Vishakh
81472764
Nicholas Urrea
23438530

# Formula 1 Prediction System

## Project Report

**ICS 175B**
**Fall 2003**
**Dan Frost**

# Table of Contents



Note: Four pages of our Cyc Knowledge Base are attached to this document

# 1. Introduction

This is a project report for the Formula 1 Prediction System, initially created for the ICS 175B project class. It lists the purpose, working and evaluation of the system and also a glimpse into its future. Formula 1 Racing (F1) itself is the world's most expensive sport and certainly the most glamorous motor sport of any kind. Ten teams fielded two drivers each and competed in sixteen races this year for the drivers' and constructors' world championships. There is tremendous interest worldwide in this sport and large sums of money are involved in racing and betting. Unsurprisingly, many powerful and complex predictive systems exist that allow teams and betters to predict outcomes of F1 races. Our system emulates them at a simpler level and suggests race outcomes based on a set of chosen setups, tracks and weather conditions.

An obsession shared by many F1 fans is thinking of hypothetical setups of F1 chassis, engines, tyres and drivers. For example, they often wonder how their favorite drivers would fare in other teams' cars. Assertions such as, "Fisichella would beat any driver if he were given a seat in a top car like Ferrari, Williams or McLaren" are common on WWW forums and are the subjects of intense discussions. By letting users switch around the chassis, engines, tyres and drivers in our system, we can let them create hybrid setups to shed some light on their arguments.

Our expert system allows users to choose the setups from the year 2003 and switch them around and then see what the race outcomes would be on a particular circuit. It relies on data culled from WWW sources and feedback from knowledgeable F1 fans (our "experts") to make these predictions. Since the outcomes generated by our system reflect the opinions of F1 fans to a large extent, it can make predictions of how they would place their bets were they to gamble on race outcomes. Users can use this information to place bets or simply to play around with setups and dabble in some amusing fantasy.

# 2. The System

## 2.1 Knowledge Engineering

Knowledge Engineering played a very important part in our project. We collected information about many variables and needed to ensure that it was accurate and unbiased. The knowledge we gathered was of two types:

1. **Facts:** Specific data about circuits, chassis, engines, drivers and tyres. These included:
    a. Chassis reliability
    b. Engine reliability
    c. Driver reliability
    d. Driver experience
    e. Circuit grips levels
    f. Circuit wear levels
    g. Circuit surface types
    h. Engine horsepower
    i. Circuit throttle levels

    This data was gathered from the internet, using sites such as f1db.com (http://www.f1db.com/) and the official Formula 1 site (http://www.formula1.com/).

2. **F1 Expert Opinions:** Since we needed opinions from F1 Experts, we turned to technically informed members of online F1 forums. We posed questions to them which asked them to rate chassis, engines, etc according to their subjective perceptions out of a maximum score of 100. We feel these ratings are much more accurate than they might seem. There are many variables involved in coming up with these ratings. If we were to do them mechanically, we would have to take all these into account and spend a long time refining the resultant scores. A rating provided by a human, on the other hand, is very easy to collect. It also includes all the possible variables. Our experts implicitly or explicitly consider recent performances, experience, long-term performance, type of car and many other factors before forming an opinion, for example, about a driver. Thus, human ratings are far more sophisticated than anything we feel we could have come up with in the same amount of time. Of course, even experts often hold biases, but these should have been ironed out since our sample size was large enough. An example of a public survey can be found at http://f1.motorsportforum.com/forums/showthread.php?s=&threadid=81066.

After collecting some preliminary expert opinion, we narrowed down the variables we wanted to include in our model. Next, we created our model by setting dependencies and weights. We showed it to the online experts and refined the weights. It should be stressed that this was quite an arduous process since the interaction with experts took up time and the data collection also required enormous patience and improvisation.

## 2.2 Ontology

The information we gathered was laid down in a Cyc knowledge base. Some pertinent pages of the KB have been attached for reference. Since the volume of the data we collected was enormous, the KB had become quite large (20 pages) even before our knowledge representation was done. As a result, we had to sacrifice some variables and make a compromise on the volume of data. There is a lot that can be said about the F1 teams of 2003 and it could easily occupy thousands of pages. For our purposes, around 30 pages seemed sufficient.

The ontology contains the information we collected in a modular way. We compartmentalized in a way such that any property (e.g. horsepower) is bound to a very specific object (e.g. engine). This makes it very easy to change the default bindings between chassis and other parts to the ones the users specify.

Some examples of the represented knowledge are now produced:

Chassis used in 2003:

```
;;ferrari
(cyc-create "F1FerrariF2003-GA" nil)
(cyc-assert '(#$isa #$F1FerrariF2003-GA #$F1Chassis)
#$UniversalVocabularyMt)

;;mcLaren
(cyc-create "F1McLarenMP4-17D" nil)
(cyc-assert '(#$isa #$F1McLarenMP4-17D #$F1Chassis)
#$UniversalVocabularyMt)

;;bmw williams
(cyc-create "F1WilliamsFW25" nil)
(cyc-assert '(#$isa #$F1WilliamsFW25 #$F1Chassis) #$UniversalVocabularyMt)

;;renault 23b
(cyc-create "F1RenaultR23B" nil)
(cyc-assert '(#$isa #$F1RenaultR23B #$F1Chassis) #$UniversalVocabularyMt)
.
.
.
```

Allocating engines to teams:

```
(cyc-assert '(#$hasEngine #$F1Ferrari #$F1Ferrari3000Type52)
#$UniversalVocabularyMt)
```

```
(cyc-assert '(#$hasEngine #$F1McLaren #$F1MercedesBenzF0110)
#$UniversalVocabularyMt)
(cyc-assert '(#$hasEngine #$F1Williams #$F1BMWP83)
#$UniversalVocabularyMt)
(cyc-assert '(#$hasEngine #$F1Renault #$F1RenaultRS23)
#$UniversalVocabularyMt)
(cyc-assert '(#$hasEngine #$F1Bar #$F1HondaRA003e)
#$UniversalVocabularyMt)
```

Some circuit information:

```
;;Hungaroring
(cyc-assert '(#$circuitSurface #$F1Hungaroring #$F1Dusty)
#$UniversalVocabularyMt)
(cyc-assert '(#$circuitGripLevel #$F1Hungaroring #$F1Medium)
#$UniversalVocabularyMt)
(cyc-assert '(#$circuitTyreWear #$F1Hungaroring #$F1High)
#$UniversalVocabularyMt)
(cyc-assert '(#$circuitMaxThrottle #$F1Hungaroring 50)
#$UniversalVocabularyMt)

;;Monza
(cyc-assert '(#$circuitSurface #$F1Monza #$F1Smooth)
#$UniversalVocabularyMt)
(cyc-assert '(#$circuitGripLevel #$F1Monza #$F1High)
#$UniversalVocabularyMt)
(cyc-assert '(#$circuitTyreWear #$F1Monza #$F1Medium)
#$UniversalVocabularyMt)
(cyc-assert '(#$circuitMaxThrottle #$F1Monza 65) #$UniversalVocabularyMt)
```

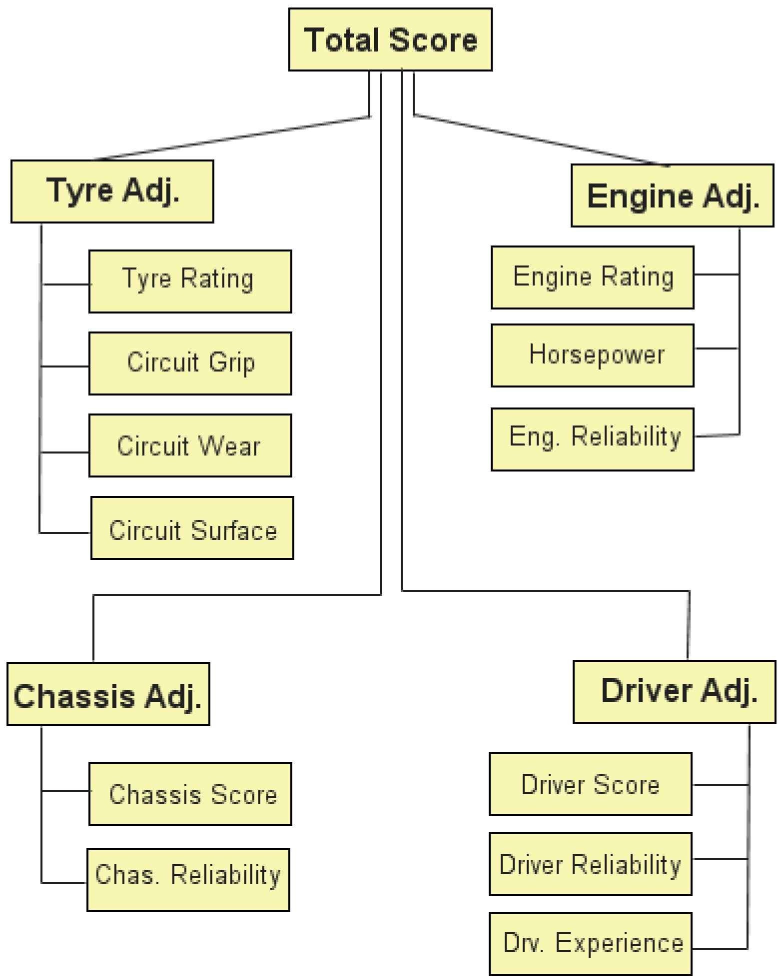
..finally, an inference rule that adjusts adjusted weights according to circuit throttle levels:

```
if(circuitMaxThrottle <= 49 )
{
shift = SHIFT_CONSTANT;
}
else if(circuitMaxThrottle <= 59)
{
	shift = 0;
}
else
{
	shift = 0-SHIFT_CONSTANT;
}
```

## 2.3 Our model

We broke up F1 cars into four ontological pieces- chassis, engines, drivers and tyres. Not only are they physical parts of the car, but they also provide convenient abstract containers for the many parts that F1 car have. The actual list of parts that play an important role in performance is quite large. This includes gearboxes, barge boards, size and inclination of wings, hydraulics, etc. It would be very difficult to handle all these parts separately, so we filed most of them under "chassis." The chassis also takes into account the aerodynamic properties of the car.

The model consists of factors and weight values. Factors are combined using weights to create larger factors. All factors are out of 100 points. Here is the model itself:




The factors we used are:

- **Total Score:** This gives the overall score of a chosen combination. The scores of the engine, chassis, driver and tyre are summed up using weights that depend on the circuit throttle and weather combinations.

- **Chassis Adjusted:** Total score for the chassis, taking into account constituent factors.
  - **Chassis Score:** Score assigned to this chassis after looking at ratings given by experts.
  - **Chassis Reliability:** The tendency of the chassis to remain intact through a race. There are many ways a chassis could fail, e.g. electronics failure, gearbox failure, etc.

- **Engine Adjusted:** Total score for the chassis, taking into account constituent factors.
  - **Engine Score:** Score assigned to this engine after looking at ratings given by experts.
  - **Engine Reliability:** The tendency of the engine to remain intact through a race and not blow up.
  - **Horsepower:** The horsepower of the engine. This was included because we wanted to stress the importance of horsepower since otherwise well-designed engines can have stifled horsepower. A powerful engine would certainly make a difference in how its car performs.

- **Driver Adjusted:** Total score for the driver, taking into account constituent factors.
  - **Driver Score:** Score assigned to this driver after looking at ratings given by experts.
  - **Driver Reliability:** The inverse of the tendency of a driver to make a mistake and run himself out of a race.
  - **Driver Experience:** Takes into account how much experience a driver has. More experienced drivers will, of course, have a slight edge in difficult situations.

- **Tyre Adjusted:** Total score for the tyre, taking into account constituent factors. Tyre scores are tricky since unlike chassis and engines, we don’t know exactly which tyre a team will run on a given course. We do know which manufacturer’s tyres the team uses, but not which type tyre (hard, soft, intermediate, and rain) will be in use. Hence, we deduce the probability of each type of tyre being used by looking at weather and circuit conditions. The tyre adjusted score is actually the expected value of the tyre score, derived using the probabilities generated from the four constituent factors.
  - **Tyre Score:** Score assigned to this tyre after looking at ratings given by experts. Tyres have variable scores, which depend on the weather conditions.
  - **Circuit Grip:** The amount of grip a circuit has. The best tyre from the manufacturer for the current grip condition is chosen and its Tyre Score is used here.

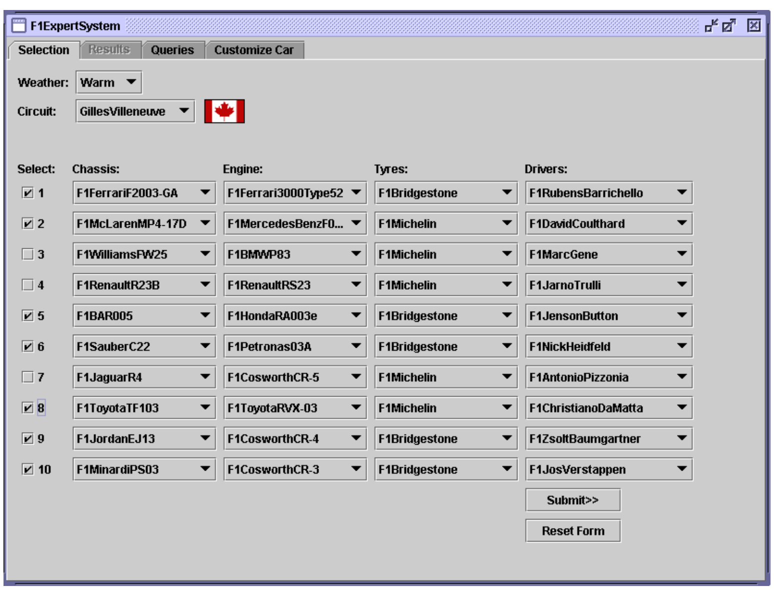
- **Circuit Wear:** The amount of grip a circuit has. The best tyre from the manufacturer for the current wear condition is chosen and its Tyre Score is used here.
- **Circuit Surface:** The type of surface a circuit has. The best tyre from the manufacturer for the current surface condition is chosen and its Tyre Score is used here.

The weights of the engine, chassis, driver and tyre adjusted scores going to the Total Score aren't shown. This is because they are variable. The maximum throttle and weather conditions are used to adjust the weights to perform more accurate predictions. For example, Hungaroring in Hungary is a very curvy circuit and thus the maximum throttle used there is very low. Teams with better chassis have an edge over ones that simply have high horsepower. As a result, the chassis weight is increased and engine weight is lowered. Conversely, Monza in Italy is much less curvy and powerful engines are crucial. For this circuit, the engine weight would be increased and chassis weight would be decreased. Finally, the tyres become crucial for cars when it is raining. Hence, the quality of rain tyres is considered and the tyre weight is increased.

The model is intentionally created in this manner so that it resembles a neural network. This is done so that the weights can be trained to produce better results. More about this can be found under "Future Work."

## 2.4 How it works

The system has a GUI written in Java:




The user selects a set of car setups. He can either stick with the actual 2003 setups or create hybrids. When he clicks "Submit," the model is used to generate scores for each setup. The scores assigned to each part are used to calculate higher factors using weights. The final adjusted weights are set according to the circuit throttle level and weather conditions. After the scores are generated, the output is displayed in the second tab of the UI in descending order of scores. In this process, Cyc is used the store information and Java is used for fetching and modifying it. The reasoning process we used involved mathematical calculations and we found Cyc unfit to perform then. As a result, the “reasoning” is done by our Java program.

### 2.5 Future Work

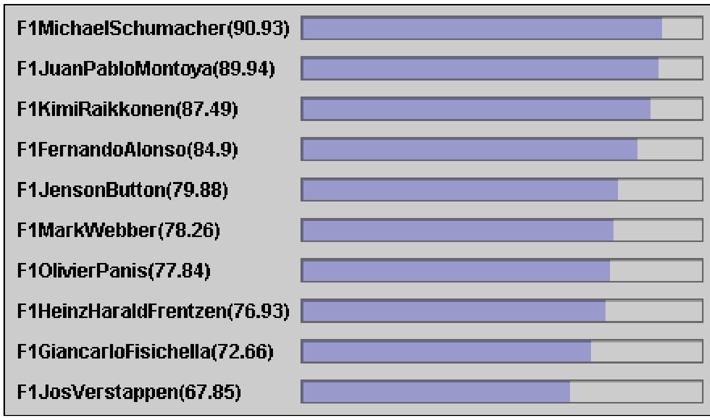
We plan to continue this project over the coming months. Already, we have put the results online on a MySQL database. A PHP script interfaces with the database. Interested F1 fans can access an HTML-based interface similar to our Java GUI. They choose setups, a circuit and weather conditions and then indicate what they think the results will be. Our system processes the inputs and shows the user the results. It looks at the discrepancies between outputs expected by users and the actual outputs. We plan to use a neural network to train the weights used in our model in this way. Our hope is that we will collect enough “training” output from users to make our system truly accurate. We then plan to offer this system to F1 fans as a handy tool for the purposes mentioned in the introduction. We will also add more complexity to our model in order to refine our system. A fanciful expectation is that our system will become accurate enough that we will be able to charge people small sums for predictions on upcoming races.

# 3. Evaluation

Our system is producing reliable results. We will now discuss general trends and then two special cases to see how our system fares.

## 3.1 General trend

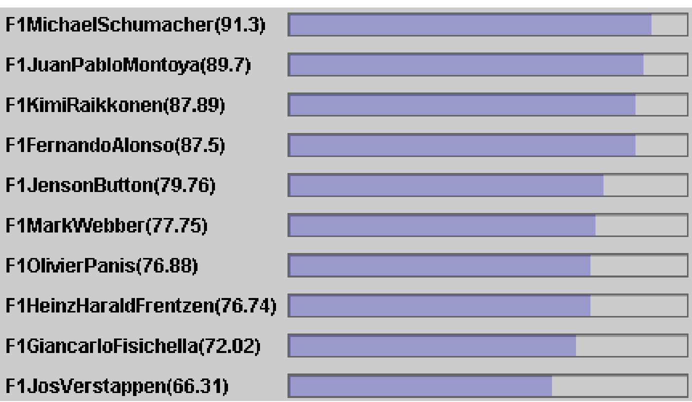
First, here are the results of our expert system produced for a random track compared to their world championship rankings at the end of the 2003 season:




1. Schumacher
2. Raikkonen
3. Montoya
6. Alonso
9. Button
10. Webber
11. Frentzen
12. Fischichella
15. Panis
21. Verstappen

There seems to be a close correspondence between the typical result on the left and the actual standings of the season on the right. Given how unpredictable F1 can be, this is quite a good result.

## 3.2 Special case 1



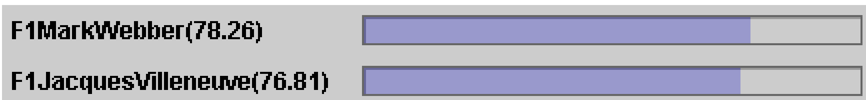

For our special test, we will consider the Hungaroring circuit in Hungary. This is a very curvy circuit and the chassis is paramount here. Engine powers don't matter here much as the maximum throttle reached on this circuit is very low. Renault has a fine chassis, but a weak engine. We should see the Renault driver, Alonso, scoring more points than in the previous example. Williams has a very powerful engine, but the opinion of our experts is that its chassis is not as good as Renault's. Thus the Williams driver Montoya's score should decrease slightly.

In the given figure, it can indeed be seen that the above is the case. Alonso’s score has gone up from 84.9 to 87.5 and Montoya’s down from 89.94 to 89.7.

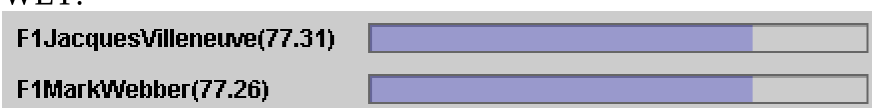
### 3.3 Special Case 2

Rain can always wreak havoc with the results, especially since in 2003 Bridgestone tyres far outperformed Michelin tyres in the wet. In the dry we should see the Michelin-runner Mark Webber outscoring the Bridgestone-runner Jacques Villeneuve. However, when the weather is set up rain afterwards, Villeneuve should edge out Webber. Here are the results:

DRY:

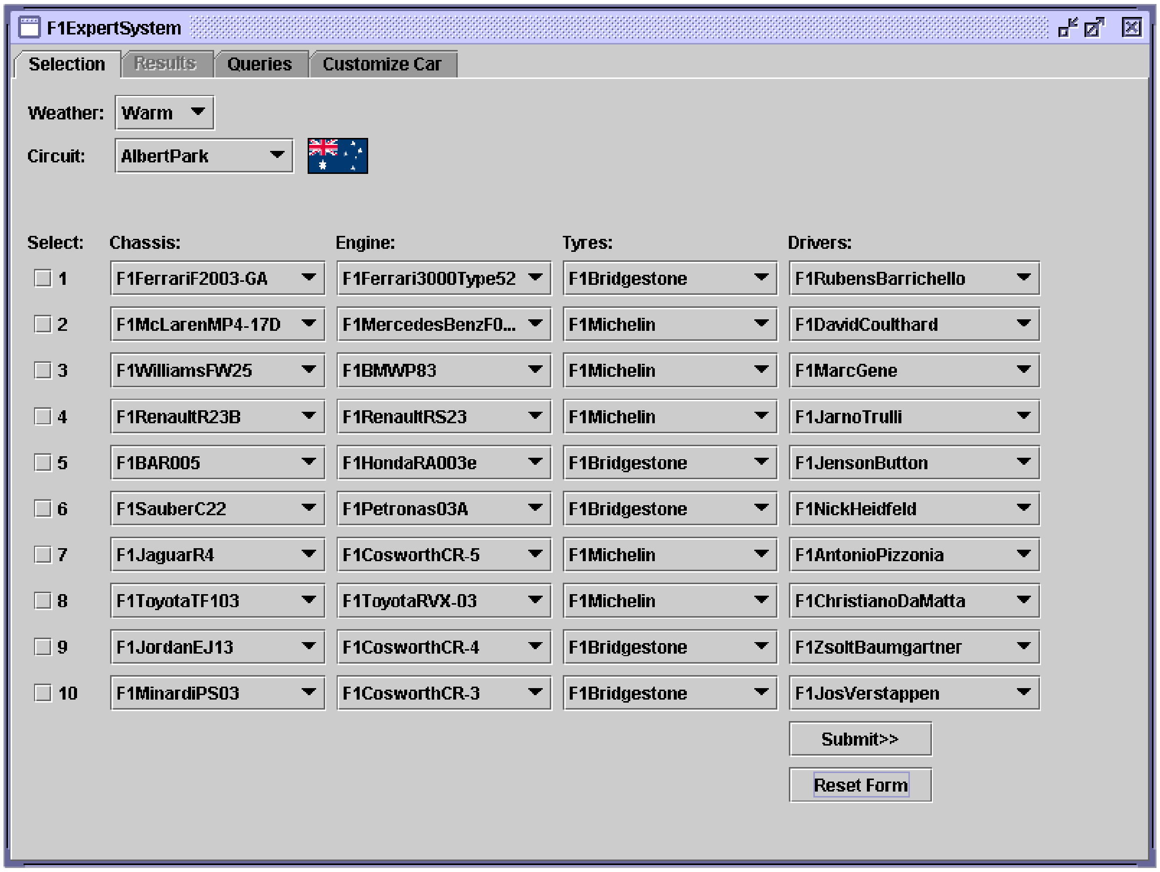



WET:

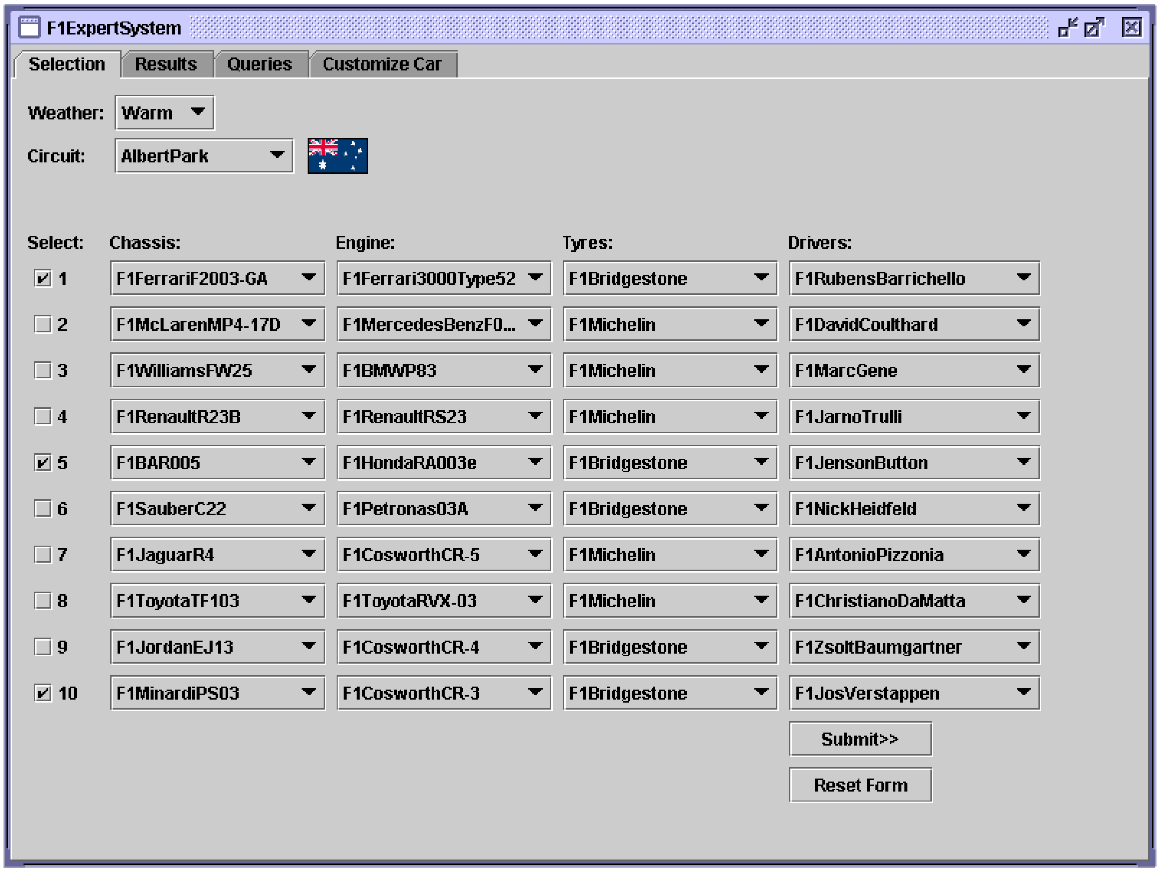



Indeed, the system has worked in this special case too. It seems like our expert system is doing its job satisfactorily. The system can be tested for all the circuits and should make reasonable predictions. Of course, chance plays a very large part in actual races, so the results work in a very “idealistic” setting. Overall, our system seems like a reasonable guide to go by.

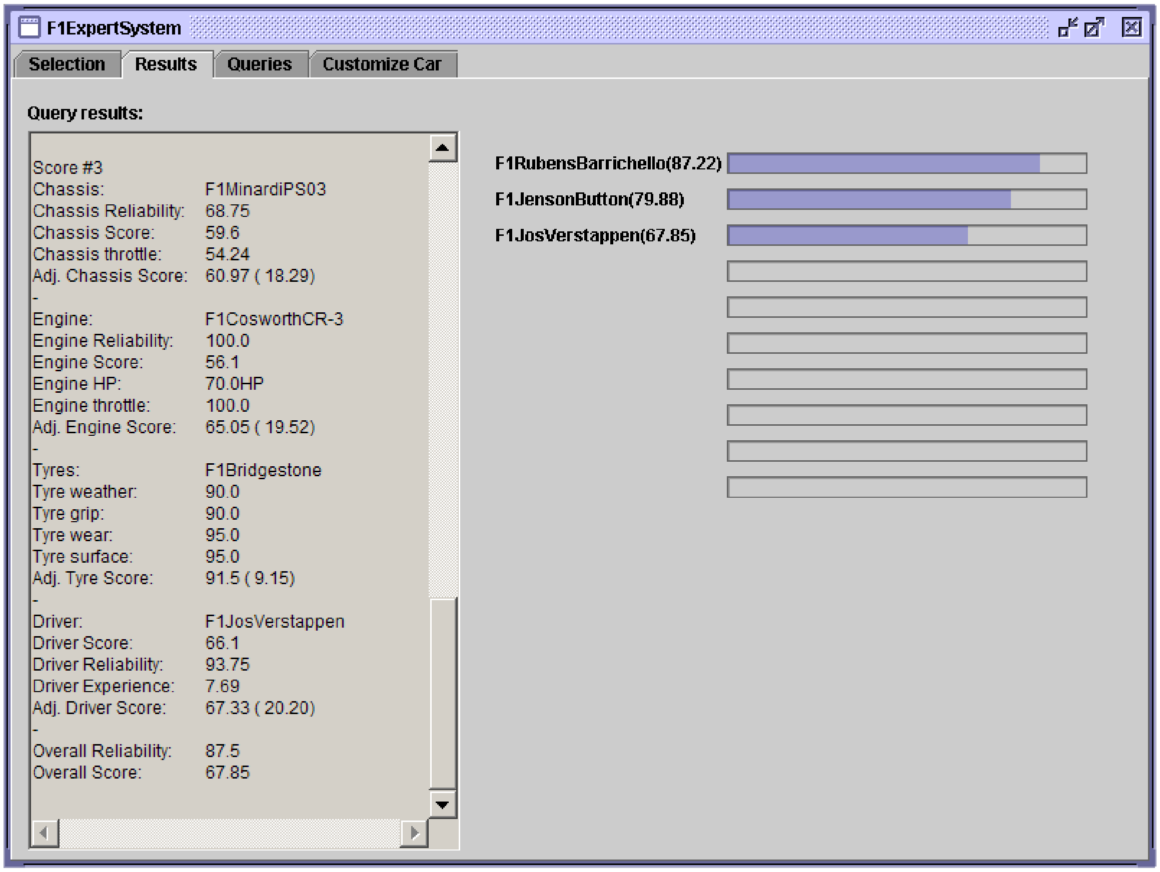
# 4. Transcript




This is the GUI users are presented with. They can choose any combination of setups. Each row is a setup and up to ten setups can be compared together.




The user had made her selections. She has chosen three setups and will now click on “Submit” to view the results.




The user now sees the results. The chosen setups’ scores are shown in order in the bars on the right. On the left, there is a text area that has a detailed breakup of the results, explaining the total scores.