George Andersen

QWOP in JavaScript

Computer Science Tripos – Part II

Robinson College

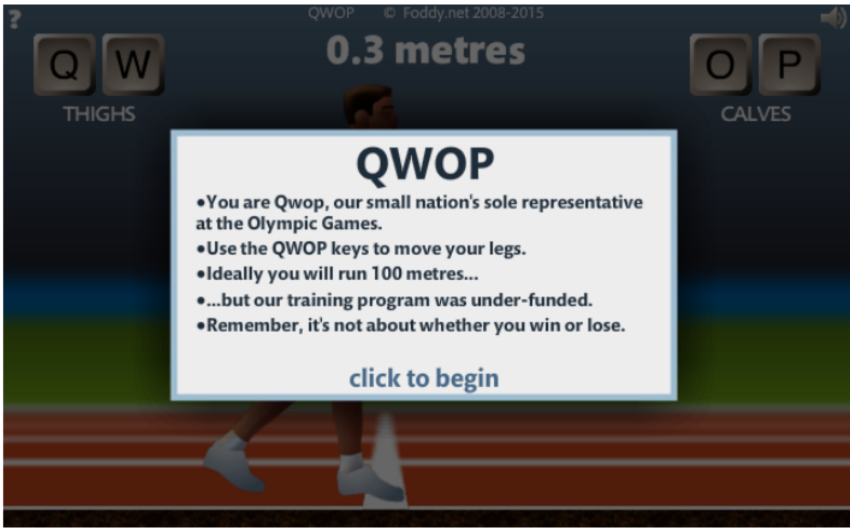
April 10, 2018

Chapter 1

Introduction

1.1 What is QWOP?

As described in the project proposal, my project is to recreate QWOP using JavaScript. QWOP is a browser video game created by Bennett Foddy. The goal of the player is to complete a 100m race without falling over. Figure 1.1 shows the screen on game start. Players control the athlete using only the Q, W, O, and P keys. Whilst the Q key is held down, the runner's right thigh is driven forward and his left thigh is driven backward. This moves the legs apart at the bottom of the torso in a scissor-like motion. The W key does the opposite of this, and holding it down will put the athlete in a similar stance but with his outstretched legs reversed. The O and P keys have a similar effect but instead power the movement of the calves. The O key applies a force that moves the left calf towards the back, and the right calf forwards until it is in line with the thigh. Again the P key is the reverse of this.



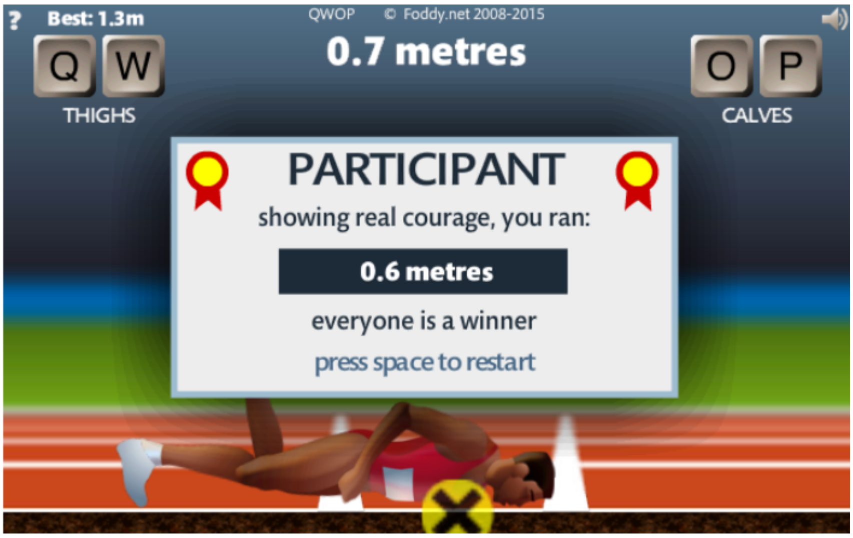
QWOP

- You are Qwop, our small nation's sole representative at the Olympic Games.
- Use the QWOP keys to move your legs.
- Ideally you will run 100 metres...
- ...but our training program was under-funded.
- Remember, it's not about whether you win or lose.

click to begin

Figure 1.1: Starting screen of QWOP

Once the player clicks on the screen they are free to press the keys to attempt to move forwards. This invariably leads to Figure 1.2 which shows the athlete fallen over. Below the athlete there is an X symbol. This shows where the athlete first touched the floor



PARTICIPANT

showing real courage, you ran:

0.6 metres

everyone is a winner

press space to restart

Figure 1.2: The athlete has fallen over so the player must restart.

with his arms or head. These section of his body are not allowed to touch the floor as doing so counts as falling over. After this the player can press space and start again. Most players attempt to keep the athlete upright and work out a key combination that will move the athlete in a motion as close to a normal gait as possible. Others however manoeuvre the runner into a stable position on the floor, and then press lots of keys in the hope that there is a net movement forward. Both these methods are entertaining and give some success. Though the objective of QWOP is simple, it has been notorious for being difficult to master ever since it went viral in 2010.

Chapter 2

Preparation

2.1 section 1

In this chapter I will discuss the preparation

alan physics model mathematical model of QWOP or how you model the runner

2.2 Mathematical model of the QWOP world

To create QWOP in a physics simulation it first helps to know how it is going to be modelled. So when preparing it was important to first design a mathematical model of the QWOP world that could later be implemented in a physics simulation.

2.2.1 Planar rigid body mechanics

The basis of this model is planar rigid body dynamics. As the floor that the athlete is standing on and the limbs of the athlete do not change size and do not need to deform, they can be modelled as rigid bodies. Save the floor, each of these rigid bodies represent one section of the athlete that has it's own position and rotation. For example the foot, upper leg and lower leg. Planar rigid body mechanics uses kinematics, and solves the equations of motion described by Newton's second law to determine the motion of the bodies in the system. Since this model can describe the acceleration, velocity and position of the individual rigid bodies over time, the development of the system as a whole under gravity can be simulated. With only basic planar rigid body dynamics, the limbs will just fall to the floor. So next I will describe some extensions to this model that will be made to emulate the behaviour of QWOP.

2.2.2 joints

Adjacent sections are connected by joints. For example the foot and lower leg are connected at the heel, and the lower leg and upper leg are connected at the knee. This is a constraint that the joints force the members to follow. For each joint-body connection, the joint stays in the same position relative to the frame of reference of the body - joints keep a consistent offset from each connected member. For example, the heel is always at the back of the foot, and the heel is always at the bottom of the lower leg. These two constraints work together in each joint to connect two limbs.

Another constraint the joints keep is that the angle the joints rotate to is limited. For example the knee allows the lower leg to rotate up until the point where the knee is locked out. In the same way the joints in the model limit the angle they can rotate to.

2.2.3 collisions

A final constraint the limbs follow is that they collide with the floor. Each rigid body has an area, and the areas of the limbs and the area of the floor cannot overlap, otherwise the athlete would fall through the floor. For this reason the model needs to take collisions between the limbs and floor into account. The collisions between the limbs do not however need to be taken into account, as one limb can go behind another in the 2D view the user can see. To model this each rigid body can have a rectangular collision box, specified by a width and height. The boxes for the limbs can then be stopped from entering the box for the floor.

The QWOP world so far amounts to a set of rigid bodies and a set of joints on which the physics rules of planar rigid body dynamics can be applied. The set of rigid bodies will contain 13 in total; firstly a foot, lower leg, upper leg, lower arm and upper arm for each side of the body, and also a torso, head and finally the floor. Each of these have a width and height that define their collision box, a starting position, and a mass. The values of these variables are such that the floor is immoveable and spans the bottom of the world, and the attributes of the rigid bodies match the attributes of the limbs of the QWOP athlete. The set of joints contains 11 in total; a left and right heel, knee, hip, shoulder, elbow, and a neck. Each of these specify which two rigid bodies they join together, the offset from each of those bodies that the joint must stay, and the maximum and minimum angles between which the joints can rotate. These connect the limbs together in the right order and keep them together with the physical properties of an athlete.

This model so far delivers an athlete with limbs that can rotate freely, move in Cartesian space and has correct joint constraints and collision checking. The last thing to add is the

forces rubber?

joint constraints made up of an offset from the first joint it is connected to and an offset for the second joint it is connected to. these work in two ways

todo: make more graphs with the ordering like what Alastair said and then work out the t test and write that up in the prep.

The first thing that I did to prepare was to get going with Phaser phaser is... objects sprites control flow/code sections pseudo code?

Next I learnt how to do these thingys stuff in course stats test open coding?

In this chapter, I have discussed the theory that had to be understood, and work that had to be done, before implementation could begin

Chapter 3

Implementation

todo: fix this This section explains how the mathematical model of QWOP (explained in the Prep section) was implemented along with the evaluation scaffolding used during the user study. Section 3.1 focuses on the implementation of QWOP (and perhaps explain what other subsections do). Section 3.2 (or 3.N if you've added more subsections to improve structure) explains the code written to produce and analyse the results of the user study.

This chapter describes the implementation of the project, moving on from preparation in previous chapters. The main implementation tasks in the project were writing my version of QWOP, and writing the set of software used to perform the user study and evaluate the project. These can be divided as follows:

- Section 3.1 Phaser Implementation of QWOP
  - Section 3.1.1 Modelling joints in physics
  - Section 3.1.2 Modelling the athlete
  - Section 3.1.3 Graphics
  - Section 3.1.4 Game flow
- Section 3.2 Evaluation software
  - Section 3.2.1 Web page hosting user study
  - Section 3.2.2 Server software to receive and store user study data
  - Section 3.2.3 Chrome extension for reading distance from QWOP

3.1 Phaser Implementation of QWOP

Phaser is a JavaScript framework designed for making games. I have described how it works in the preparation section, and will now break down how it was used to make the different sections of the game.

3.1.1 Modelling joints in physics

The first hurdle was to work out how to create a body in Phaser's physics engine. 3.1 shows my first prototype for modelling joints and limbs, each rod is a sprite, and the joints between them are fixed together by Phaser's 'revoluteContraint's. This allows them to swing freely.

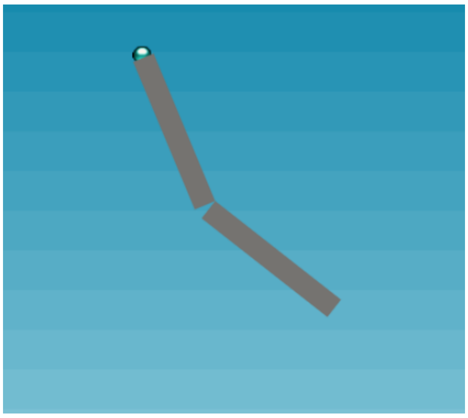


Figure 3.1: Swinging rods initial model of limbs

The next step was make the joints apply forces to each other. This would be used to not only move the limbs when the user presses the keys, but also provide forces so that the body would stay in a similar position when keys are not pressed.

The way I did this was to use Phaser's motors on the joints. These work by applying the force needed to move the joint at the specified speed, capped off at a max value. This way of powering joints could also be used easily to apply the static forces by setting the speed to 0.

3.1.2 Modelling the athlete

To model the athlete from these sprite joints, I specified the dimensions, mass, offset and starting rotation of each limb. Then forward kinematics is used on this data to get the starting points of the limbs. The limbs are then instantiated in these starting positions with the correct mass, dimensions and 'revoluteConstraint' joints. These joints are given joint limits so that each limb has a realistic range of motion, for example the knee joint doesn't let your lower leg rotate all the way round. The motors for these joints are then attached to the user input according to the control scheme. If a key is pressed that powers the limb, the motor is set to the limb power, otherwise it is set to 0 so that the static forces apply.

Each limb is given a group so that materials either part of the or group, that

3.1.3 Graphics

Next the graphics were added to the model so that it looked similar to the original QWOP. Each sprite was given an image taken from the original.

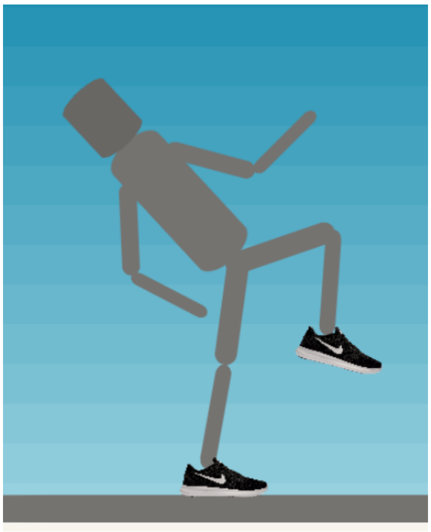


Figure 3.2: Model of athlete

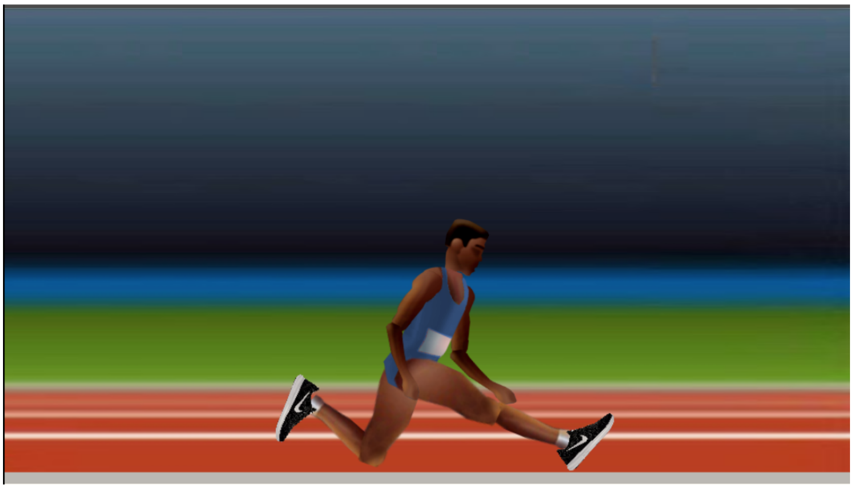


Figure 3.3: model with graphics added

3.1.4 Game flow

Finally all the other components of the game were added:

1. Collisions with the floor - Each limb that cannot touch the floor is put into a collision group. This group is used to check whether a collision with the floor occurs, and if so end the game.

2. Calculation of distance reached so far - a function of the x value of the athletes torso

3. Text showing the distance reached on the screen.

4. Recording the high score and displaying it on the screen

5. A User help box shown on game start-up

6. Checking whether the user has reached the end of the 100m and displaying the congratulations message if the player gets to 100m

3.2 Evaluation software

To evaluate the project, I will be analysing the results from a user study. This analysis will be happening in the evaluation section, however a large amount of work went into implementing the testing environment so I will describe that here.

3.2.1 Web page hosting user study

As I wanted to get lots of participants to take part in my user study, I decided to host it on a web page so that participants can take part concurrently. It was also useful to host it on a web page because I can host both versions of QWOP in iframes, embed Google forms as questionnaires, and record the actions of the user. The actions of the user such as key presses and distances reached over time can be analysed along with other data to evaluate the success of the game.

The main components of the web page are as follows:

1. A html page that guides the participant through the sections of the study

2. An embedded Google form that finds out demographics and the previous experience of each participant.

3. An A/B test. Each participant is randomly placed into one of the two groups, each of which play the two versions of QWOP in a different order. Each game is played for 5 minutes before the page moves onto the next section. While each game is played, the key presses and distances reached for each participant are recorded. The benefit of doing a randomised control trial is that selection bias is minimised. The splitting of participants into two groups means that the effect of playing one game before the other can be taken into consideration and not affect the analysis.

4. A final questionnaire that gives feedback on the two versions

3.2.2 Server software to receive and store user study data

I hosted the web page page on my SRCF web space. When the web page is recording data, it sends the data it has recorded to a PHP script at regular time intervals. This script records the data into separate files for each participant, so that all the data can be accessed in one place afterwards.

3.2.3 Chrome extension for reading distance from QWOP

todo: Start by telling me what the problem is. Is it that the implementation didn't expose its distance variable to something else? You see I don't understand – and it's your job to help me feel I do understand your program, and this needs you explaining in more detail :-)

Later on in the implementation I discovered that I couldn't access the distance that the player had reached for the original QWOP. It it impossible to access data from the game when hosting the original website inside an iframe, since all the game data is inside an anonymous function. Unfortunately When attempting to run the game by taking the html, JavaScript and other local resources that it accessed, and running them myself, the game's canvas goes orange and the game doesn't start. Since running an edited version of the original website did not work, I tried accessing the colour values of the game's canvas, then I could use OCR to read the distance from the screen itself. Accessing the pixel values from the canvas inside the iframe didn't work as QWOP has a OpenGL setting that made the GLBuffer unreadable.s In the end however the distance was made accessible by creating a chrome extension that took a screenshot of the page and placed it in a canvas. Then the data collection page takes the section of the screenshot that contains the distance text, inverts the colours so that it's black text on a white background, and places it into a smaller canvas. Then a lightweight JavaScript OCR script called ocrad.js is used to to read the text. With a little editing of the script's output, it gave an accurate value.

Chapter 4

Evaluation

4.1 success criteria?

4.2 User Study

To evaluate the project I have performed a user study. The user study is made up of these sections, the design on which has been discussed in the implementation:

1. A html page that guides the participant through the sections of the study
2. An embedded Google form that finds out demographics and the previous experience of each participant.
3. An A/B test. Each participant is randomly placed into one of the two groups, each of which play the two versions of QWOP in a different order. Each game is played for 5 minutes before the page moves onto the next section. While each game is played, the key presses and distances reached for each participant are recorded. The benefit of doing a randomised control trial is that selection bias is minimised. The splitting of participants into two groups means that the effect of playing one game before the other can be taken into consideration and not affect the analysis.
4. A final questionnaire that gives feedback on the two versions

4.3 Intro Questionnaire

The first questionnaire asked for the demographics of each participant. 30 participants took part in the user study. This is a good number because... Figure 4.1 shows the age and gender of the participants.

The next question asked was whether the user had played QWOP beforehand, Figure 4.2 shows the options that were available to choose from and the frequency that the answers were picked. Most of the user hadn't played QWOP before. that matters because..???

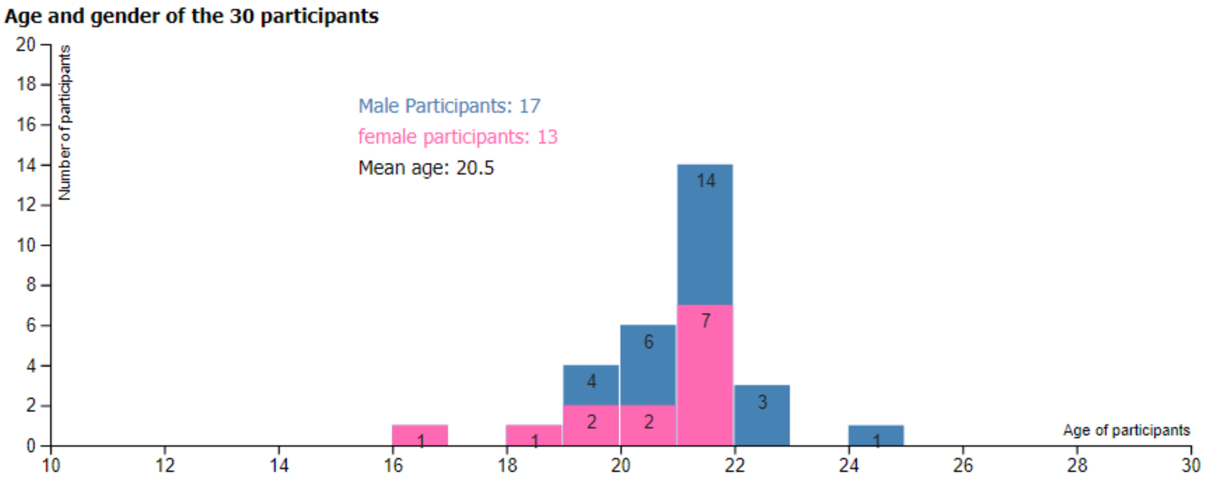


Figure 4.1: Participant demographics

Previous experience playing QWOP	Frequency of choice
Never played before	19
Less than 10 minutes	5
Between 10 and 30 minutes	3
Longer than 30 minutes	2
A long time over multiple sessions	1

Figure 4.2: Previous experience of participants

The third question asked for how long each participant plays video games during a typical week outside term time. Figure 4.2 shows the options that were available to choose from and the frequency that the answers were picked. This matters because..???

Time spent playing video games per week during a typical week outside of term time	Frequency of response
Less than an hour	22
Between 1 and 5 hours	2
Between 5 and 20 hours	4
More than 20 hours	2

Figure 4.3: Game playing experience of participants

4.4 Controlled experiment

After the introductory questionnaire, each participant is randomly placed into one of the two groups, each of which play the two versions of QWOP in a different order. Each game is played for 5 minutes before the page moves onto the next section. The participant has

5 minutes to get as far as they can. Whenever they fall over they start again and can have as many restarts as they want in the 5 minutes. While each participant plays both versions, the keys being pressed and the distance reached is being recorded. Here I shall use this data to analyse whether my version is faithful to the original, and how they compare.

Figure 4.4 shows some examples of the distance recorded over time. Notice how the participants attempts to go forward, reach a distances, and then restart. Participant 2 reaches 85m at around 250 seconds.

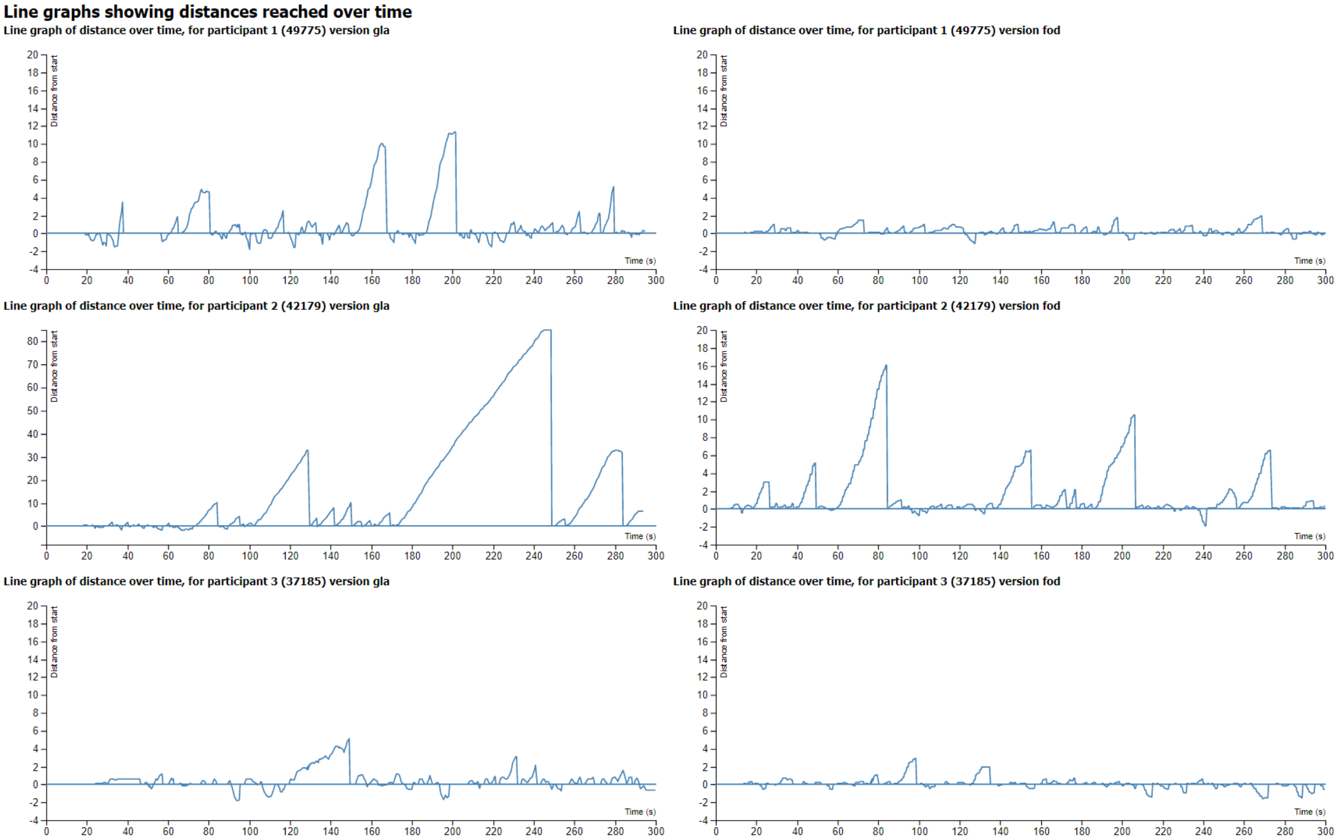


Figure 4.4: A selection of Distance over time graphs

To begin the analysis I shall compare the distances that the participants reach for each run; before they fall and restart or reach the end of the 5 minutes. Figure 4.5 shows a histogram of all of these restart distances over the 30 participants for my JavaScript remake, version gla. Figure 4.6 shows the same but for the original QWOP, version fod.

One striking fact is that the total number of restarts is so close. the number of runs for version gla is 924, and for version fod it is 925. My version is very faithful to the original in this regard and it shows that their gameplay share a property...

Version	Average time of run (s)
gla	9.74
fod	9.73

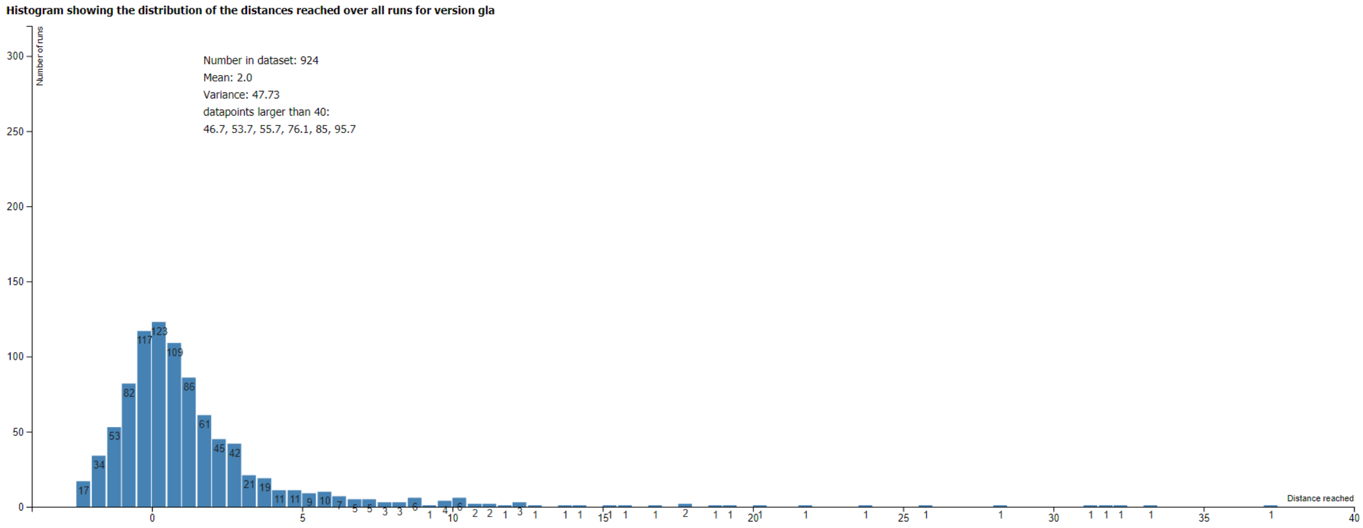


Figure 4.5: Restart distances for version gla

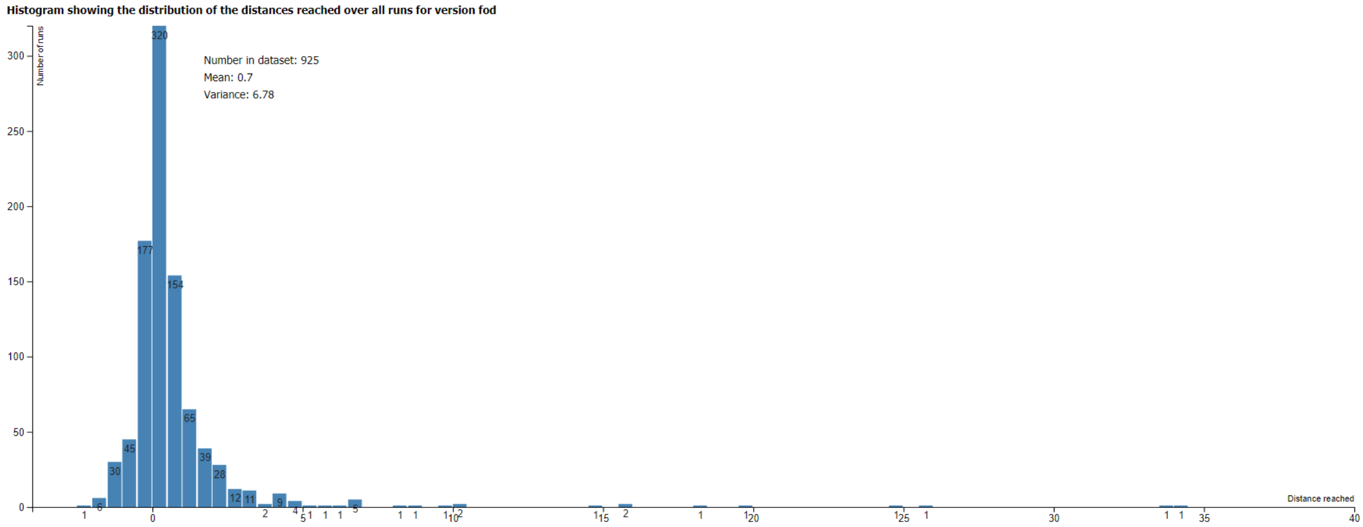


Figure 4.6: Restart distances for version fod

When comparing the restart distances of the two versions in Figure 4.5 and 4.6, the graphs share a similar shape. As they share the same area, 4.5 is a stretched version of 4.6

4.5 Second Questionnaire

The second questionnaire happens after the participants have played both versions of the game. Here are the questions, with the figures containing the data. and it contains the following questions aimed at comparing the two versions:

1. Which version of the game do you think you were more successful at? Figure 4.8
2. Which version felt like you had more control over the athlete? Figure 4.9
3. Why do you think this was?

4. Do you think playing the first version of the game improved your performance in the second version? Figure 4.7
5. Which version did you enjoy more? Figure 4.9
6. Why do you think this was?

Which version of the game do you think you were more successful at? could look at thought vs reality but doesn't seem that useful?

Which version felt like you had more control over the athlete? see Fig 4.9

why ? open coding?

first improved second?

Opinion whether playing the first	Frequency of response		
helped improve the second	Group A	Group B	Total
Definitely	1	5	6
Maybe	5	7	12
I'm not sure	3	0	3
Maybe not	3	0	3
Definitely not	3	3	6

Figure 4.7: Participant opinion of whether playing the first game helped improve their score for the second, by group.

Version	Frequency of response		
	Group A	Group B	Total
First played	8	5	13
Second played	7	10	17
Version gla	8	10	18
Version fod	7	5	12

Figure 4.8: Participant opinion of whether they were more successful at the first or second version.

As stated in the proposal the (find quote) aim is not to exactly copy qwop? the stats show that it is similar i made some changes to make it easier and more fun for players new curve more fun - backed up by hci stuff?

	Group A		Group B		Total over groups			
	First version (gla)	Second version (fod)	First version (fod)	Second version (gla)	gla	fod	First	Second
Which version felt like you had more control over the athlete?	7	8	4	11	18	12	11	19
Which version did you enjoy more?	9	6	5	10	19	11	14	16

Figure 4.9: Results of participant choice of version