# American Football Simulation

*0611156*

## Abstract

This paper introduces an innovative model for an imaginary offensive team in American football. The project attempts to simulate both the offensive play and the defensive play, with the aim of maximising the offensive yardage gain. The focus of this project is the interaction between the Receiver position (Offensive player) and his corresponding Corner Back (Defensive player), looking at the way in which Empirical Modelling can handle this. In addition, this project will allow the routes run by the Receiver in the offensive team to be changed, to see what impact this makes on the gaining of yardage/movement of the defence. To accomplish this route selection, a small number of preset game tactics will be loaded into the project from an up-to-date playbook, which is currently being used by a local American football team. As the user of this system will mainly be concerned with changing the tactics and layout of the offensive team, the defensive team will be automatically positioned relative to the changes in the offense. Therefore, as certain key offensive players (as mentioned previously) change their location on the field, the defensive counter-players will move accordingly (as happens in real American football games).

## 1. Introduction

American football is a difficult sport for beginners since the game play is rather different from any other type of sport. The fact that all the tactics and plays are determined before the game also makes it difficult to learn American football quickly. One of the main aims of this model is therefore to help beginners in American football see what would happen in a game from a bird's eye view, whilst also giving the user the ability to change several options (e.g. the Receivers' routes).

This model differs slightly from others using dependencies. For instance, both Turner (2000) and Stein (2005) focus on the relationships between only a few objects. This project, on the other hand, attempts to model the relationships and dependencies between 23 objects that each have a different role and objective within the game.

## 2. The game of American football

The game of American Football consists of 11 offensive players and 11 defensive players. The offense consists of Receivers (usually 3, depending on the play and the number of Tight Ends on field), 1 Quarterback, the Offensive line (consisting of the left and right Offensive Tackles, the left and right offensive guards and the centre), 1 full back and 1 halfback. The defense consists of (typically) 2 cornerbacks, the Defensive line (consisting of the left and right defensive ends the left and right defensive tackles), 3 linebackers, 1 free safety and 1 strong safety. The objective of the game is for the offense to carry the ball as far up the field as possible, with the ultimate goal of getting it into the Endzone of the opposing team – scoring a touchdown. There are a number of methods used to get the ball up the field, but this project focuses on the "passing play" in which the Quarterback throws the ball to one of the Receivers, who will aim to catch the ball and run as far as possible with it.

## 3. Model study

To model this game effectively there needed to be a clear goal, with smaller objectives which, when completed, helped add to the whole project.

## 3.1 Aims and objectives

The main aim of this project was to illustrate the basics of an American Football play (passing play) where the Quarterback would throw the ball to a Receiver who would then catch the ball and run towards the Endzone.

In order for this to be modelled as realistically as possibly, the 22 players and the ball would need to be modelled as individual objects. Some objects would need to have dependencies (see the *Results* section for more information) and some would remain static. Thus the main objectives were:

- to first move the players properly,
- getting the Quarterback to throw,
- getting the Receiver to catch and to run up the field, and
- for defenders to attempt to tackle the Receivers and Quarterback.

## 3.2 Results

This project was created entirely using DOSTE, running on tkeden version 2.10.

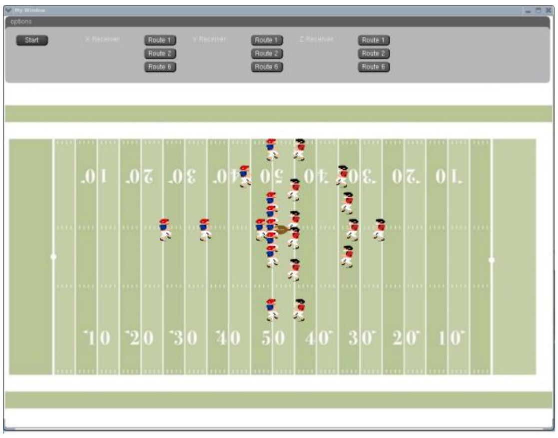

Figure 1 - The initial screen before the play starts

The figure above shows the initial starting screen before any buttons are pressed. There is the ball (being held by the Centre), and the 22 players (11 offense and 11 defense) with the offense wearing the red, blue and white, and the defense wearing the black, red and white.

There is an options pane which allows the user to select (out of 3 routes) which route each Receiver should run.

The Centre is initialised to holding the ball, as is with the real game.

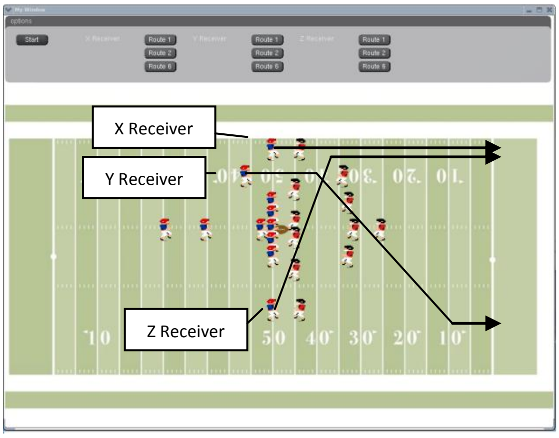


Figure 2 - Explanation of the routes

These routes were taken from an up-to-date playbook currently used by the University of Warwick's American Football team. X Receiver is showing the route 1, Y Receiver is showing the route 2 and the Z Receiver is showing the route 6. For this illustration the route selection is 1, 1, and 1. X, Y and Z all run route 1 (straight up field).

The offensive line also needs to play a part in this run, as they need to protect the Quarterback – giving him time to throw the ball. For this particular play, they will run the formation "Crash Pass", which means that the Offensive line will move to create a human wall, stopping anyone attempting to get through the middle. This leaves two defensive players – the left and right defensive ends, who will sneak past the side of this wall. This is why the halfback and the fullback will run forward to stop them. Now that the settings are finalised, the start button is pressed and the Centre will snap the ball back to the Quarterback.

To show how realistically this relates to American Football, in the huddle (before playing the real game of American Huddle) the Quarterback may vocalise the following play "Twins Left, Crash pass, 1, 6, 2. On 1, on 1, ready, ready, Break". This means that the "twins" (i.e. two Receivers) are on the left of the field, the Offensive line and Runningbacks perform Crash Pass protection, the X Receiver runs a 1, the Y Receiver runs a 6 and the Z Receiver runs a 2. Then “On 1” means that the QB will shout “Hut” once to start the game.

For this play then, this model is true to the game.

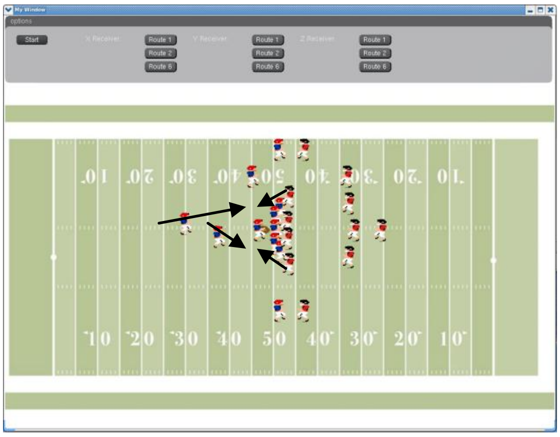

Figure 3 - Running backs protecting due to Crash Pass play

This offensive Crash Pass protection allows the Quarterback to have time to run back a bit and throw the ball to one of the Receivers...

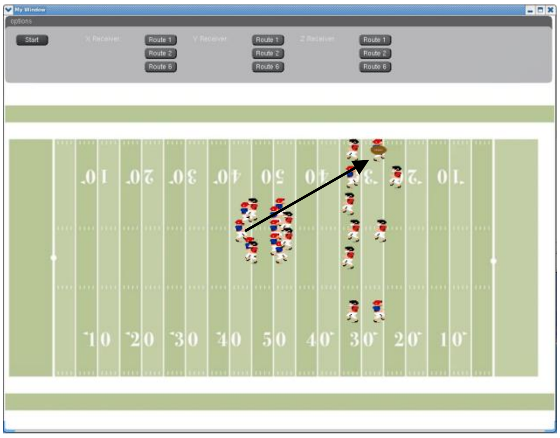

Figure 4 - Quarterback throws to Receiver

...who will catch the ball and run to score a touchdown:

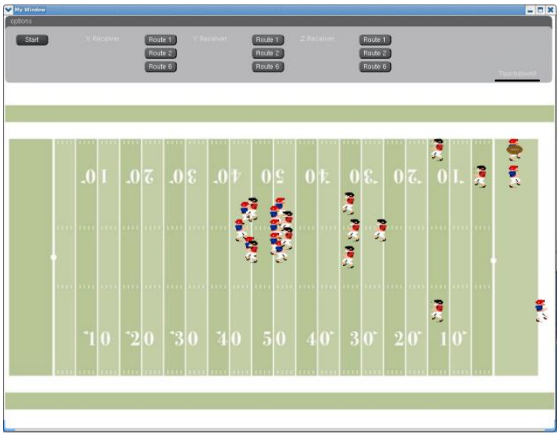

Figure 5 - Touchdown!

This play utilises a large number of different dependencies between the various objects: the initial line up of defense depends on the line up of the offense; the Cornerbacks positioning depends on the Receivers position – running backwards until the Receiver passes him, making him then chase the Receiver; the Runningbacks defense for the crash pass protection depend on the movement of the left and right defensive ends as they run towards the Quarterback; and the ball’s x and y position has an initial value (as in the real game) but then depends on who is holding the ball.

## 3.3 Further Work

There are some adjustments that could be made to the project that would not only allow it to be more useful (from a learning perspective) but also to make it more realistic to the actual game of American football.

One very interesting dynamic that would be difficult – but possible – to integrate into this project, would be the notion of intuition. When the Quarterback has the ball he needs to decide to which Receiver he should throw it. In the real game, the Quarterback would have to look around the field, and judge the defensive play and see which Receiver is open.

Also, currently the Linebackers and the Free Safety do not move, but are there merely to enhance the realism of what the play will look like.

In addition, the Receivers each have a speed variable associated with themselves, which is changeable through the tkeden input window. As the plays in American Football are all decided before the ball is snapped back to the Quarterback, it would not make too much sense to change the speed during a play.

However, a skill variable could be assigned to each player, which would determine (when the player is up against his respective defender) whether he succeeds in passing the defender or whether his is tackled. There could also be a fitness variable which could determine the chance of being injured after a tackle, in which the speed of the player may have to be changed mid-play.

This would give the feeling of being the coach, choosing the fastest, most skilled players for certain positions in order to see the outcomes.

Also, if this were to be completed, a yard counter could be displayed, which shows how many yards the player has made with the ball. A down marker could also be incorporated along with the rules of the game make it even more realistic.

Currently, due to time constraints, the Quarterback will only throw the ball to the X Receiver. The Quarterback knows which route he is running and will throw the ball into his path, as in real American Football, but this could be furthered to allow each Receiver to have the ball thrown to them.

Another main future extension is to allow for a whole play book to be loaded, which would also allow for the runningbacks to run with the ball, and also allow other protections from the offensive line.

# 4. The role of Empirical Modelling in this project

## 4.1 Why EM?

As this project is modelling a sport where there is both an offense and a defense, there are a lot of relations and dependencies, which is why Empirical Modelling was so helpful. Empirical Modelling simulates experiences through the use of artefacts as representations, which was very useful when creating players, as they were just a **union** of a standard base player, inheriting code from that. This allowed 22 players to be created on only 2 standard base players – a base defender and a base offensive player.

Also, although the model was entirely coded in DOSTE, EDEN statements could be used even in the tkeden input window. For example,

```
%eden
xReceiverspeed = 0.2;
#Set the X Receiver's Speed to be 0.2

%dasm
@screen xReceiver speed = {xReceiverspeed};
#Link the DOSTE variable to the EDEN value
```

Then in the tkeden input window:

```
xReceiverspeed = 0.0;
```

This will change the variable @screen xReceiver speed to 0, stopping the player immediately.

This is just one example of how different languages under Empirical Modelling work seamlessly.

## 4.2 Dependencies in DOSTE

Unlike most procedural languages, DOSTE allows for the constant updating of variables (using the **is { }** function), which allowed dependencies to be kept throughout the play. This allowed, for example, the Cornerbacks to run relative to the Receivers position, in a very true-to-the-game way.

Also, DOSTE allowed objects to have an initial starting position, but after the play started, their position on the screen depended entirely on what was happening in the play. For example, the ball has an initial position on the field, but when the play is started, it is held by the Centre, who passes it back to the Quarterback. The Quarterback will then run backwards with the ball, and throw it forward to a Receiver, who catches it and runs with it into the end zone. All the while the ball's position was only ever moved relative to other players. This is one huge advantage of choosing DOSTE over procedural languages, thanks to the ease at which it allows these relations.

### 4.2 Improvements to DOSTE

DOSTE is an impressive language with lots of positive points which helped massively when creating the model (as mentioned before) but it does, however, also have a few downsides.

One issue that was commonly encountered was the fact that there was no real code interpreter which would tell you if you made a coding error. Sometimes if there was a problem it would inform the user via the console (e.g. There is a missing bracket), but most of the time it would seem to ignore the problem but then fail to run correctly.

Also, if there was a coding error, it sometimes stopped the functionality of the correct code around the error, making the error seem to be in a place where there was none.

Another possible improvement, which would have helped simplify my code, would be the ability to assign two or more variables under one IF statement. For example, my model allows American football players to run diagonally on the field, yet to do this I needed two of the same IF statements, one for the x positioning and one for the y positioning.

## 5. Conclusion

In conclusion, I believe the model produced gives a realistic account of a passing play game – even the Offensive line is incorporated to add to the realism. I believe that if the extensions in the Future Work section were completed, this model could very well be used to teach all of the rules and plays of American football to beginners in a realistic way, especially if it allows the players' speed and skill to be changed.

I also believe that in the future, if the main flaws are overcome, Empirical Modelling should and will be chosen to program this sort of problem in the future (rather than standard procedural languages).

## 6. Acknowledgements

Thanks to Meurig Beynon for looking over my model, and for John Harvey and James McHugh for the help and ideas.

## 7. References

*"Five a side football"*
http://empublic.dcs.warwick.ac.uk/projects/footballTurner2000/

*"Road Traffic"*
http://empublic.dcs.warwick.ac.uk/projects/roadtrafficStein2005/

http://www.nfl.com/rulebook
*The Official NFL Rulebook*

http://www.gridiron3.com/
*Gridiron Game*

http://maddennfl.easports.com/home.action
*Official NFL Madden Game*

Football For Dummies (US Edition) (Paperback), by Howie Long (Author), John Czarnecki (Author), Publisher: John Wiley & Sons; 3rd Edition edition (22 Jun 2007)

http://www.dkimages.com/discover/Home/Sports-Games-Recreation/Ball-Games/American-Football/American-Football-84.html
*The American Football Field image*

http://www.clker.com/cliparts/e/8/1/c/1195445617639175942johnny_automatic_american_football.svg.hi.png
*The American Football Ball image*

http://www.videogameobsession.com/videogame/nesboxes/10-YardFight.jpg
*American Football player, copied for the offense and changed for the defense.*