	10.0.0.119	6.02172803879
w	10.0.0.225	6.91268897057

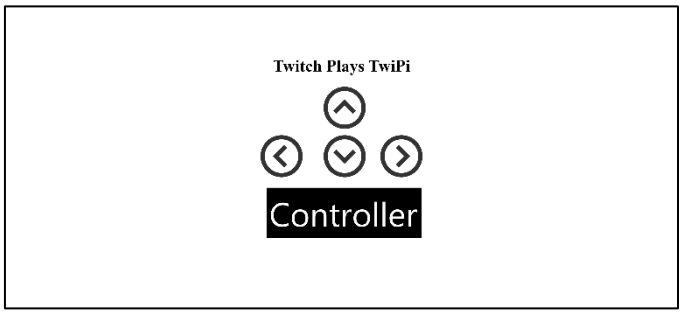


Fig. 9. Screenshot of Local Django Webpage to control TwiPi

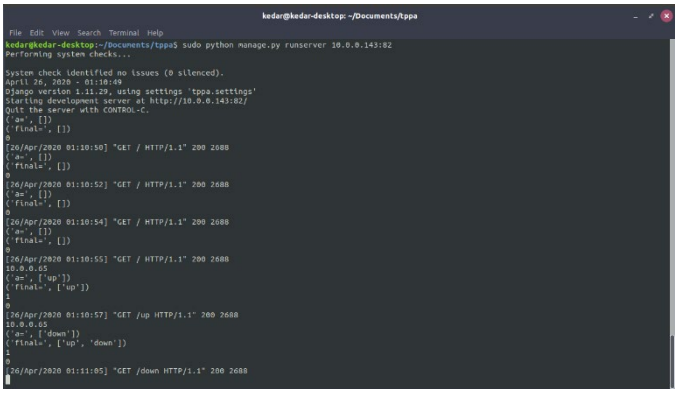


Fig. 10. Screenshot of Terminal with inputs from Django Server

2) YouTube Server

To take commands from the players over the internet, YouTube Live Chat is used with a Python library ytcaht. The players message the client on the chat window (right) with one of the WASD commands for forward, left, back, right respectively. This is done while watching the live stream (left) from the camera on TwiPi. The same and every command will be executed on TwiPi in the order of its input.

Log file:

e.g.

great bro	Hemant Kumar	15.734517097473145
w	Abhijeet D.Srivastava	20.893465757369995
s	nitik gupta	22.522578239440918
w	Abhijeet D.Srivastava	24.151798486709595
rr rocks	Han Solo	25.781222581863403
s	nitik gupta	27.21024799346924
s	nitik gupta	28.83951473236084
s	nitik gupta	30.468663215637207
w	Fifa Mobile	32.099449634552
w	Abhijeet D.Srivastava	33.13379645347595

There is a 5 to 8 seconds delay between the input of the command and its execution. In the original Twitch Plays Pokémon Red, there was a delay of 30 to 40 seconds in execution of commands. Even now when Twitch conducts the live game stream there is a 10 second delay.

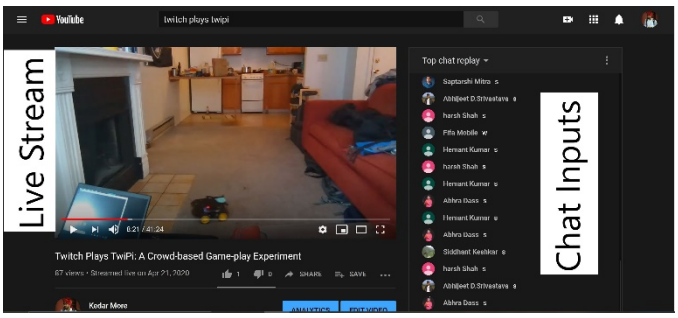


Fig. 11. Screenshot of YouTube Live Chat

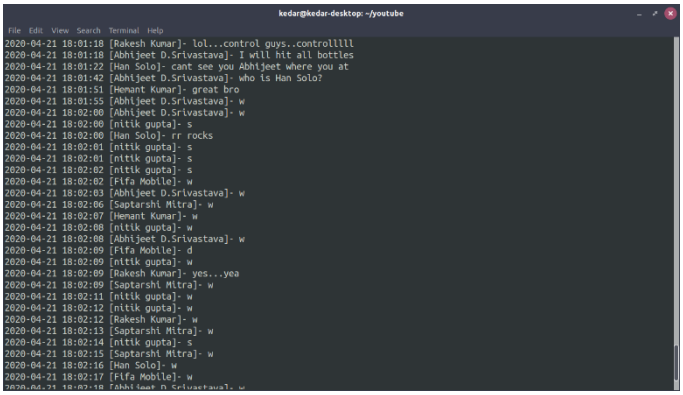


Fig. 12. Screenshot of Terminal with inputs from YouTube Server

D. Integration

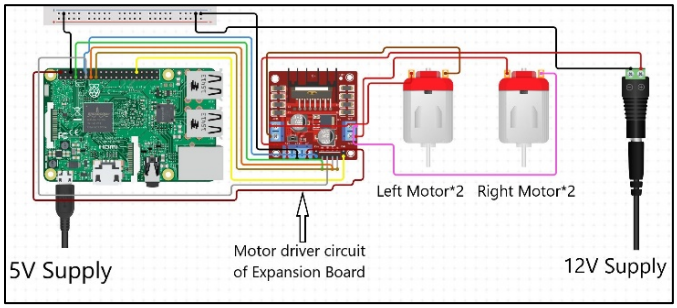


Fig. 13. Basic Circuit Representation

IV. EXPERIMENTS

A. Increasing difficulty (levels) with varying number of players (1, 2, 4).

The red circle indicates the starting position of TwiPi whereas the green circle indicates the targets it must reach. The red rectangle is levels 3, 4 and 5 is a barrier which has to be dodged by TwiPi.

The players can decide (individually) which target to reach first and which route to take for consecutives targets.

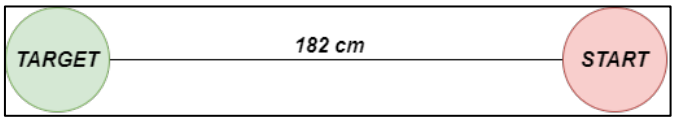


Fig. 14. Top view of Level 1 path

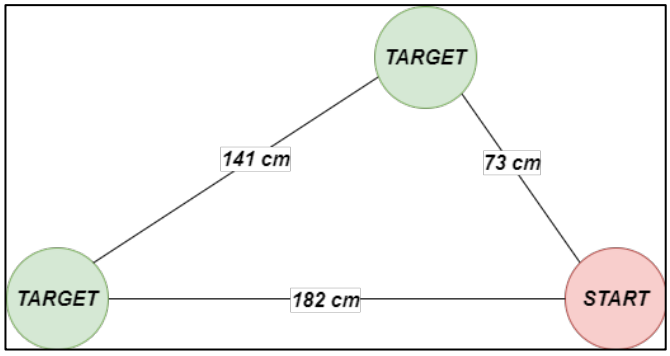


Fig. 15. Top view of Level 2 path

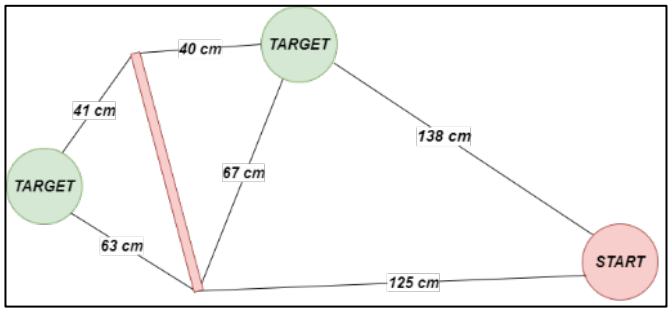


Fig. 16. Top view of Level 3 path

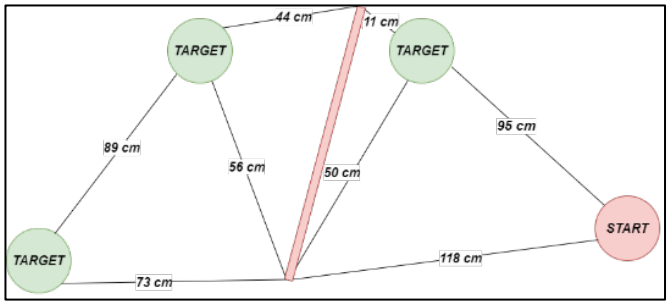


Fig. 17. Top view of Level 4 path

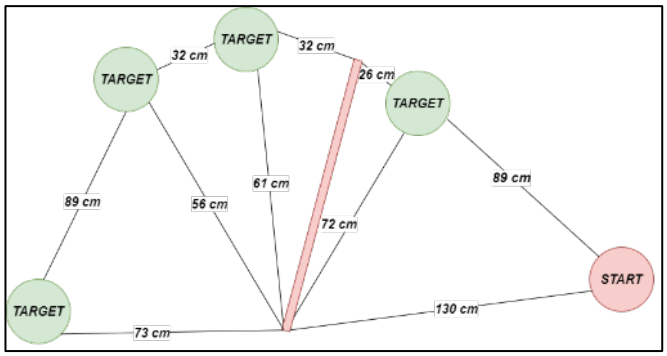


Fig. 18. Top view of Level 5 path

B. *Longer path with 6 and 29 players and non-continous participation*

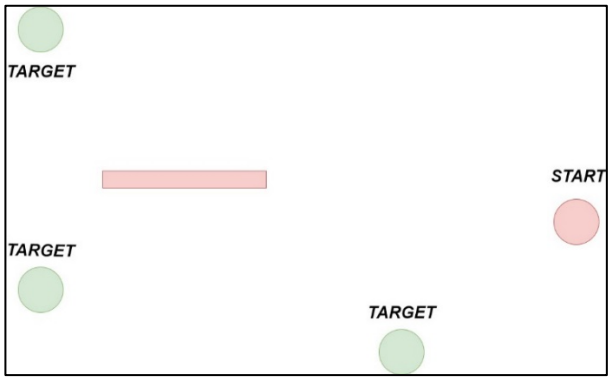


Fig. 19. Top view of Path for Live Stream with 29 players

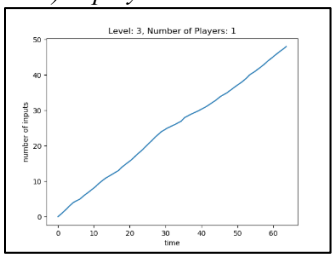
This path is similar to the Level 3 map in terms of layout, hence it will be easier to compare the outcomes with increasing no of total number of players playing.

While playing the commands were coming over the internet and there was a 5 to 8 seconds Server delay.

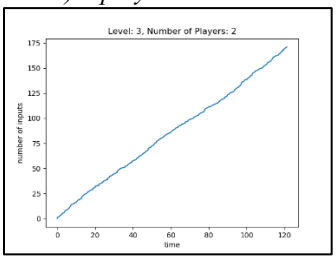
V. Observations on Less Number of Players (Experiment *A.*)

A. *Number of Inputs over Time (Level 3 path)*

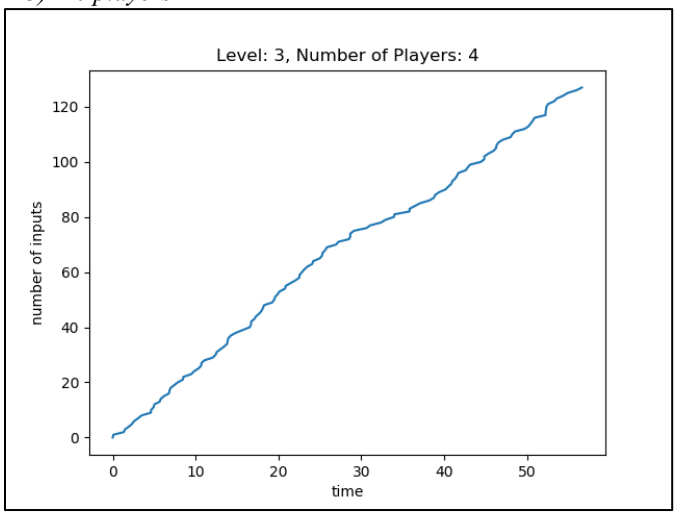
1) 1 player



2) 2 players



3) 4 players



As expected, the no of inputs increases in an irregular manner with time. The slope of this line can be used to find the change in value of inputs with time.

TABLE I. summaries the total time taken by varying no of players in the all the levels.

TABLE I. TOTAL TIME TAKEN TO COMPLETE THE LEVELS (SECONDS)

Levels	Number of Players		
	1	*2*	*4*
1	31.12	23.09	27.63
2	41.59	44.09	79.91
3	63.56	121.29	56.64
4	73.24	71.38	87.88
5	76.02	111.49	153.49

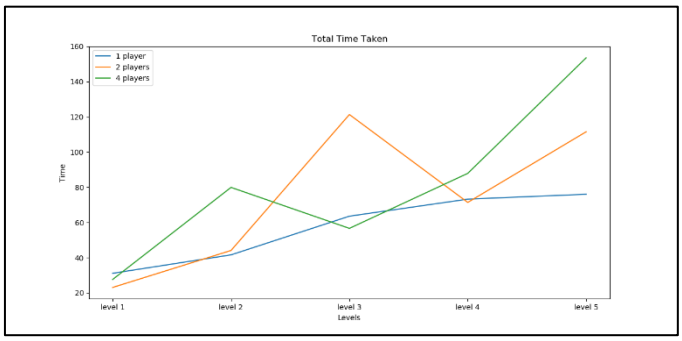


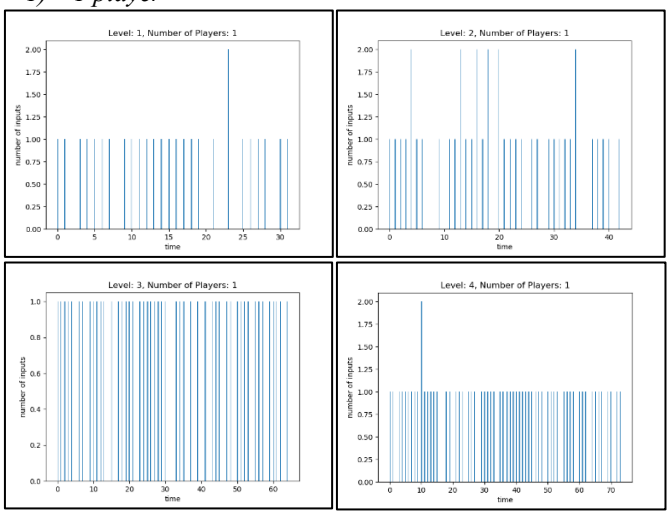
Fig. 20. Graphical Visualisation of Total Time

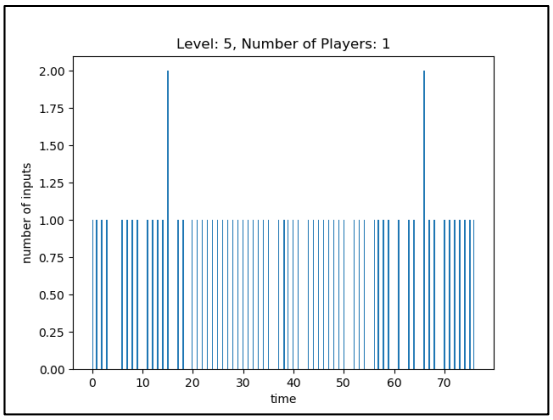
This trend of total time versus the number of players and complexity does not change linearly for lower number of players. But as the complexity will change significantly the total time will also increases almost linearly.

B. *Number of Inputs per Second*

A more interesting trend to observe would be the number of inputs given in the interval of 1 second. These graphs can be seen as slope of the previous number of inputs vs time graphs. For simplicity and understanding this slope is represented as bars in a discrete way. We can conclude about the interest of players in the game with the density of lines on the x axis.

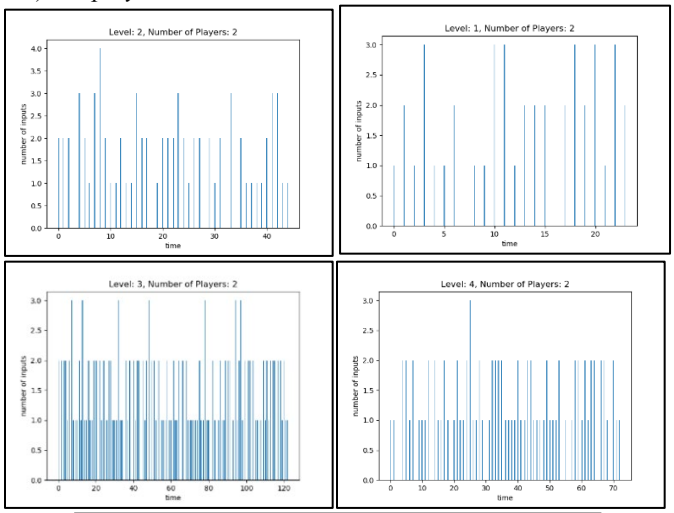
1) *1 player*

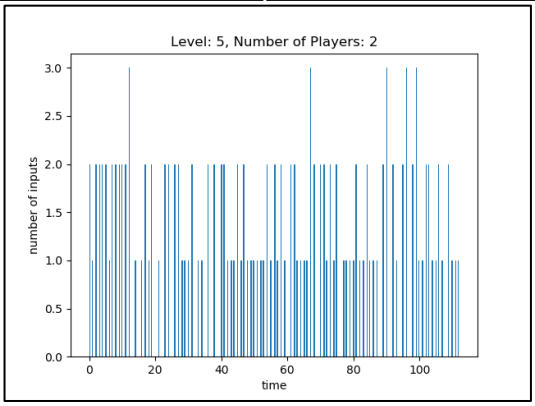




The increased density at the time 15, 39 and 65 seconds is due to the proximity of first target to the barrier, turning around the barrier and less space around the second target. The same trend is seen with 2 and 4 players. The only difference being the increased number of inputs per second.

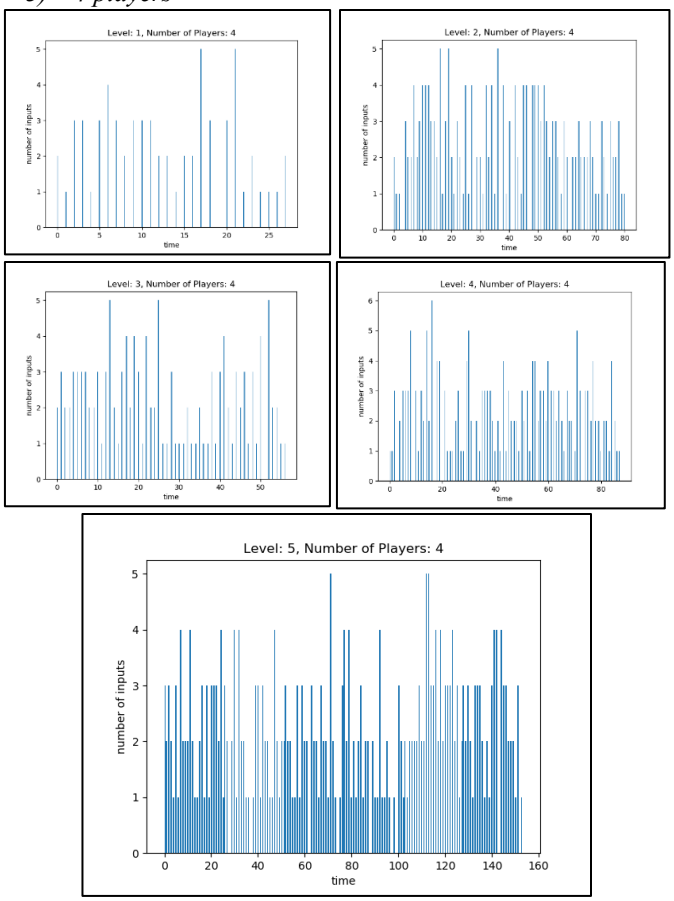
2) *2 players*





Here the increased density is more spread out due to the confusion created between the players about which route to take and the overlapping in the inputs. Overlapping of inputs is the major cause of anarchy in crowd-based gameplays. When more than one player thinks of a move and tries to implement it at the same time the character (TwiPi in this case) considers these to be multiple inputs and implements everything.

3) 4 players



We can make out the high-density patches at 15, 75 and 140 seconds. This is similar to the single player with mapped time interval. It is natural for the confusion to rise with many players and close spaces. There are a few patches where exact inputs are needed to complete the obstacle. Even if one player inputs a wrong command the obstacle is to be started all over again.

Such an obstacle was faced in Pokémon Red which is famous by the name “The Ledge”.

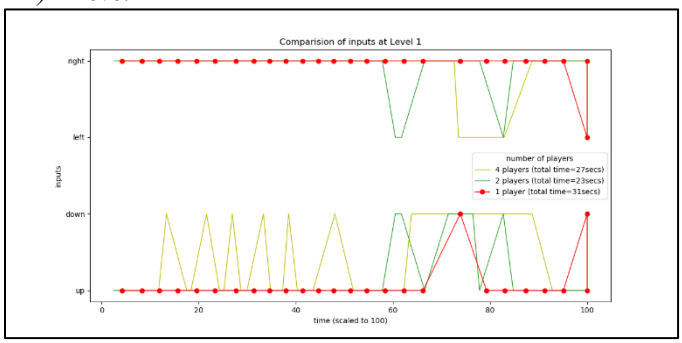
Fig. 21. Screenshot of Pokémon Red with the character jumping over “The Ledge”

Ledges are in-game obstacle from which we can jump down but can’t climb up. Such obstacles need much cooperation and are also reasons of many inputs given at the same time.

C. Comarision with single player inputs

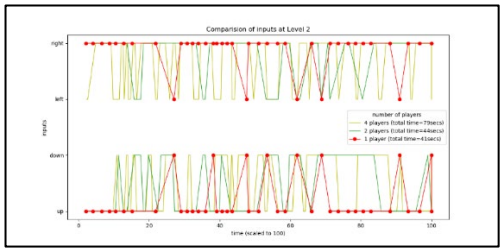
The Red dots indicate the correct inputs (single player) which should be given in order to complete the task. Time of all the inputs (1, 2 and 4 players) is scaled to 100 for comparison along the path paved by single player. This enables us to check the trend of inputs at the same position along the path (at different times as per number of players).

1) Level 1

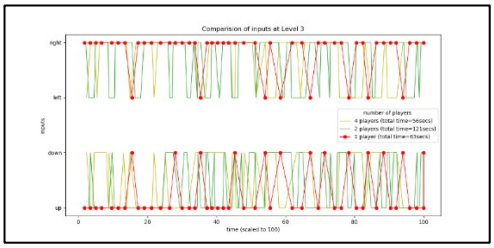


From this graph we can easily make out the one of the players was inputting ‘down’ instead of ‘up’. This type of player in crowd-based gaming is known as a ‘Rouge’. These players want to create more chaos and confusion within the game in turn increasing the total time. After 60 seconds the chaos was started to align TwiPi in the straight direction. This is where overlapping of inputs come in. Some commands are given to rectify this overcompensation.

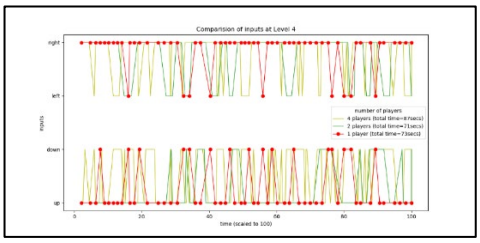
2) Level 2



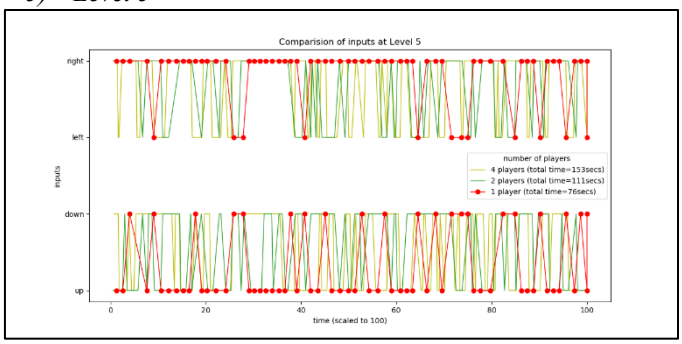
3) Level 3



4) Level 4



5) Level 5



All other graphs show total chaos by the players. We can see a similar trend in the inputs from single player and multiple players. The reason is that this is a simple task with limited possibilities for completion. When the complexity of the task increases there will be many possibilities with which all the players will be divided. The confusion created in that situation will be more in terms of randomness.

VI. Observation on Many Players (Experiment *B.*)

A. Number of Inputs per Unit Time

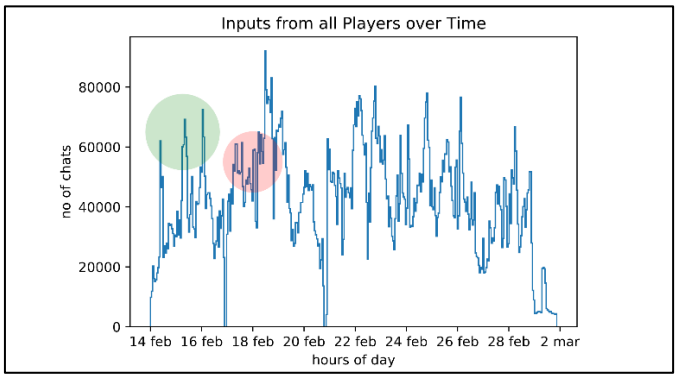


Fig. 22. Command inputs in Twitch Plays Pokémon Red per hour [5]

Here the green patch is when first 2 badges were obtained. Here the players gained some confidence to play the game enthusiastically. After that when the game reached a point where the progress was slow the viewership decreased.

The red patch is just before the concept of democracy was introduced. It was a voting system of 20 secs where the command with maximum votes was executed. It made the game slightly easy and less random. This also gave rise to two side: pro-democracy and pro-anarchy (the earlier system). This interest gave a boost in the viewership.

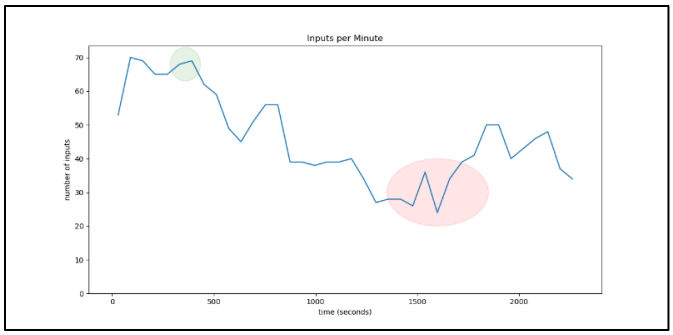


Fig. 23. Command inputs per minute with 29 players

The green patch indicates the knock over of the first target. When people sense that they can achieve something, and they are very close to it they will put extra effort towards it. And after they have achieved it there will be a slack in their efforts. This can be seen with the decrease in inputs after the green patch.

The red patch is when the difficulty of the targets was reduced. This gave a boost to the players to complete the game in less time.

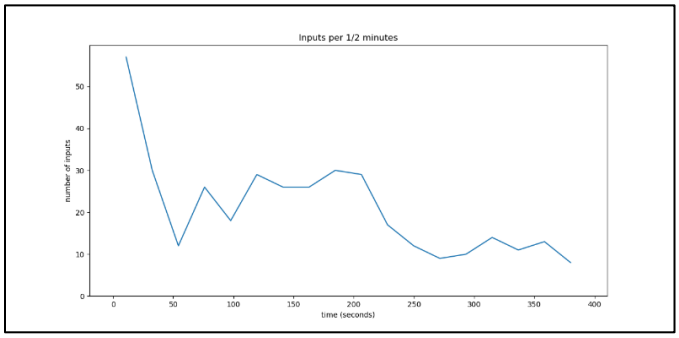


Fig. 24. Command inputs per ½ minute with 6 players

B. Number of Chats per Player

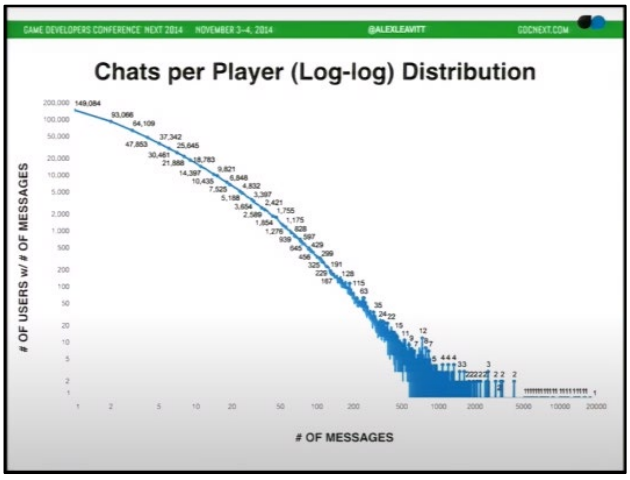


Fig. 25. Twitch Plays Pokémon Red [6]

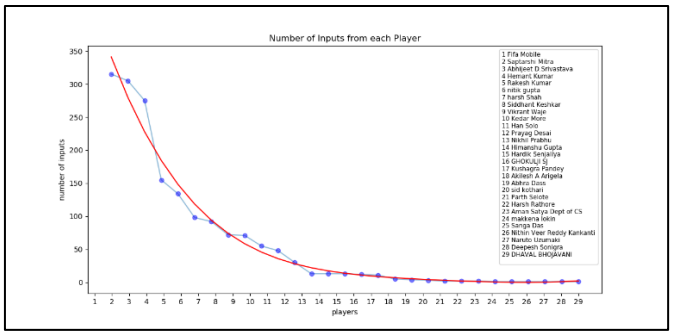


Fig. 26. Local Twitch Plays TwiPi

We can see a similar trend in every crowd where some people contribute the most and the contribution decreases in a logarithmic fashion (red).

C. *Trend of the most active players*

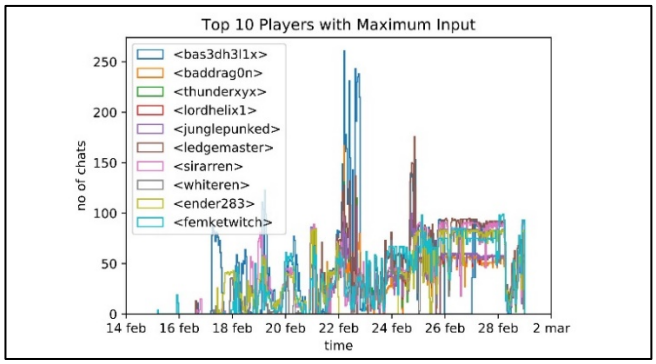


Fig. 27. Twitch Plays Pokémon Players [5]

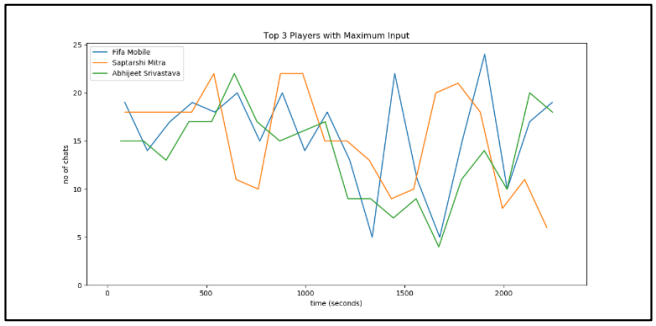


Fig. 28. Twitch Plays TwiPi Players

D. *Correct vs Incorrect Inputs compared to Gameplay of Single Player*

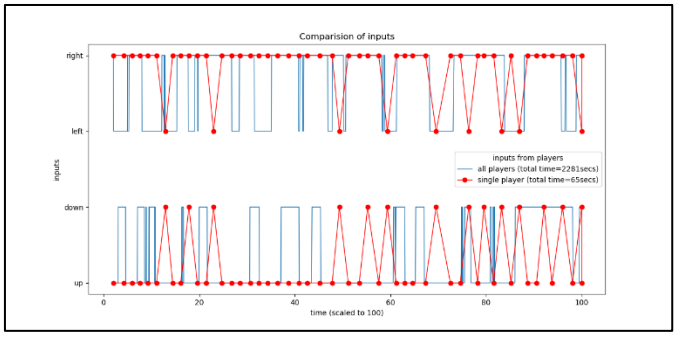


Fig. 29. Local Twitch Plays TwiPi comparison of inputs

The percentage of correct inputs with respect to all inputs is 66.98%. This percent is calculated based on scaled time where TwiPi is almost at the same position with both single and multiple players.

E. *Comparision of Total Time on similar Level of Difficulty (Level 3)*

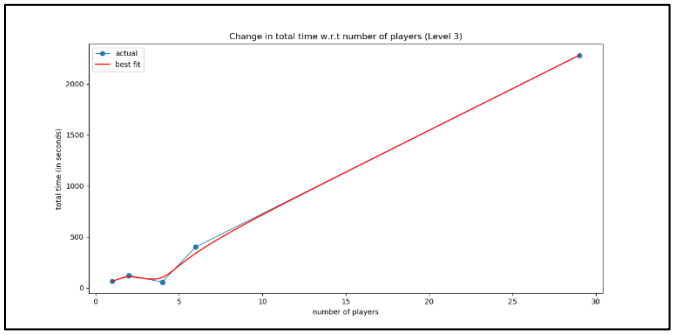


Fig. 30. Actual total time taken in Twitch Plays TwiPi

The first 3 points represent 1,2,4 players and the last 2 represent 6 and 29 players. We can also see a dip in total time at around 4 players and the linear increase after that.

VII. Conclusion

A. *Conclusion 1, Le Bon's Theory*

We can see peaks in the number of chats if there is an interesting event going on with the game. This was seen in Twitch Plays Pokémon when the system of Democracy was introduced on 19th February, 2014. Similarly, with TwiPi when targets were brought closer at around 1500 seconds the players started inputting more commands. This shows that if the players feel that the difficulty level has decreased (there is a higher chance of completing the game) by a certain amount, they start engaging in the game more than usual. This conclusion can be derived from The Crowd: A Study of the Popular Mind by Gustave Le Bon [7] that Crowds are only powerful for destruction. Here the destruction refers to completion of goals or some interesting events of conflict.

B. *Conclusion 2, Trend of the Total Time with Increase in the Number of Players*

In the game Pokémon Red, the average time of completing the game by a single player is 8 to 25 hours (16.5 hours average). Twitch Plays Pokémon with 748006 players took 16 days, 9 hours, 55 minutes, 4 seconds (394 hours) which is 23.87 times more than the average player. While, the time taken by 29 players in Twitch Plays TwiPi is 2281seconds which is 35.88 times the time taken by a single player. We cannot directly compare the gameplay of TwiPi and Pokémon Red, but we can see that with a significantly large number of players and average game time, the multiplication factor of the total time as that of a single player reduces. This contradicts the results of the paper [1]:

Expected total time of the game is given by:

$$Et = \frac{n}{N\lambda_h(2qe^{-\lambda_h(N-1)T_d} - 1)}(1 + N\lambda_h T_d)$$

Et =Expected total time

n = no of total inputs to win the game (53 in this case as a single player took 53 inputs to complete the path)

N = number of players

λ_h = inverse of average reaction of human (0.1/sec in this case)

q = probability of inputting a wrong command (1 as all the players know the correct inputs)

T_d = Observation delay (1.5 secs = Execution time)

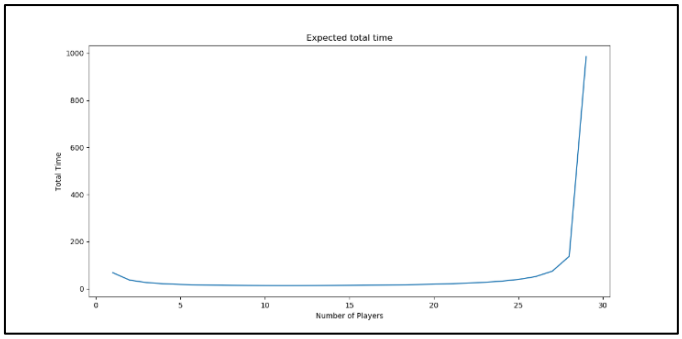


Fig. 31. Plot of Expected total time

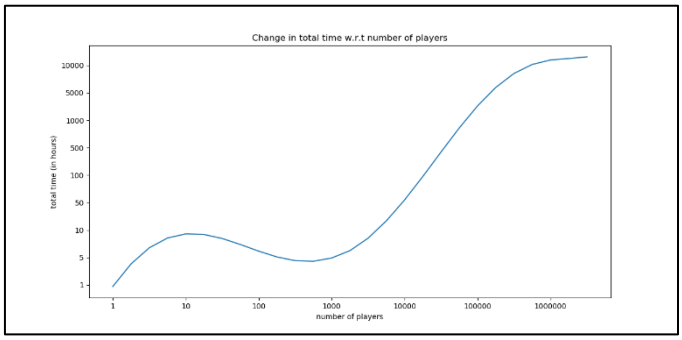


Fig. 32. Concluded Trend of Expected total time

When the path or gameplay is known by the players before-hand it becomes easier to navigate and due to many commands given at the same time the goal is reached in less time. The above graph shows a trend concluded when the path or the game is known to the players. The rise in total time is evident owing to the randomness created. The dip in the time is when most players input the same (correct) command and the implementing in takes lesser time than less no of players inputting the commands. As the observed accuracy of the commands is nearly 70% there will be more correct inputs to cancel out the wrong ones. After that, the 30% incorrect input also become significant making it difficult to navigate correctly. To compensate for the incorrect inputs more no of correct ones are needed. This takes significantly more time than lesser number of players. After the number crosses a certain amount the change does not matter as much. Hence the curve flattens are reach an asymptotic value.

C. *Conclusion 3, Effect of Server Lag*

The total time taken by the players in Level 3 path is 63.56, 121.29, 56.64, 391.57, 2281 seconds respectively. The total time taken by 2, 4, 6 and 29 players is 1.91, 0.89, 6.16 and 35.88 times more than the time taken by single player. Thus, the total time does not increase in proportion to the number of players alone. There are other parameters which must be taken into consideration.

The total time required depends on number of players, average time required by single players, difficulty which includes (number of correct inputs needed and how far can you go away from target). This was seen in the earlier mentioned paper. But from these experiments one more parameter has shown up along with the obvious ones which is the lag experienced. There are two kinds of lags:

1) *Server Lag*
2) *Execution Time*

A. *Server Lag*

It is the lag experienced between the actual input of the command and the visualization if its execution to the player. In the experiment with 1,2 and 4 players there is no lag between the execution and implementation. For the experiment with 6 and 29 players we experienced a lag of 5 to 8 seconds. This led to the non-linear rise in the total time taken after 6 players.

B. *Execution Time*

This is the actual time taken by the robot to implement the input command. This increases the total time with increase in number of commands.

Both these lags lead to a confusion among the players who play the game according to the current situation and not with the future reference. The greater the delay, more is the total time required.

VIII. References

[1] Chen, S. D. (2014). A Crude Analysis of Twitch Plays Pokémon, 1–6.

[2] Ramirez D, Saucerman J, and Dietmeier J (2014) Twitch Plays Pokémon: a case study in big G games. In: Proceedings of DiGRA 2014, 3–6 August, Snowbird, UT.

[3] Aleta, A., Moreno, Y. The dynamics of collective social behavior in a crowd controlled game. EPJ Data Sci. 8, 22 (2019).

[4] https://category.yahboom.net/products/4wdrobot.

[5] https://archive.org/details/tpp_logs.

[6] A Leavitt. 2014. TwitchPlayedPokémon: An Analysis of the Experimental Interactive Phenomenon. In *Game Developers Conference*.

[7] Le Bon, Gustave, 1841-1931. The Crowd: A Study of the Popular Mind. New York: Penguin Books, 1977.