RouteRender Language Specification

Tim Forth, Eric Gage (RouteRender)

0.1 Introduction

In American Football pre-designed plays are extremely important. If one player forgets the play or makes the wrong move, then the offense can be left vulnerable to defensive schemes. Therefore, it is imperative that players have the playbook memorized. For this to be accomplished, every play must be drawn out in an understandable way. However, this task is more daunting than it might seem. The average NFL playbook has 6,000 viable variations and almost infinite total variations. Therefore, we propose a programming language that would allow coaches to easily draw up plays for their playbook would bring efficiency to a currently inefficient task.

This problem requires a programming language because it simplifies the creation process. Every team runs a different offense and a different defense so there are numerous inputs that would be required to create a play (such as blocking schemes, routes, opposing defensive schemes, etc.). A programming language would allow coaches from different teams to create game-specific playbooks tailored to a specific team's defense.

0.2 Design Principles

In terms of design, a coach should be able to input very straightforward and simple inputs in order to create a play. It is also important that there is some way to save certain blocking schemes since there are a few schemes that are used frequently with small adjustments. Similarly, the language of football should be intertwined with the playbook language; the coach should be able to use keywords as inputs for schemes, routes, and formations (routes: go, basic, corner, dig. Schemes: power, counter, inside zone) as well as creating new ones. Simplicity is really the key factor in our design implementation. We want people with very little to no programming experience to be able to use this language efficiently.

0.3 Examples

1. dotnet run example1.rr

   Expected Output:

   ```
   <svg width="1500" height="1500" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://

       <defs>

         <marker id="arrow" markerWidth="10" markerHeight="10" refX="8" refY="3" orient="

           <path d="M0,0 L0,6 L9,3 z" fill="black" />

         </marker>

       </defs>
   <!--OLine-->

     <circle cx="530" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

     <circle cx="610" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

     <rect x="660" y="395" width="60" height="60" stroke="black" stroke-width="3" fill="n

     <circle cx="770" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

     <circle cx="850" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>
   ```

```
    <!--Three Four Box-->
        <text x="510" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">F<

        <text x="670" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">N<

        <text x="830" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">R<

        <text x="590" y="275" font-size="60" font-family="Arial, Helvetica, sans-serif">B<

        <text x="750" y="275" font-size="60" font-family="Arial, Helvetica, sans-serif">M<

        <text x="400" y="325" font-size="60" font-family="Arial, Helvetica, sans-serif">W<

        <text x="940" y="325" font-size="60" font-family="Arial, Helvetica, sans-serif">S<
    <!--Cover 2-->
        <text x="370" y="100" font-size="60" font-family="Arial, Helvetica, sans-serif">S<

        <text x="970" y="100" font-size="60" font-family="Arial, Helvetica, sans-serif">FS

        <text x="150" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">C<

        <text x="1190" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">C
    <!--Counter Scheme against 3-4 Defense-->

        <!--Labeling Keys-->

        <text x="417" y="275" font-size="30" stroke="red" font-family="Arial, Helvetica, s

        <text x="597" y="225" font-size="30" stroke="red" font-family="Arial, Helvetica, s

        <!--Pull for -1-->

        <line x1="530" y1="455" x2="545" y2="495" stroke="black" stroke-width="3"/>

        <line x1="545" y1="495" x2="790" y2="495" stroke="black" stroke-width="3"/>

        <line x1="790" y1="495" x2="790" y2="300" stroke="black" stroke-width="3"/>

        <line x1="770" y1="300" x2="810" y2="300" stroke="black" stroke-width="3"/>

        <!--Kickout-->

        <line x1="610" y1="455" x2="610" y2="485" stroke="black" stroke-width="3"/>

        <line x1="610" y1="485" x2="870" y2="480" stroke="black" stroke-width="3"/>

        <line x1="870" y1="460" x2="870" y2="500" stroke="black" stroke-width="3"/>

        <!--Down Block on Frank-->

        <line x1="690" y1="395" x2="560" y2="375" stroke="black" stroke-width="3"/>

        <line x1="555" y1="390" x2="565" y2="355" stroke="black" stroke-width="3"/>
```

```
     <!--Down Block on Nose-->

     <line x1="770" y1="395" x2="730" y2="375" stroke="black" stroke-width="3"/>

     <line x1="725" y1="390" x2="735" y2="355" stroke="black" stroke-width="3"/>

     <!--Block to BS Key-->

     <line x1="850" y1="395" x2="640" y2="290" stroke="black" stroke-width="3"/>

     <line x1="630" y1="305" x2="650" y2="275" stroke="black" stroke-width="3"/>
<circle cx="450" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

            <text x="437" y="440" font-size="45" font-family="Arial, Helvetica, sans-s
<!--Go Route-->

                <line x1="450" y1="395" x2="450" y2="100" stroke="black" stroke-width=".

                <text x="465" y="90" stroke="black" font-family="Arial, Helvetica, sans-
<circle cx="1080" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

            <text x="1067" y="440" font-size="45" font-family="Arial, Helvetica, sans-.
<!--Go Route-->

                <line x1="1080" y1="395" x2="1080" y2="100" stroke="black" stroke-width=

                <text x="1095" y="90" stroke="black" font-family="Arial, Helvetica, san.
<!--Under Formation-->

        <text x="668" y="510" font-size="60" font-family="Arial, Helvetica, sans-serif

        <circle cx="690" cy="660" r="30" stroke="black" stroke-width="3" fill="none"/>

        <text x="677" y="675" font-size="45" font-family="Arial, Helvetica, sans-serif

        <!--RB Block-->

        <line x1="690" y1="630" x2="580" y2="505" stroke="black" stroke-width="3"/>

        <line x1="560" y1="505" x2="600" y2="505" stroke="black" stroke-width="3"/>
</svg>
```

2. dotnet run example2.rr

Expected Output:

```
<svg width="1500" height="1500" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http:/

     <defs>

       <marker id="arrow" markerWidth="10" markerHeight="10" refX="8" refY="3" orient=".

         <path d="M0,0 L0,6 L9,3 z" fill="black" />

       </marker>
```

```
        </defs>
<!--OLine-->

  <circle cx="530" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

  <circle cx="610" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

  <rect x="660" y="395" width="60" height="60" stroke="black" stroke-width="3" fill="n

  <circle cx="770" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

  <circle cx="850" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>
<!--Three Four Box-->
    <text x="510" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">F<

    <text x="670" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">N<

    <text x="830" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">R<

    <text x="590" y="275" font-size="60" font-family="Arial, Helvetica, sans-serif">B<

    <text x="750" y="275" font-size="60" font-family="Arial, Helvetica, sans-serif">M<

    <text x="400" y="325" font-size="60" font-family="Arial, Helvetica, sans-serif">W<

    <text x="940" y="325" font-size="60" font-family="Arial, Helvetica, sans-serif">S<
<!--Cover 2-->
    <text x="370" y="100" font-size="60" font-family="Arial, Helvetica, sans-serif">SS

    <text x="970" y="100" font-size="60" font-family="Arial, Helvetica, sans-serif">FS

    <text x="150" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">C<

    <text x="1190" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">C
<circle cx="450" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

             <text x="437" y="440" font-size="45" font-family="Arial, Helvetica, sans-s
<!--In Route-->

                <line x1="450" y1="395" x2="450" y2="170" stroke="black" stroke-width="2

                <line x1="450" y1="170" x2="550" y2="170" stroke="black" stroke-width="2

                <text x="565" y="160" stroke="black" font-family="Arial, Helvetica, sans

             <text x="287" y="500" font-size="45" font-family="Arial, Helvetica, sans-s
<!--Post Route-->

                <line x1="300" y1="455" x2="300" y2="230" stroke="black" stroke-width="2

                <line x1="300" y1="230" x2="500" y2="130" stroke="black" stroke-width="2

                <text x="485" y="120" stroke="black" font-family="Arial, Helvetica, sans
```

```
    <circle cx="1080" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

              <text x="1067" y="440" font-size="45" font-family="Arial, Helvetica, sans-
<!--Out Route-->

                <line x1="1080" y1="395" x2="1080" y2="170" stroke="black" stroke-width=

                <line x1="1080" y1="170" x2="1180" y2="170" stroke="black" stroke-width=

                <text x="1195" y="160" stroke="black" font-family="Arial, Helvetica, san

        <text x="668" y="510" font-size="60" font-family="Arial, Helvetica, sans-serif

        <circle cx="690" cy="660" r="30" stroke="black" stroke-width="3" fill="none"/>

        <text x="677" y="675" font-size="45" font-family="Arial, Helvetica, sans-serif

        <!--RB Block-->

        <line x1="690" y1="630" x2 ="580" y2="505" stroke="black" stroke-width="3"/>

        <line x1="560" y1="505" x2="600" y2="505" stroke="black" stroke-width="3"/>
</svg>
```

3. dotnet run example3.rr

   Expected Output:

```
<svg width="1500" height="1500" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http:/

    <defs>

      <marker id="arrow" markerWidth="10" markerHeight="10" refX="8" refY="3" orient="

        <path d="M0,0 L0,6 L9,3 z" fill="black" />

      </marker>

    </defs>
<!--OLine-->

  <circle cx="530" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

  <circle cx="610" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

  <rect x="660" y="395" width="60" height="60" stroke="black" stroke-width="3" fill="n

  <circle cx="770" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

  <circle cx="850" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>
<!--Four Three Box-->
    <text x="480" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">F<

    <text x="590" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">E<
```

```
<text x="750" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">N<

    <text x="870" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">R<

    <text x="670" y="275" font-size="60" font-family="Arial, Helvetica, sans-serif">M<

    <text x="800" y="275" font-size="60" font-family="Arial, Helvetica, sans-serif">S<

    <text x="540" y="275" font-size="60" font-family="Arial, Helvetica, sans-serif">W<
<!--Cover 1-->

    <text x="460" y="250" font-size="60" font-family="Arial, Helvetica, sans-serif">SS

    <text x="590" y="100" font-size="60" font-family="Arial, Helvetica, sans-serif">FS

    <text x="150" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">C<

    <text x="1190" y="375" font-size="60" font-family="Arial, Helvetica, sans-serif">C
<!--Power Scheme against 4-3 Defense-->

<!--Labeling Keys-->

<text x="555" y="215" font-size="30" stroke="red" font-family="Arial, Helvetica, sans-

<!--Block F-->

<line x1="530" y1="395" x2="500" y2="375" stroke="black" stroke-width="3"/>

<line x1="490" y1="390" x2="510" y2="360" stroke="black" stroke-width="3"/>

<!--Pull for -1-->

<line x1="610" y1="455" x2="610" y2="495" stroke="black" stroke-width="3"/>

<line x1="610" y1="495" x2="790" y2="495" stroke="black" stroke-width="3"/>

<line x1="790" y1="495" x2="790" y2="300" stroke="black" stroke-width="3"/>

<line x1="770" y1="300" x2="810" y2="300" stroke="black" stroke-width="3"/>

<!--Down Block on Eagle-->

<line x1="690" y1="395" x2="640" y2="375" stroke="black" stroke-width="3"/>

<line x1="635" y1="390" x2="645" y2="360" stroke="black" stroke-width="3"/>

<!--B to K-->

<line x1="770" y1="395" x2="770" y2="380" stroke="black" stroke-width="3"/>

<line x1="750" y1="380" x2="790" y2="380" stroke="black" stroke-width="3"/>

<line x1="750" y1="355" x2="600" y2="275" stroke="black" stroke-dasharray="4" stroke-w
```

```
<line x1="590" y1="285" x2="610" y2="260" stroke="black" stroke-width="3"/>

<!--B to K-->

<line x1="850" y1="395" x2="800" y2="375" stroke="black" stroke-width="3"/>

<line x1="795" y1="390" x2="805" y2="360" stroke="black" stroke-width="3"/>

<circle cx="450" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

            <text x="437" y="440" font-size="45" font-family="Arial, Helvetica, sans-s
<!--Go Route-->

                <line x1="450" y1="395" x2="450" y2="100" stroke="black" stroke-width=".

                <text x="465" y="90" stroke="black" font-family="Arial, Helvetica, sans-
<circle cx="300" cy="485" r="30" stroke="black" stroke-width="3" fill="none"/>

            <text x="287" y="500" font-size="45" font-family="Arial, Helvetica, sans-s
<!--Post Route-->

                <line x1="300" y1="455" x2="300" y2="230" stroke="black" stroke-width="2

                <line x1="300" y1="230" x2="500" y2="130" stroke="black" stroke-width="2

                <text x="485" y="120" stroke="black" font-family="Arial, Helvetica, sans-
<circle cx="150" cy="485" r="30" stroke="black" stroke-width="3" fill="none"/>

            <text x="137" y="500" font-size="45" font-family="Arial, Helvetica, sans-s
<!--WR Screen-->
                <path d="M150,515 Q200,575 250,585" stroke="black" fill="none" marker-en

                <text x="265" y="575" stroke="black" font-family="Arial, Helvetica, sans-
<circle cx="1080" cy="425" r="30" stroke="black" stroke-width="3" fill="none"/>

            <text x="1067" y="440" font-size="45" font-family="Arial, Helvetica, sans-
<!--WR Screen-->
                <path d="M1080,455 Q1030,515 980,525" stroke="black" fill="none" marker-

                <text x="965" y="515" stroke="black" font-family="Arial, Helvetica, sans-
<circle cx="1230" cy="485" r="30" stroke="black" stroke-width="3" fill="none"/>

            <text x="1217" y="500" font-size="45" font-family="Arial, Helvetica, sans-
<!--Go Route-->

                <line x1="1230" y1="455" x2="1230" y2="100" stroke="black" stroke-width=

                <text x="1245" y="90" stroke="black" font-family="Arial, Helvetica, san
<!--Shotgun Formation-->

        <text x="668" y="635" font-size="60" font-family="Arial, Helvetica, sans-serif

        <circle cx="770" cy="615" r="30" stroke="black" stroke-width="3" fill="none"/>
```

```
        <text x="757" y="630" font-size="45" font-family="Arial, Helvetica, sans-serif

        <!--RB Block-->

        <line x1="770" y1="585" x2 ="770" y2="505" stroke="black" stroke-width="3"/>

        <line x1="750" y1="505" x2="790" y2="505" stroke="black" stroke-width="3"/>
    </svg>
```

0.4 Language Concepts

In order to create plays, users need a firm grasp of the core concepts: primitives and combining forms. Primitives encompass player positions, routes, offensive and defensive formations, schemes, and defensive coverages. Coaches put the primitives together in numerous different variations to create game plans against specific defenses, which they also implement. Understanding the primitives involves knowing how routes complement specific coverages or how run schemes exploit defensive weaknesses based on formations.

The combining forms, offensive and defensive plays, are the combination of these primitives. To create effective offensive plays, users must comprehend how different route concepts and offensive formations interact with defensive coverages and formations to exploit vulnerabilities. While understanding the primitives is more simple, such as a specific route against man or zone coverage, understanding the combining forms of the language are much more complex. It involves, for example, knowing how the combination of a 3x2 formation and 5 specific routes can pull apart a defense leaving some receiver open.

0.5 Formal Syntax

$\langle program \rangle ::= \langle play \rangle;$
$\langle play \rangle ::= \langle defense \rangle \langle scheme \rangle \langle routes \rangle \langle formation \rangle$
$\langle defense \rangle ::= (\langle box \rangle, \langle coverage \rangle)$
$\langle box \rangle ::= ThreeFour \mid FourThree$
$\langle ThreeFour \rangle ::= 34$
$\langle FourThree \rangle ::= 43$
$\langle coverage \rangle ::= \langle Cover1 \rangle \mid \langle Cover2 \rangle$
$\langle Cover1 \rangle ::= cover1$
$\langle Cover2 \rangle ::= cover2$
$\langle formation \rangle ::= (\langle unit \rangle, \langle receivers \rangle)$
$\langle unit \rangle ::= \langle Shotgun \rangle \mid \langle Under \rangle$
$\langle Shotgun \rangle ::= shotgun$
$\langle Under \rangle ::= under$
$\langle receivers \rangle ::= (axb); \ a + b < 6; \ a, \ b \ are \ integers \ greater \ than \ 0 \mid \_$
$\langle scheme \rangle ::= \langle Power \rangle \mid \langle Counter \rangle \mid \langle InsideZone \rangle \mid \langle OutsideZone \rangle \mid \langle Pass \rangle$
$\langle Power \rangle ::= power$
$\langle Counter \rangle ::= counter$
$\langle InsideZone \rangle ::= \textbf{inside zone} \mid insidezone \mid iz \mid insideZone \mid IZ$
$\langle OutsideZone \rangle ::= \textbf{outside zone} \mid outsidezone \mid oz \mid outsideZone \mid OZ$
$\langle Pass \rangle ::= pass$
$\langle routes \rangle ::= \langle route \rangle^+$ given length of routes $= a + b$ from receivers
$\langle route \rangle ::= (\langle player \rangle, \langle movement \rangle, \langle read \rangle)$
$\langle player \rangle ::= x \mid y \mid z \mid h \mid a$
$\langle movement \rangle ::= Go \mid Slant \mid Out \mid In \mid Post \mid Corner \mid Curl \mid Dig \mid Hitch \mid Comeback \mid Block \mid Fade \mid Screen \mid Run$
$\langle Go \rangle ::= go$
$\langle Slant \rangle ::= slant$
$\langle Out \rangle ::= out$
$\langle In \rangle ::= in$

$\langle Post \rangle ::= post$
$\langle Corner \rangle ::= corner$
$\langle Curl \rangle ::= curl$
$\langle Dig \rangle ::= dig$
$\langle Hitch \rangle ::= hitch$
$\langle Comeback \rangle ::= comeback$
$\langle Block \rangle ::= block$
$\langle Fade \rangle ::= fade$
$\langle Screen \rangle ::= screen$
$\langle read \rangle ::= First \mid Second \mid Third \mid Fourth \mid Fifth$
$\langle First \rangle ::= 1$
$\langle Second \rangle ::= 2$
$\langle Third \rangle ::= 3$
$\langle Fourth \rangle ::= 4$
$\langle Fifth \rangle ::= 5$

0.6   Semantics

0.7   Remaining Work

There are always more schemes and types of defenses and plays to implement. If we had more time that would be the first place we would start. We were unable to implement this simply because of the time constraint. Other than that, functions to move receivers into different alignments rather than just the standard alignment we currently have would be a nice addition. We were unable to implement this because of how much effort and time it took just to get the receivers working the way they are.

Syntax	Abstract Syntax	Type	Meaning
43	Box of string	string	tells evaluator what the front 7 looks like i.e. how it should be represented in the svg file; stored as an F# primitive string data type
cover1	Coverage of string	string	tells evaluator what the coverage of the defense is i.e. how it should be represented in the svg file; stored as an F# primitive string data type
(43, cover1)	Defense of Box * Coverage	string * string	two-tuple consisting of a box and coverage that tells eval how to draw defense
power	Scheme of string	string	tells evaluator what scheme the offense should be in
1	Read of int	int	tells a Quarterback what read a receiver is
x	Player of char	char	identifier for which player is running a route
post	Movement of string	string	identifies what movement a player is doing
(x, post, 1)	Route of Player * Movement * Read	char * string * int	player runs the movement specified with the read specified for the Quarterback
[(x, post, 1), (y, corner, 2), (z, dig, 3)]	Routes of Route list	List⟨ Route ⟩	list of Routes
iformation	Unit of string	string	describes the unit of the offense
(2x1)	Receivers of int * int	int * int	describes the alignment of the receivers
(iformation, (2x1))	Formation of Unit * Receivers	string * (int * int)	describes a full formation for the offense including the scheme and where the receivers are
(43, cover1)power[(x, post, 1), (y, corner, 2), (z, dig, 3)](iformation, (2x1));	Play of Defense * Formation * Scheme * Routes	(string * string) * (string * (int * int)) * string * List ⟨ Route ⟩	describes a full play with all needed information for the svg evaluator to draw it
;	n/a	char	tells the parser that a play is done

Table 1: Semantics of data types.