**Kick Returners** - The kick return thing is screwy, I always test them out before finishing. What I've figured out, is that putting any WR in the last spot causes problems. DB2/DB1 seem to usually work in the third spot, putting starting corners in the last spot always works as well.

**SNES Tecmo Hacking (Public)**

**Player string tables**

String tables consist of pointers and string data (It's kinda complicated to explain).

The pointers are 2 bytes (and they are swapped). You will need to swap the bytes and add an offset to get to the first string. You will need to get the value of the next pointer, subtract the previous one to get the length of the string.

In the SNES ROM, the name pointers start at location 0x178038 (I'll use the original rom in this example).

First pointer = 3A87, swap bytes, data = 873A

Second pointer = 4387 swap bytes = 8743

Length of first string = 8743 - 873A = 9 bytes.

String data starts at 0x178730 (12jimKELLY) (sort of, player number is 1 bye raw hex)

Calculate the offset:

offset + 0x873A = 0x178730

offset = 0x178730 - 0x873A

offset = 16fff6

Player attributes possibly at A311 if I remember correctly...

**Ratings**

NES:

RS = Start Speed

RP = Acceleration to MS

SNES:

RS = Acceleration to MS

RP = Start Speed

Someone from Knobbe:

It's often viewed that SNES for some reason reversed them, and that the NES way is the right interpretation. Therefore TSBTool probably labeled them differently so your always putting Start Speed for RS and Acceleration for RP, then reverses it on the SNES version of the game to accomodate for the switch. I believe TSBM2000 does this, I don't remember, it's been months since I've looked at it.

Soby:

Linebackers. Slowing them down to 44/44 is a huge deal... maybe you should really consider 50/38 instead of 50/44 for studs. RP (start speed) is the biggest deal for the pass rush anyway, not RS (accel). 44/44 will extremely underpower LBs while not really changing the sacking problem that you guys are seeing.

MRNFL:

The kick return thing is screwy, I always test them out before finishing. What I've figured out, is that putting any WR in the last spot causes problems. DB2/DB1 seem to usually work in the third spot, putting starting corners in the last spot always works as well.

Starting RBs generally shouldn't have an RS higher than 44 if you are also giving them high MS. High RS is for FBs or for a guy in the Okoye mold like Jacobs.

Madden attributes explained, if needed:

(OVR) Overall - The overall rating for the player. The higher the better!

(MOR) Morale -

(SPD) Speed - How fast the player can run. This attribute is associated with the Speed and Speed QB weapons.

(ACC) Acceleration - How fast a player reaches full speed. Great for sprinting through open holes in your line.

(STR) Strength - The player's strength, which affects his ability to break tackles.

(AGI) Agility - The player's agility, enhancing his ability to switch directions.

(AWR) Awareness - A player's ability to react and adjust. High awareness generally comes with more experience. This attribute is associated with the Smart QB weapon.

(CTH) Catching - How well the player can catch. A higher rating means fewer drops. This attribute is associated with the Hands weapon.

(CAR) Carrying - How well the player holds onto the football. A higher rating means the less chance of a fumble.

(THP) Throw Power - How far a player can throw the ball. This attribute is associated with the Cannon Arm QB weapon.

(THA) Throw Accuracy - How accurate a player throws. This attribute is associated with the Accurate QB weapon.

(KPW) Kick Power - A kicker's power. Higher power means longer kicks. This attribute is associated with the Big Foot Kicker weapon.

(KAC) Kick Accuracy - A kicker's accuracy. This attribute is associated with the Accurate Kicker weapon.

(RBK) Run Block - How well a player run blocks. This attribute is associated with the Crushing Run Blocker weapon.

(PBK) Pass Block - How well a player pass blocks. This attribute is associated with the Pass Blocker weapon.

(TAK) Tackle - A player's ability to tackle. This attribute is associated with the Brick Wall Defender weapon.

(JMP) Jumping - The player's ability to jump, such as for grabbing high passes.

(RET) Return - Skill of the player as a punt or kick returner.

(INJ) Injury - The likelihood of an injury. The higher the rating, the less likely a player will get injured.

(STA) Stamina - The player's stamina level. The higher the rating, the more the player can be in the game before he becomes fatigued.

(TGH) Toughness - A players ability to recover from injuries

(IMP) Importance - How important a player is to his team

(CEL) Celebration -

(TRK) Trucking - Player's ability to run over a defender attempting a tackle. Think of it as the offense's hit stick. This attribute is associated with the Power Back weapon.

(ELU) Elusiveness - Likely the ability to escape tackles (or perhaps sacks). This is a key statistic for running backs, receivers, and quarterbacks. This attribute is associated with the Elusive Back weapon.

(BCV) Ball Carrier Vision - Ball carrying vision: ability to spot the open field. This is a key statistic for running backs.

(SFA) Stiff Arm - Player's ability with the stiff arm maneuver. This attribute is associated with the Stiff Arm Ball Carrier weapon.

(SPM) Spin Move - Player's ability with the spin maneuver.

(JKM) Juke Move - Player's ability with the juke move.

(IBK) Impact Blocking - Would seem to be the blockers ability to make a leveling "impact" \*\*\*\*block.

(RBS) Run Block Strength - How well the player run blocks.

(RBF) Run Block Footwork - How well the player maintains footing during a run block.

(PBS) Pass Block Strength - How well the player pass blocks.

(PBF) Pass Block Footwork - How well the player maintains footing during a pass block.

(PMV) Power Moves - Look for high power moves in defensive linemen. Power Moves is the ability to power through the block. This attribute is associated with the Power Move D-Lineman weapon.

(FMV) Finesse Moves - Similar to power moves but it's the ability to spin or swim around blocks. This attribute is associated with the Finesse Move D-Lineman weapon.

(BSH) Block Shedding - How well the player can shed a block.

(PUR) Pursuit - Skill in finding and pursuing the ball carrier--likely laterally. For instance: a linebacker moving down the line to cut off the ball carrier.

(PRC) Play Recognition -

(MCV) Man Coverage - Skill of the defender in man coverage. This attribute is associated with the Shutdown Corner weapon.

(ZCV) Zone Coverage - Skill of the defender in zone coverage.

(RST) Running StyIe-

(SPC) Spectacular Catch - A player's ability to make a spectacular catch. This attribute is associated with the Spectacular Catch Receiver weapon.

(CIT) Catch in Traffic - How well the player can catch the ball and maintain possession in tight coverage. This attribute is associated with the Possession Receiver weapon.

(RTE) Route Running Rating - How well the player runs the play's passing route. This attribute is associated with the Quick Receiver weapon.

(POW) Hit Power - The strength of the player's hit, which can determine the success of a tackle or the possibility of a fumble. This attribute is associated with the Big Hitter weapon.

(PRS) Press Rating - Ability of defender in press coverage. This attribute is associated with the Press Coverage Corner weapon.

(RLS) Release Rating - Ability of receiver to release from press coverage.

**Race Cutscene Hex**

With that, the data is in binary, every 4 players corresponds to one hex bit. I went through the Bills in the original, looking at their players, and got this if light skin = 0 and dark skin = 1

3D E0 CF D3

I typed that into a hex editor, and found that at x17EF. If my theory was right, putting a C in place of the first 3 would make Kelly and Reich dark, and Thomas and Davis light.

**Team city/name string tables**

Pointers start at x7000

Data starts at x703C

From MrNFL on team name lengths:

To my knowledge, the longest name total that fit was 87 Cleveland Browns. Putting a number in front of the city name will cause it not to read on top of team names, and the number will be added on the play select screen. (EX 87Browns) Certain lengths for mascot names would be too long too, I think if its longer than Buccaneers.

From jstout

City can be 15, Mascot can be 10, Combined is 20 (for NES, could be same for SNES)

**Rosters**

Return Players

x170CB0 x00 - x1F (convert to decimal for player position 0 - 31)

Starters

Buffalo starts at x170C90

0, 1, or 2 for player in position 1, 2, or 3

Pro-Bowl

AFC 170C00-170C44

NFC 170C48-170C8C

0200070208020002070205070c0607

Bills QB starts as "00" mine is "02" Griese.

Every other byte inb bold are the guys in order.

QB1

QB2

RB1

RB2

and so on.......

so "02" and "07" in bold are your QB's

"08", "00", "07" and "05" are your RB's

**Misc**

In-season quarter length location: 0xA0EE (x9EEE on headerless)

Possible locations for turning injuries on in pre-season (untested):

x1AB1E or x20525 are the most likelies, change the x80 to x02 (from jstout)

**Schedules**

|  |  |  |
| --- | --- | --- |
|  |  |  |

1991

15F066-15F1E1

1992

15F1E2-15F3BD

1993

15F3BE-15F5B5

Very easy to do the schedule through hex. I did all three in like 40 minutes or so. It's basically just data entry.

Buffalo is team 00

Colts team 01

Fins team 02

and so on........ in the order on the team select screen. ending with....

Saints as 1A

49ers as 1B

00

01

02

03

04

05

06

07

08

09

0A

0B

0C

0D

0E

0F

10

11

12

13

14

15

16

17

18

19

1A

1B

1C

**SRAM**

To write to SRAM, normally you'd need to update the checksum with every value you change.  To disable this check:

x807E in the rom has C9 00 00 D0 02

That location compares the Calculated Checksum to the SRAM Checksum and if not equal then jump to CLEAR SRAM

so all those bytes change to xEA

xEA = No Command

so it will not check and will branch as if the checksum is always right

so, change C9 00 00 D0 02 to EA EA EA EA EA  in the rom and it will bypass the SRAM check.

**Injuries & Conditions**

Preseason injuries

x1AB1E or x20525 are the most likelies, change the x80 to x02

Save state (ZST file)

Playbook: x114E Home Starters: x1152 Home Injuries: x1157 Home Conditions: x115B

Playbook: x122B Away Starters: x122F Away Injuries: x1234 Away Conditions: x1238

so the "home starters" is who the user set as the starter at the beginning of the game?

at the moment of the game.  So if you change during the game those numbers would change

if I look at x1152...

it's got 03 49

The starters are in nibbles.  0 = QB1, 1 = QB2, 2 = RB1, etc so 03 is QB1 and RB2

if the starter code is 03 that means that:

qb1 started at quarterback (0), and RB2 was the the starting running back (3)

and then 49 is next, so that means

the second running back is RB3 (4)

For a default, most teams should be mostly 02 36 7A

For conditions in SRAM:

it is the same spots as the save state. The SRAM and Save State will have the same exact order for everything

Each team is listed with stats, playbook, starters, injuries, conditions. The only difference is the save state just has the home and away for the current game

SRAM

(21 bytes total, 5 bytes for starters, 4 bytes for injuries, 4 bytes for conditions, 8 bytes for playbook)

Buffalo x221

Ind x309

League Injury Hack

(headerless rom)

x31CC6:

20 6D B3 C2 10 EA EA EA EA EA EA EA EA

That tells the game to jump to the code I wrote instead of writing 00 as the injuries

x3336D:

E2 30 A5 00 0A 0A AA BF 90 B3 86 97 07 C8 BF 91 B3 86 97 07 C8 BF 92 B3 86 97 07 C8 BF 93 B3 86 97 07 60

That is the new code that tells the game to read the bytes I want and save them to the teams

x33390 to x333FF:

Is the injury bytes to read.  Replace FFs with 00s unless you want to test it.

4 bytes per team

So if you make the first 4 all FF then Buffalo would be completely injured

xE623:

A9 00 EA

Tells the game to instead of reading the injury bytes to check for recovery that all bytes are 00 (as in not injured)

Byte 1: QB1, QB2, RB1, RB2 /

Byte 2: RB3, RB4, WR1, WR2 /

Byte 3: WR3, WR4, TE1, TE2 /

Byte 4: The other sub players (I forget the order)

In binary a byte of 00 = %00,00,00,00.  A byte of FF = %11,11,11,11.  Wherever a 11 is there is an injured player

**Uniforms**

Matchups

x1752 to x1755 (headerless rom) was Buffalo and Indianapolis directly followed. There was just the binary and the color areas are elsewhere (from jstout)

x00 x00 x00 x00 = When to Use Jersey 1 and 2 this is done bitwise (Buffalo to Atlanta then AFC to NFC with final 2 bits always being 0s) where a 0 = Jersey 1 and 1 = Jersey 2.

(x00000000 would use Jersey 1 vs every team and xFFFFFFFC would use Jersey 2 vs every team)

[Matchup spreadsheet](http://www.homeloaf.com/files/Uniform%20SpreadSheet.ods) - generates "SET" commands for TSB Tool (requires [Open Office](http://www.openoffice.org/))

**Hex**

Multipliers

(256 - xFF) \* -255 + xFC

(4:53:13 PM) Jlstout123: so 1 \* -255 + 252.  -255 + 252.  -3 Yards

(4:54:25 PM) Jlstout123: In my program it has 2 equations with an if statement that checks for positive or negative

(4:54:54 PM) Me: so if it's a negative number, you have to use that formula?

(4:55:15 PM) Jlstout123: Yeah

(4:55:44 PM) Me: how do I know when to use that?

(4:55:46 PM) Jlstout123: So check the multiplier for less than x80 then use the positive equation else use the negative equation

(4:56:02 PM) Me: x80 = 128

(4:56:43 PM) Jlstout123: where it turns positive to negative. x80 to xFF is negative and x00 to x7F is positive

function GetMultipliedValue($remainder,$multiplier)

{

  if( $multiplier < 128 )

    return $multiplier \* 256 + $remainder;

  else

    return (256-$multiplier) \* ($multiplier\*-1) + $remainder;

}

**Palettes**

Changing Colors

(Red) + (Green \* 32) + (Blue \* 1024)

RGB would be the number / 8

so white.  255, 255, 255

is 255/8 = 31

31 + 31\*32 + 31\*1024 = 32767

32767 = x7FFF

and then reverse it xFF7F

Small Helmets

158740 IND/MIA/CIN/CLE/TEN/RAI/SEA/PHX/DET/TB

158760 BUF/NE/KC/WAS/GB

158780 PIT/DEN/SD/DAL/GIA/CHI/MIN/ATL/RAMS

1587A0 JETS/PHI/NO/SF

Large Helmets

The first offset is the helmet, the 2nd is the logo and facemask. The letter in () is for teams that share palettes. Example: IND(A) and PHX (A) share helmet colors.

BUF 15AA60 159388

IND 159560(A) 1595A0

MIA 15ABA0 159600

NE 15AB60 1595C0

JETS 15AA00 159260(B)

CIN 1592E0 159320

CLE 15AA40 159300

HOU 15ABC0 1595E0

PIT 15AB00 159480

DEN 159420 159440

KC 159340 159380

RAI 15AB40 159540 (C)

SD 1593E0 159400

SEA 159520 159500

DAL 15AB20 159540 (C)

GIA 15AAE0 159460 (D)

PHI 159240 159260 (B)

PHX 159560 (A) 159580

WAS 15AA80 159360

CHI 15AAA0 159460 (D)

DET 15AB80 1594E0

GB 1593A0 1593C0

MIN 159200 159220

TB 15ABE0 159620

ATL 15AAC0 159460 (D)

RAMS 1594A0 1594C0

NO 159280 1592C0

SF 15AA20 1592A0

SNES colors are 2 hex values (FF7F is white, for example). Each team logo uses a 16 color string with the last 3 colors used for the facemask. By default, the first 2 values are 0000's, the first one is transparent & allows the helmet color to show though, the 2nd one is pure black.

Each team's helmet shell also uses a string of 16 values, starting with two 0000's.

Team name logo palettes 3D402-3D4DA (definitely 4 colors per team)

City name palette 2B532-2B60A (4 colors per team)

To edit the team name logos, the graphics are compressed and need to be viewed in 2BPP or game boy format in a tile editor.

Endzones

There are different palettes for the endzones depending on the weather:

TEAM CLEAR RAIN SNOW

BUF 1597E0 15A1E0 15A5E0

IND 159800 15A200 15A600

MIA 159820 15A220 15A620

NE 159840 15A240 15A640

JETS 159860 15A260 15A660

CIN 159880 15A280 15A680

CLE 1598A0 15A2A0 15A6A0

HOU 1598C0 15A2C0 15A6C0

PIT 1598E0 15A2E0 15A6E0

DEN 159900 15A300 15A700

KC 159920 15A320 15A720

RAI 159940 15A340 15A740

SD 159960 15A360 15A760

SEA 159980 15A380 15A780

DAL 1599A0 15A3A0 15A7A0

GIA 1599C0 15A3C0 15A7C0

PHI 1599E0 15A3E0 15A7E0

PHX 159A00 15A400 15A800

WAS 159A20 15A420 15A820

CHI 159A40 15A440 15A840

DET 159A60 15A460 15A860

GB 159A80 15A480 15A880

MIN 159AA0 15A4A0 15A8A0

TB 159AC0 15A4C0 15A8C0

ATL 159AE0 15A4E0 15A8E0

RAMS 159B00 15A500 15A900

NO 159B20 15A520 15A920

SF 159B40 15A540 15A940

Correction to the palette locations.....................

Bills 159800

15A200

15A600

Colts 159820

15A220

15A620

Fins 159840

15A240

15A640

\*\*\*\*\* Patriots \*\*\*\* 159B60

15A560

15A960

Introduction screens

1st screen (game start) starts at hex- x162D00

2nd screen (preseason, season, etc) - x162C00

Uniforms

Each team

16 bytes "white guys" 4th byte 9C 42 = skin color

next 16 bytes "black guys" 4th byte 90 21 = skin color

\*\*\*\*Primary\*\*\*\*

Team/white guys/black guys

BILLS/15FE40/15FE60

COLTS/15FE80/15FEA0

<><><><><><><><><>

49ERS/160500/160520

\*\*\*\*Secondary\*\*\*\*

Team/white guys/black guys

BILLS/160600/160620

COLTS/160640/160660

<><><><><><><><><>

49ERS/160CC0/160CE0

Team data screen location

x161FC0

Field locations (I think these are correct vs. Elway7's below)

Clear - x159B80/x159BA0

Rain - x159EFC/x159F1C

From Elway7

Normal - x161980/x1619A0

Rain - x161D00/161D20

Snow - x161D60/161D80

2 sets for each field. Right and left hand side, from what I see. The 11th and 14th bytes are the actual field colors for each field.

Playbook palette

From Elway7: x160F20

On my rom: x15911C

**Juice**

Look for the following sequence at hex key x2699a:

0001df10h: 00 01 00 00 00 01 02 01 01 01 01 02 01 02 02 01

0001df20h: 02 01 03 02 02 02 02 03 03 02 02 02 04 03 02 02

0001df30h: 02 04 04 02 02 02 05 04 02 02 03 05 05 02 02 03

0001df40h: 06 05 02 02 04 06 06 03 02 04 07 06 03 03 04 07

0001df50h: 07 03 03 05 08 07 03 03 05 08 08 03 03 05 09 08

0001df60h: 03 04 06 09 09

The numbers go in blocks of 5, paste the following over each block of 5 to increase difficulty:

03 04 06 09 09

That will make every week play like you're undefeated in week 17.

Hex Boost Order: Defensive Speed, Offensive Speed, Interception, Pass Control, Reception

Each level is decided by the team's Win/Loss Record with each win raising the level of juice. A team with 0 wins and 0 losses is the first set of numbers (also the preseason juice). A team with 1 win and 0 losses plays against the same juice as a team with 6 wins and 5 losses (the second set of numbers). Any team with 16 or more wins will play vs the final set of computer juice.

**Grappling hacks**

<http://www.knobbe.org/phpBB2/viewtopic.php?f=22&t=11628>

Jstouts:

At x18D83:

20 10 D0  JSR $D010 ; MAN VS CPU GRAPPLING

EA        NOP

D1 03     CMP ($03),Y

At x18DC3:

20 37 D0  JSR $D037 ; MAN VS MAN GRAPPLING

EA        NOP

D1 63     CMP ($63),Y

At x1D010:

MAN VS CPU GRAPPLING:

A0 1C     LDY #$1C

B1 63     LDA ($63),Y ; LOAD PLAYER HP

4A        LSR

4A        LSR

4A        LSR

C8        INY

18        CLC

71 63     ADC ($63),Y ; HP/8 + Presses

91 63     STA ($63),Y

20 0E C4  JSR $C40E ; LOAD CPU PLAYER

A0 1C     LDY #$1C

B1 03     LDA ($03),Y ; LOAD CPU HP

4A        LSR

4A        LSR

4A        LSR

C8        INY

91 03     STA ($03),Y ; HP/8

AD 02 02  LDA $0202 ; RANDOM NUMBER

29 0F     AND #$0F ; Make Random x0-F

18        CLC

71 03     ADC ($03),Y ; HP/8 + RANDOM

91 03     STA ($03),Y

; RETURN TO NORMAL

B1 63     LDA ($63),Y

60        RTS

MAN VS MAN GRAPPLING:

A0 1C     LDY #$1C

B1 63     LDA ($63),Y ; LOAD OFFENSE HP

4A        LSR

4A        LSR

4A        LSR

C8        INY

18        CLC

71 63     ADC ($63),Y ; HP/8 + Presses

91 63     STA ($63),Y

AD 02 02  LDA $0202 ; RANDOM NUMBER

29 03     AND #$03 ; Make Random x0-3

18        CLC

71 63     ADC ($63),Y ; HP/8 + Presses + RANDOM

91 63     STA ($63),Y

A0 1C     LDY #$1C

B1 03     LDA ($03),Y ; LOAD DEFENSE HP

4A        LSR

4A        LSR

4A        LSR

C8        INY

18        CLC

71 03     ADC ($03),Y ; HP/8 + Presses

91 03     STA ($03),Y

AD 00 02  LDA $0200 ; RANDOM NUMBER

29 03     AND #$03 ; Make Random x0-3

18        CLC

71 03     ADC ($03),Y ; HP/8 + Presses + RANDOM

91 03     STA ($03),Y

; RETURN TO NORMAL

B1 03     LDA ($03),Y

60        RTS

At x18DAC:

4C 68 D0  JMP $D068 ; MAN VS MAN POPCORNING

At x1D068:

MAN VS MAN POPCORNING:

  20 0E C4  JSR $C40E ; MAY NOT BE NEEDED

  A0 1C     LDY #$1C

  B1 63     LDA ($63),Y ; LOAD OFFENSE HP

  38        SEC

  F1 03     SBC ($03),Y ; OFFENSE HP - DEFENSE HP

  90 07     BCC :+

; IF OFFENSE HP > DEFENSE HP

  C9 20     CMP #$20

  B0 0D     BCS :+++

  4C 7F D0  JMP :++ ; OFFENSE HP < POPCORN so to Grapple

; IF DEFENSE HP > OFFENSE HP

: C9 E2     CMP #$E2

  90 09     BCC :+++

; NORMAL GRAPPLE

: 20 B0 C3  JSR $C3B0

  4C B1 8D  JMP $8DB1

; OFFENSE POPCORNS

: 4C CB 8D  JMP $8DCB

; DEFENSE POPCORNS

: 4C D8 8D  JMP $8DD8

Modified 0-7 scale (CPU drones get 7 presses + HP adjustment)

At x18D83:

20 10 D0  JSR $D010 ; MAN VS CPU GRAPPLING

EA        NOP

D1 03     CMP ($03),Y

At x18DC3:

20 34 D0  JSR $D034 ; MAN VS MAN GRAPPLING

EA        NOP

D1 63     CMP ($63),Y

At x1D010:

MAN VS CPU GRAPPLING:

A0 1C     LDY #$1C

B1 63     LDA ($63),Y ; LOAD PLAYER HP

4A        LSR

4A        LSR

4A        LSR

C8        INY

18        CLC

71 63     ADC ($63),Y ; HP/8 + Presses

91 63     STA ($63),Y

20 0E C4  JSR $C40E ; LOAD CPU PLAYER

A0 1C     LDY #$1C

B1 03     LDA ($03),Y ; LOAD CPU HP

4A        LSR

4A        LSR

4A        LSR

C8        INY

91 03     STA ($03),Y ; HP/8

A9 07     LDA #$07 ; SET CPU PRESSES

18        CLC

71 03     ADC ($03),Y ; HP/8 + Presses

91 03     STA ($03),Y

; RETURN TO NORMAL

B1 63     LDA ($63),Y

60        RTS

MAN VS MAN GRAPPLING:

A0 1C     LDY #$1C

B1 63     LDA ($63),Y ; LOAD OFFENSE HP

4A        LSR

4A        LSR

4A        LSR

C8        INY

18        CLC

71 63     ADC ($63),Y ; HP/8 + Presses

91 63     STA ($63),Y

A0 1C     LDY #$1C

B1 03     LDA ($03),Y ; LOAD DEFENSE HP

4A        LSR

4A        LSR

4A        LSR

C8        INY

18        CLC

71 03     ADC ($03),Y ; HP/8 + Presses

91 03     STA ($03),Y

; RETURN TO NORMAL

B1 03     LDA ($03),Y

60        RTS

At x18DAC:

4C 68 D0  JMP $D068 ; MAN VS MAN POPCORNING

At x1D068:

MAN VS MAN POPCORNING:

  20 0E C4  JSR $C40E ; MAY NOT BE NEEDED

  A0 1C     LDY #$1C

  B1 63     LDA ($63),Y ; LOAD OFFENSE HP

  38        SEC

  F1 03     SBC ($03),Y ; OFFENSE HP - DEFENSE HP

  90 07     BCC :+

; IF OFFENSE HP > DEFENSE HP

  C9 20     CMP #$20

  B0 0D     BCS :+++

  4C 7F D0  JMP :++ ; OFFENSE HP < POPCORN so to Grapple

; IF DEFENSE HP > OFFENSE HP

: C9 E2     CMP #$E2

  90 09     BCC :+++

; NORMAL GRAPPLE

: 20 B0 C3  JSR $C3B0

  4C B1 8D  JMP $8DB1

; OFFENSE POPCORNS

: 4C CB 8D  JMP $8DCB

; DEFENSE POPCORNS

: 4C D8 8D  JMP $8DD8

**Playbooks/Formations**

Play Formations

Runs 161460-16175E

Passes 16175F-161A55

00-Kickoff

01-Punt

02-XP/FG

03-XP/FG

04-Pro T (a)

05-Slot

06-Onesetback (a)

07-2 TE

08-Motion Down

09-Shifting Onesetback

0A-Oneback (as in Z-Cross)

0B-Offset I

0C-Run and Shoot

0D-Run and Shoot 3-Wing

0E-Shotgun (a)

0F-Shotgun (b)

10-Shotgun 3-Wing

11-Shotgun (c)

12-Redgun

13-No Back

14-Pro T (b)

15-Oneback (b)

Playbooks 170D2F

0-7 is the avaiable plays for each slot

8-F is a play from the next slot over

**Substitutions**

NORMAL:

QB1 = QB2

RB1 = RB3, RB4, WR3, WR4, TE2, RB1, RB2, WR1, WR2, TE1

RB2 = RB3, RB4, WR3, WR4, TE2, RB1, RB2, WR1, WR2, TE1

WR1 = WR3, WR4, TE2, RB3, RB4, RB1, RB2, WR1, WR2, TE1

WR2 = WR3, WR4, TE2, RB3, RB4, RB1, RB2, WR1, WR2, TE1

TE1 = TE2, RB3, RB4, WR3, WR4, RB1, RB2, WR1, WR2, TE1

KR = RB3, RB4, WR3, WR4, TE2, WR1, WR2, TE1, RB1, RB2

PR = RB3, RB4, WR3, WR4, TE2, WR1, WR2, TE1, RB1, RB2

3-WIDE and R&S:

RB2 and WRs = WR4, TE2, RB3, RB4, WR3, RB1, RB2, WR1, WR2, TE1

**Tecmo SNES**

Uniforms

Each team

16 bytes "white guys" 4th byte 9C 42 = skin color

next 16 bytes "black guys" 4th byte 90 21 = skin color

\*\*\*\*Primary\*\*\*\*

Team/white guys/black guys

BILLS/15FE40/15FE60

COLTS/15FE80/15FEA0

<><><><><><><><><>

49ERS/160500/160520

\*\*\*\*Secondary\*\*\*\*

Team/white guys/black guys

BILLS/160600/160620

COLTS/160640/160660

<><><><><><><><><>

49ERS/160CC0/160CE0

Team data hex

x161FC0

General

Player rating levels: 6, 13, 19, 25, 31, 38, 44, 50, 56, 63, 69, 75, 81

ny difference of I think 4 levels of HP between players and its an auto-toss

(7:57:17 PM) GoBlue2400: so like say you have 75 HP DL

(7:57:22 PM) GoBlue2400: and i have a 38 HP OL

(7:57:27 PM) GoBlue2400: as soon as the DL touches him he'll fly

(7:57:31 PM) GoBlue2400: and the DL will be running loose

**Introduction Palettes**

1st screen (game start) starts at hex- x162D00

2nd screen (preseason, season, etc) - x162C00

**Schedules**

1991

15F066-15F1E1

1992

15F1E2-15F3BD

1993

15F3BE-15F5B5

Very easy to do the schedule through hex. I did all three in like 40 minutes or so. It's basically just data entry.

Buffalo is team 00

Colts team 01

Fins team 02

and so on........ in the order on the team select screen. ending with....

Saints as 1A

49ers as 1B

00

01

02

03

04

05

06

07

08

09

0A

0B

0C

0D

0E

0F

10

11

12

13

14

15

16

17

18

19

1A

1B

1C

**Remaining Graphics**

Graphics changes:

Broncos

Vikings

Jets

Seahawks

Rams

Bengals

Eagles

Color changes:

Patriots

Packers green

Eagles green

Jets green

Patriots facemask

Errors:

Browns logo has "43" in it (endzone)

**Unknown**

Preseason Injury Flag

Editing name logos

Playbooks

Sim stats (manual)

Injury location in SRAM

**Injuries & Conditions**

Home Starters: x1152 Home Injuries: x1157 Home Conditions: x115B

Away Starters: x122F Away Injuries: x1234 Away Conditions: x1238

so the "home starters" is who the user set as the starter at the beginning of the game?

at the moment of the game.  So if you change during the game those numbers would change

if I look at x1152...

it's got 03 49

The starters are in nibbles.  0 = QB1, 1 = QB2, 2 = RB1, etc so 03 is QB1 and RB2

but if I switch RB3 in to the RB1 slot...how does that change the starter code

if the starter code is 03 that means that:

qb1 started, and RB2 was the the starting running back

and then 49 is next, so that means

the second running back is RB3

For a default, most teams should be mostly 02 36

For conditions in SRAM:

it is the same spots as the save state. The SRAM and Save State will have the same exact order for everything

(5:53:41 PM) Jlstout123: Yeah, so each team is listed with stats, playbook, starters, injuries, conditions. The only difference is the save state just has the home and away for the current game

Old way

home team is at 0x500 and away is at 0x605

read in 3 bytes for the 12 offensive players

convert to binary

so 0C 00 00

turns into 0000 1100 0000 0000 0000 0000

the numbers are then broken up into sets of 2

00 = not injured

01 = probable return

11 = questionable

10 = doubtful

01 10 11 = injured

and it goes in order from qb1-te2

going by the above set of data

it shows that RB1 is INJURED and is questionable for his return

SRAM

(21 bytes total, 5 bytes for starters, 4 bytes for injuries, 4 bytes for conditions, 8 bytes for playbook)

Buffalo x221

Ind x309

Injury SRAM - Buffalo x226

(2:52:11 PM) Jlstout123: IND should be x30E and down the line.  It is an obvious pattern because the playbooks and starters get listed right next to them

(2:52:20 PM) Jlstout123: 4 bytes

(2:52:48 PM) Jlstout123: The first byte has QB1, QB2, RB1, RB2

(2:53:13 PM) Jlstout123: where bit 11 = injured

(2:53:52 PM) Jlstout123: the checksum is x50 and x51 (maybe be longer)

(2:54:03 PM) Jlstout123: x50 is the low and x51 is the high

unless there are more bytes it is like this

(2:56:00 PM) Jlstout123: x00 and you change to xFF.  Then you take your checksum say 50 has x82 and 51 has xBF then take xBF82 + xFF and then write the answer back

(2:56:30 PM) TecmoTurd: oh, ok

(2:56:38 PM) Jlstout123: the change of number gets added or subtracted from the previous checksum

(2:56:47 PM) TecmoTurd: so there's one checksum

(2:56:50 PM) Jlstout123: yes

(2:56:56 PM) TecmoTurd: for the all the teams

(2:56:58 PM) TecmoTurd: not one each

(2:57:02 PM) Jlstout123: yeah

(2:57:26 PM) Jlstout123: It is possible x52 is also part but I wouldn't know until you had to change that much info

(2:57:28 PM) TecmoTurd: so for each byte you update, you update the checksum

x807E in the rom has C9 00 00 D0 02

That location compares the Calculated Checksum - SRAM Checksum and if not equal then jump to CLEAR SRAM

so all those bytes change to xEA

xEA = No Command

so it will not check and will branch as if the checksum is always right

so, C9 00 00 D0 02 changes to EA EA EA EA EA

**Blanking Helmets**

3:47:49 PM) TecmoTurd: hey, is there away to blank out helmet sin the nes version?

(3:48:32 PM) Jlstout123: Mini or large?  You can do both but it is different between them

(3:48:49 PM) TecmoTurd: well, I guess both, but start with mini

(3:49:58 PM) Jlstout123: The minis you just need to go to the designs and tell each one to use the blank tile

(3:50:21 PM) TecmoTurd: in hex you mean

(3:50:37 PM) TecmoTurd: just point it to any blank tile?

(3:51:23 PM) Jlstout123: yeah, 00 is blank so just make each 00 00 00 00 00

(3:53:21 PM) TecmoTurd: ok, i'll try that...

(3:53:24 PM) TecmoTurd: and for large helmets?

(3:54:06 PM) TecmoTurd: yay, that worked

(3:54:10 PM) Jlstout123: You have to point the pointers to a FF in the design.

(3:54:31 PM) TecmoTurd: point the pointers to a FF...I don't follow :)

(3:55:38 PM) Jlstout123: The design is like 90 3D 00 FF.  The pointer would be to 90 but you would change that to the FF.  Each team has a pointer to its code and a pointer to how to line up and where the code is.

(3:56:08 PM) TecmoTurd: oh, so just changing 90 to FF would blank it?

(3:56:12 PM) Jlstout123: yes

(4:02:54 PM) TecmoTurd: where do I find that "design"

(4:02:57 PM) TecmoTurd: i'm using this post:

(4:02:58 PM) TecmoTurd: <http://www.knobbe.org/phpBB2/viewtopic.php?t=2988>

(4:03:09 PM) TecmoTurd: so if I wanted to blank buffalo

(4:03:19 PM) TecmoTurd: I'd go to hex offset x1008C

(4:03:27 PM) TecmoTurd: and change the first number to FF?

(4:03:31 PM) TecmoTurd: er, first byte

(4:04:40 PM) Jlstout123: go to x106D0 and in that will be a pointer to the design.  Change the one pointer to FF and then for all the team pointers chang to use C0A6 like Buffalo

(4:04:59 PM) Jlstout123: I'd have to look to see the design location itself though

(4:05:10 PM) TecmoTurd: where did you get the 106D0?

(4:05:53 PM) Jlstout123: C0A6 is a pointer to a spot that does Buffalo's tileset, xy coordinates, and pointer to design

(4:08:33 PM) TecmoTurd: oh, so this just blanks out the logo

(4:08:39 PM) TecmoTurd: it doesn't remove the helmet

(4:09:57 PM) Jlstout123: lol, if you wanted to I guess you could make the entire helmet the same color as the background

(4:10:41 PM) TecmoTurd: yeah...if you hadn't noticed i'm too lazy to edit all the helmets, so I thought I'd just blank them all out

(4:11:05 PM) TecmoTurd: not a big deal really cause those screens usually just get skipped anyways

**SRAM Stats**

Stats for BUFQB1 starts at decimal offset 334

**Palettes**

Small helmet palettes

158740 IND/MIA/CIN/CLE/TEN/RAI/SEA/PHX/DET/TB

158760 BUF/NE/KC/WAS/GB

158780 PIT/DEN/SD/DAL/GIA/CHI/MIN/ATL/RAMS

1587A0 JETS/PHI/NO/SF

Endzone palettes 1597E0-159B40

Team name logo palettes 3D402-3D4DA City name palette 2B532-2B60A

To edit the team name logos, the graphics are compressed and need to be viewed in 2BPP or game boy format in a tile editor.

Pro Bowl Rosters

AFC 170C00-170C44

NFC 170C48-170C8C

Play Formations

Runs 161460-16175E

Passes 16175F-161A55

00-Kickoff

01-Punt

02-XP/FG

03-XP/FG

04-Pro T (a)

05-Slot

06-Onesetback (a)

07-2 TE

08-Motion Down

09-Shifting Onesetback

0A-Oneback (as in Z-Cross)

0B-Offset I

0C-Run and Shoot

0D-Run and Shoot 3-Wing

0E-Shotgun (a)

0F-Shotgun (b)

10-Shotgun 3-Wing

11-Shotgun (c)

12-Redgun

13-No Back

14-Pro T (b)

15-Oneback (b)

Playbooks 170D2F

0-7 is the avaiable plays for each slot

8-F is a play from the next slot over

To edit the year (1993 Regular Season) find the 1993 text in a tile editor in 2BPP or game boy format, and edit the numbers.

(Red) + (Green \* 32) + (Blue \* 1024)

RGB would be the number / 8

so white.  255, 255, 255

is 255/8 = 31

31 + 31\*32 + 31\*1024 = 32767

32767 = x7FFF

and then reverse it xFF7F

There are different palettes for the endzones depending on the weather:

TEAM CLEAR RAIN SNOW

BUF 1597E0 15A1E0 15A5E0

IND 159800 15A200 15A600

MIA 159820 15A220 15A620

NE 159840 15A240 15A640

JETS 159860 15A260 15A660

CIN 159880 15A280 15A680

CLE 1598A0 15A2A0 15A6A0

HOU 1598C0 15A2C0 15A6C0

PIT 1598E0 15A2E0 15A6E0

DEN 159900 15A300 15A700

KC 159920 15A320 15A720

RAI 159940 15A340 15A740

SD 159960 15A360 15A760

SEA 159980 15A380 15A780

DAL 1599A0 15A3A0 15A7A0

GIA 1599C0 15A3C0 15A7C0

PHI 1599E0 15A3E0 15A7E0

PHX 159A00 15A400 15A800

WAS 159A20 15A420 15A820

CHI 159A40 15A440 15A840

DET 159A60 15A460 15A860

GB 159A80 15A480 15A880

MIN 159AA0 15A4A0 15A8A0

TB 159AC0 15A4C0 15A8C0

ATL 159AE0 15A4E0 15A8E0

RAMS 159B00 15A500 15A900

NO 159B20 15A520 15A920

SF 159B40 15A540 15A940

**Large Helmets**

The first offset is the helmet, the 2nd is the logo and facemask. The letter in () is for teams that share palettes. Example: IND(A) and PHX (A) share helmet colors.

BUF 15AA60 159388

IND 159560(A) 1595A0

MIA 15ABA0 159600

NE 15AB60 1595C0

JETS 15AA00 159260(B)

CIN 1592E0 159320

CLE 15AA40 159300

HOU 15ABC0 1595E0

PIT 15AB00 159480

DEN 159420 159440

KC 159340 159380

RAI 15AB40 159540 (C)

SD 1593E0 159400

SEA 159520 159500

DAL 15AB20 159540 (C)

GIA 15AAE0 159460 (D)

PHI 159240 159260 (B)

PHX 159560 (A) 159580

WAS 15AA80 159360

CHI 15AAA0 159460 (D)

DET 15AB80 1594E0

GB 1593A0 1593C0

MIN 159200 159220

TB 15ABE0 159620

ATL 15AAC0 159460 (D)

RAMS 1594A0 1594C0

NO 159280 1592C0

SF 15AA20 1592A0

SNES colors are 2 hex values (FF7F is white, for example). Each team logo uses a 16 color string with the last 3 colors used for the facemask. By default, the first 2 values are 0000's, the first one is transparent & allows the helmet color to show though, the 2nd one is pure black.

Each team's helmet shell also uses a string of 16 values, starting with two 0000's.

**Hex**

(4:52:48 PM) Jlstout123: (256 - xFF) \* -255 + xFC

(4:53:13 PM) Jlstout123: so 1 \* -255 + 252.  -255 + 252.  -3 Yards

(4:53:31 PM) Me: ok, how did you know to do that?

(4:54:00 PM) Jlstout123: something common in 6502 programming that I picked up over time

(4:54:25 PM) Jlstout123: In my program it has 2 equations with an if statement that checks for positive or negative

(4:54:54 PM) Me: oh, wait, so if it's a negative number, you have to use that formula?

(4:55:15 PM) Jlstout123: Yeah

(4:55:44 PM) Me: how do I know when to use that?

(4:55:46 PM) Jlstout123: So check the multiplier for less than x80 then use the positive equation else use the negative equation

(4:55:51 PM) Me: oh

(4:56:02 PM) Me: what is x80

(4:56:19 PM) Me: 128

(4:56:43 PM) Jlstout123: where it turns positive to negative. x80 to xFF is negative and x00 to x7F is positive

function GetMultipliedValue($remainder,$multiplier)

{

  if( $multiplier < 128 )

    return $multiplier \* 256 + $remainder;

  else

    return (256-$multiplier) \* ($multiplier\*-1) + $remainder;

}

  [rushyards\_r] => 252

            [rushyards\_m] => 255

**Uniforms**

Matchups

x1752 to x1755 (headerless rom) was Buffalo and Indianapolis directly followed. There was just the binary and the color areas are elsewhere (from jstout)

x00 x00 x00 x00 = When to Use Jersey 1 and 2 this is done bitwise (Buffalo to Atlanta then AFC to NFC with final 2 bits always being 0s) where a 0 = Jersey 1 and 1 = Jersey 2.

(x00000000 would use Jersey 1 vs every team and xFFFFFFFC would use Jersey 2 vs every team)

Palettes

Seattle Jersey:

Dark shade: 15858A

Light Shade: 15858C

**Juice**

Look for the following sequence at hex key x2699a:

0001df10h: 00 01 00 00 00 01 02 01 01 01 01 02 01 02 02 01

0001df20h: 02 01 03 02 02 02 02 03 03 02 02 02 04 03 02 02

0001df30h: 02 04 04 02 02 02 05 04 02 02 03 05 05 02 02 03

0001df40h: 06 05 02 02 04 06 06 03 02 04 07 06 03 03 04 07

0001df50h: 07 03 03 05 08 07 03 03 05 08 08 03 03 05 09 08

0001df60h: 03 04 06 09 09

The numbers go in blocks of 5, paste the following over each block of 5 to increase difficulty:

03 04 06 09 09

That will make every week play like you're undefeated in week 17.

Hex Boost Order: Defensive Speed, Offensive Speed, Interception, Pass Control, Reception

Each level is decided by the team's Win/Loss Record with each win raising the level of juice. A team with 0 wins and 0 losses is the first set of numbers (also the preseason juice). A team with 1 win and 0 losses plays against the same juice as a team with 6 wins and 5 losses (the second set of numbers). Any team with 16 or more wins will play vs the final set of computer juice.

From what I could tell, the first set of numbers also affects some things like return speed for the MAN players and the like.

**Simming**

For Defense Sim : [ x, y ]

x is sim value for Sacks (for each team, ALL defenders' x's should add up to 255)

y is sim value for Interceptions (for each team, ALL defenders' y's should add up to 255)

"add up to 255" means that a defenders' value should be a fraction (whole number, of course) of 255 - ie, a fraction of all his/her team defenders' total of 255; with a greater number implying more sacks/ints.

read this thread: [viewtopic.php?f=5&t=11924&st=0&sk=t&sd=a](http://www.knobbe.org/phpBB2/viewtopic.php?f=5&t=11924&st=0&sk=t&sd=a" \t "_blank)

For Offense Sim, non QB : [ a, b, c, d]

all of these can range from 0-15

"d" is sim value for "receiving targets", and is how often the RB or WR is targeted for a reception during sim - Passing and Receiving Ability work together in deciding a reception, incompletion, or interception on the play.

read this thread from first to last post: [viewtopic.php?f=5&t=11924&st=0&sk=t&sd=a](http://www.knobbe.org/phpBB2/viewtopic.php?f=5&t=11924&st=0&sk=t&sd=a" \t "_blank) and I think you will understand.

(I won't go over details of a, b, or c - they've been covered enough everywhere else, (see above thread link, page two, jstout's post), or sblueman's sim thesis which you've already read, bucs2nine)

Simming -

I decided to put together a breakdown of how I worked the sim code for accurate player and team performances. Just a note, the players physical abilities are edited but I tried to steer clear of blowing up the players stats in my Tecmo Super Bowl 2001 game so what I did was take the attributes from the players in the original game and apply them to players in my edited version. The only players in the game that do not have a sim code to edit are the offensive linemen. So without further adieu he is my breakdown by position:

Quarterbacks:

The QB is the player who dictates the flow of the offense. Have a good QB and you will be successful in Tecmo, have a bad one and expect to struggle when the defenses stack up vs. the RB. But the QB needs the targets to go to as well, and the sim code of other players impacts the QB’s performance as well. A good example is the Rams, where Bruce is the 1st option for Kurt Warner (Bruce has an 11 receiving rating) then Torry Holt (10 rating) then Conwell (9 rating) then Faulk (3 rating). So that is how Warner reads his progressions, Bruce, Holt, Conwell and then Faulk. To force the game to make the Rams pass more I gave Kurt Warner the following code:

Kurt Warner - Stl

Rushing ability: 5

Passing ability: 14

Pocket: 1

Warner has a high passing ability for two reasons. One is so he throws alot and gets the huge numbers he ends up with year in and year out and two it makes the Rams throw more than run like they did before in the 1991 version. So now the Rams have over 3,500 yards passing and a RB with close to or over 1,300 yards. Now you get to the part of QB’s who are not good. Here are a couple examples of what I did to adjust the codes of some mediocre players:

Tim Couch- Cle.

Rushing ability: 6

Passing ability: 2

Pocket: 0

Tony Banks-Was

Running ability: 2

Passing ability: 4

Pocket: 0

Charlie Batch-Det.

Running ability: 5

Passing ability: 2

Pocket: 1

Both Charlie Batch and Tim Couch play in very weak offenses that do not have a RB that is a superstar who can offset a bad passing attack. Tony Banks on the other hand has Stephen Davis and he is a workhorse who normally ends up around 1,400-1,500 yards. These QB's will not put up great #'s and will never crack the top 15 in the QB rating stats.

Running Backs:

The thing to remember is that Tecmo tends to be a RB dominated game. The RB’s seem to get many touches in the game if there are enough plays that they are involved in. Give the RB's high totals in their sim code and they will hog the ball. When I began editing I used to give great RB's with great receiving and rushing ability high numbers on both categories. Well, I found out when you do this you end up with a RB who finishes the year with over 1,000 yards receiving and almost 2,000 yards rushing and some ridiculous total of about 50 TD's. Obviously something had to be adjusted. So I tinkered with the sim code week after week, adjusting it trying to get the desired results. And here is what I came up with:

I realized first the players’ stats are definitely affected by the playbook. If you have your team running "Play action Z in" and your RB had decent receiving totals, expect him to get over 1,000 yards. So I limited this play to just teams that did not have strong receiving RB's. Obviously the number of rushing plays affect the totals. The final thing I did after that was adjust the sim code accordingly. Here are some examples of RB's and their codes:

Priest Holmes-KC (3 rushing plays out of 4)

Rushing Ability: 10

Receiving Ability: 2

Punt Return Ability: 6

Kick Return Ability: 2

Marshall Faulk-Stl. (3 out of 4 rushing plays)

Rushing Ability: 8

Receiving Ability: 3

Punt Return Ability: 5

Kick Return Ability: 3

Faulk might be the better athlete out of the two, but Holmes did have the bigger numbers running the ball. Faulk is the bigger threat as a receiver but in the Rams pass happy offense I had to adjust his Receiving and Kick Return ability (they both go hand in hand with receiving totals). That way he still gets his touches but does not end up with over 1,500 yards in the air.

Receivers and TE’s:

My version of Tecmo has receivers putting up big seasons like David Boston, Terrell Owens, Rod Smith and others. I learned that the receiving ability property is not the only one that affects the receivers season totals. Take for example my sim code entry for David Boston:

David Boston – Arz.

Rushing Ability: 6

Receiving Ability: 13

Punt Return Ability: 12

Kick Return Ability: 12

With these totals Boston regularly gets about 1,300-1,400 yards receiving with about 50-60 catches per season. 50-60 catches in Tecmo are usually about 80-90 in an actual NFL season.

Basically what I learned is that the receiver total alone does not dictate the receivers performance. The kick return ability seems to go hand in hand with the receiving ability and the punt return ability goes hand in hand with the rushing ability. Receiving ability dictates how many touches the player gets. Kick return ability seems to affect the players ability to get YAC yards (Yards after catch) and the punt return ability seems to affect the players ability to be shifty in the open field.

Offensive Linemen:

Now offensive linemen do not have sim codes that can be adjusted, so all you have to go on is their physical abilities. To make the teams offensive lines play similar to how they did in the actual 2001 season. Here is how I did it. I first looked up the sacks allowed by the teams in 2001. This usually is the best way to determine the strength of an offensive line. From there I edited the offensive line player stats then used loaded the ROM into Tecmo Manager 2000. The great thing about Tecmo Manager 2000 is it has a scale that lists the teams’ positions by strength. Using this tool I began adjusting the OL’s until I had it in order of the sacks allowed in 2001. The Bears had the best offensive line in 2001 while the Falcons had the weakest. Here are some examples of a great OL and a mediocre OL:

Top ranked OL:

Jonathan Ogden – Bal.

Rushing speed: 44

Rushing power: 81

Maximum speed: 56

Hitting power: 88

Body balance: 25

Agility: 50

Last ranked OL:

Corey Hulsey – Buf.

Rushing speed: 13

Rushing power: 38

Maximum speed: 19

Hitting power: 31

Body balance: 25

Agility: 50

Defensive Linemen & Linebackers:

The most frustrating thing about the 1991 version of Tecmo is Tecmo Inc. decided in giving every team a 3-4 defensive alignment. The 3-4 alignment did not fit over half of the teams at the time and now in 2001 only one team runs the 3-4 exclusively. So what I did was change the ROLB position to a DE. The reason I did this was because when the original game was around, I was a huge Houston Oilers fan and used Ray Childress to rush the passer (see how F’ed up Tecmo is? They have Ray Childress; a DT lined up as a linebacker! What is wrong with that picture?) T

Then when I began running simulated seasons I noticed something. The ROLB hardly picks up any sacks in the game and it does not matter what value you give that players pass rush ability in the Sim editor, he will not pick up many sacks. So I switched the DE spot to the LOLB position, the same one Lawrence Taylor played in the original game.

Well now that I got the alignment out of the way I had to adjust the sim code. I once saw somewhere in a message board that the highest total you can give a defensive player is 115. Well I was looking at how the sim code was set up in the original game and Derrick Thomas had a 137 Pass rush rating! So I began tinkering with the sim code. First thing I did was give Michael Strahan the highest pass rushing totals:

Michael Strahan – Nyg.

Pass rush: 137

Pass coverage: 5

This way he is the best pass rusher in the game. Now I began setting the pass rush code to around 90-100 for the top pass rushers in the game (Jason Taylor of Mia, Jevon Kearse of Ten, Marcellus Wiley of SD to name a few). Then I noticed I had to set it a bit lower for the DT’s (the RDE and NT in the original Tecmo). Players like La’Roi Glover, Trevor Pryce and Sam Adams were dominating the sack race. They were ending up with over 20 sacks regularly. So for DT’s I set the pass rush code a bit lower, around 50-70 for top notch DT’s. Here are a few:

Trevor Pryce – Den.

Pass Rush: 69

Pass coverage: 8

La’Roi Glover – N.O.

Pass rush: 58

Pass coverage: 8

This way the DT’s get their numbers but do not rack up 20 or more sacks every year.

Now the linebackers in the game who are obviously different than the defensive linemen. Obviously with the linebackers they have to cover more than defensive linemen do so their pass coverage code will be set higher. As for the pass rush code, I set it according to the players’ stats. Players like Peter Boulware of the Ravens has a higher pass rush code than Derrick Brooks of the Bucs.

Peter Boulware – Bal.

Pass Rush: 88

Pass coverage: 8

Derrick Brooks – T.B.

Pass rush: 78

Pass coverage: 20

Now notice the pass coverage stats. Boulware is a pass rush specialist who is not that strong in his pass defense. Derrick Brooks might be the most complete LB around, and his pass coverage stats are higher. It’s not up there like a DB’s but it is set at a good level for a LB. I try not to set the LB’s with too high a pass rush ability so you don’t end up with a MLB with 15 or more sacks. One position that is especially frustrating is the LILB position. If you have ever played a entire season and call a lot of the pass plays you will more than likely end up with the player in this position with many sacks. He seems to always be the first one who can shoot that gap and get there the fastest out of all the blitzers. So what I have done is make sure this spot’s player is NOT the pass rush specialist. People are used to rushing the passer with the ROLB like Lawrence Taylor and Derrick Thomas (you get a great angle on the OT with this spot). The player in the RILB spot usually has a pass rush code that is lower than usual.

Defensive backs:

Defensive backs are not too hard to figure out. They don’t pass the rusher much so their pass rush code does not have to be set too high. This stat seems to affect their run support more so I set it usually around 12-17 for top notch run stuffers. Now if you have a talented DB I set his pass coverage ability around 70-80. A DB who is a great defender gets his ability set at 80-90 and a DB who gets many Int’s is around 90-115. I did not go over 115 much so as to avoid having a DB end up with a crazy amount on Int’s for a season.

Take a look at some good examples like Ronde Barber of the Bucs (10 Int’s in 2001), Lawyer Milloy of N.E. (one of the top run stuffing safeties around) and Charles Woodson of Oak. (Probably the premiere cover corner)

Ronde Barber – T.B.

Pass rush: 13

Pass coverage: 116

Lawyer Milloy – N.E.

Pass rush: 18

Pass coverage: 105

Charles Woodson – Oak.

Pass rush: 14

Pass coverage: 84

Woodson is a great cover corner but he did not get many Int’s because he was rarely thrown to. Milloy’s pass rush ability is set higher than others to show his prowess in the run defense. And Barber has a 116 rating since he was one the NFL leader in Int’s.

Kickers & Punters:

These players sim code is just one digit for his kicking ability. Basically the better the kicker the higher the code. Top kickers like Jason Elam of Den. (10 rating) and Matt Stover of Bal. (11 rating) have higher sim codes than struggling kickers like Rian Lindell of Sea. (6 rating) and Kris Brown (5 rating). Same goes for punters. The better the player the higher the code. Pretty simple.

Return men:

Try this. Edit and save your return men in the Tecmo Manager program, then open up your game with whichever emulator you use. Reset the rosters and the correct return men will be set. Just a note, I noticed that unless you single out one player as the return specialist, Tecmo Manager will pick the 1st string RB as a return man for either one or both positions. A good example would be like Jermaine Lewis and Tiki Barber. In my version Jermaine Lewis is set as the punt and kickoff return man for the Ravens while Tiki Barber is set as the punt returner while Ron Dixon is the kick returner for the Giants. I recommend having one player as a return specialist. Here are some examples of some of the dangerous return men’s sim codes:

Jermaine Lewis – Bal.

Rushing ability: 2

Receiving ability: 6

Punt return ability: 14

Kick return ability: 14

Desmond Howard – Det.

Rushing ability: 2

Receiving ability: 4

Punt return ability: 15

Kick return ability: 13

Overall Team Rankings:

I wanted to make a game that was as accurate as possible when the season ended and the playoffs came around. I didn’t want a game that would make the Bills or Giants division champs in 2001 even though they were good teams in 1991. If you open up my version with the Tecmo Sim Editor, you will see there is a big gap between the good teams and bad teams. Really bad teams like the Bills and Lions have 0’s for both offense and defense. The top teams in my Tecmo 2001 have a sim code adjusted so they consistently make the playoffs. Now there are some fringe teams that did not make the playoffs but a win here and a loss there by a playoff bound team would have put them in. So teams like Denver, Seattle and Arizona might sneak into the postseason. The highest rating a team can get either on offense or defense is 15. Teams with a 15 rating on either spot are:

Pittsburgh Steelers

Offense: 11

Defense: 15

Philadelphia Eagles

Offense: 9

Defense: 15

Chicago Bears

Offense: 7

Defense: 15

St. Louis Rams

Offense: 15

Defense: 15

Now I know people are going to say that the Rams should not be the top ranked team in the game since they did not win the SB. Well here is the Super Bowl winners code:

New England Patriots

Offense: 14

Defense: 14

The reason I did this was because all throught the year the Rams were the best team in the NFL. They dominated with an offense that seemed unstoppable and a defense that was very stout. When they lost the Super Bowl many experts put this upset up with the Colts losing to the Jets in Super Bowl III as the biggest upset in NFL history. So by making the Rams the top team, anyone that plays a season and meets them in the playoffs either in the Super Bowl as an AFC team or in the NFC playoffs will have to put on a great game to beat them as did the Patriots.

Now on player talent alone I used the Tecmo Manager 2000 to adjust the teams until it was in order from the best to the worst. And here is the list:

1- St. Louis Rams: 46.6

Sim Code- Off: 15 Def:15

2- Philadelphia Eagles: 44.0

Sim Code- Off: 9 Def:15

3- (t) New England Pats: 41.8

Sim Code- Off: 14 Def: 14

3- (t) Pittsburgh Steelers:41.8

Sim Code- Off: 11 Def: 11

5- San Francisco 49ers: 41.4

Sim Code- Off:12 Def:7

6- (t) Chicago Bears:41.0

Sim Code- Off: 7 Def: 15

6- (t) Oakland Raiders:41.0

Sim Code- Off:10 Def: 5

8- Green Bay Packers: 40.2

Sim Code- Off:11 Def: 8

9- Miami Dolphins:39.3

Sim Code- Off: 8 Def: 14

10- (t) New York Jets:39.2

Sim Code- Off: 5 Def: 7

10- (t) Tampa Bay Bucs:39.2

Sim Code- Off: 5 Def: 11

12- Seattle Seahawks:38.0

Sim Code- Off: 3 Def: 3

13- Baltimore Ravens: 37.9

Sim Code- Off: 7 Def: 14

14- Arizona Cardinals: 36.9

Sim Code- Off: 1 Def: 0

15- New Orleans Saints:36.7

Sim Code- Off: 1 Def: 3

16- Denver Broncos: 36.4

Sim Code- Off: 2 Def: 0

17- Tennessee Titans:35.9

Sim Code- Off: 5 Def: 1

18- (t) Indianapolis Colts: 35.7

Sim Code- Off: 3 Def: 0

18- (t) New York Giants: 35.7

Sim Code- Off: 2 Def: 4

20- Wash. Redskins: 35.6

Sim Code- Off: 2 Def: 4

21- Atlanta Falcons: 35.3

Sim Code- Off: 3 Def: 3

22- Dallas Cowboys: 35.1

Sim Code- Off: 0 Def: 2

23- San Diego Chargers:

**TSB Tool**

TSB Tool -

(0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f) with 0 being the worst and f

**Hacks for Tecmo Super Bowl (SNES)**

*The credit for most of these go to xplosv who ported them to the SNES from existing NES hacks*

|  |  |  |
| --- | --- | --- |
| **Hack Name** | **Location** | **Notes** |
| Late season coverage hack | x19519 | set to xEAEA |
| Passing Speed | x18764 |  |
| Passing Arc | x18774 |  |
| Rushing Power | x18784 |  |
| Running Speed | x18794 |  |
| Pass Block | x187A4 |  |
| Kick Block | x187B4 |  |
| Fumbles | x187C4 |  |
| Pass Control | x187D4 |  |
| Receptions | x187E4 |  |
| Interceptions | x187F4 |  |
| Kick Arrow Speed | x1B7B6 |  |
| Kick Arrow Distance | x1B7C6 |  |
| QB Fumbles | x18FA9 |  |
| Make PR BC = 56 | x18F7E | Change x1F to x14 |
| On-field clock speed | x24727 | Change x14 to a higher value = slower clock, lower value = faster clock |
| Playcall screen clock speed | x2476D | Change x1E to higher value = slower clock, to lower value = faster clock |
| Playcall run off time | x24769 | Change x0A (10 seconds) to number of seconds in hex |
| Shorten Punts | x19F3B | Change xF0 to x60 for realistic punts |
| Arc/Speed of Punts | x19F50 | Lower the value, higher the arc & slower kick |
| Arc of kickoffs | x1B3E6 | Lower the value, higher the arc & slower kick |
| Presses to win grapple | x18D88 | Change x02 to number of presses in hex |
| Default Grapple time | x18D7F | Change x40 to time in hex (x14 = 1 tecmo second) |
| Loft of snap | x19710 | Lower the value, lower the snap |
| Speed of snap | x19714 | Higher the value, faster the snap |
| Loft of pitch | x1BA2D | Lower the value, lower the pitch |
| Speed of pitch | x1BA31 | Higher the value, faster the pitch |
| Remove defensive diving | x1AA0E | Change x20 A5 63 to xEA EA EA |
| Remove defensive jumping | x1A9FC | Change x4C 71 AA to xEA EA EA |
| More defensive jumping | x1A9F5 | Change x90 08 to xEA xEA |
| Adjust injuries | x13A61 | Change xC5 0D to xC9 "XX" where XX is the probability of injuries (out of 255) |
| Turn on preseason injuries | x123BF | Change x80 to x02 |
| In-season quarter length | x9EEE | default = x05 |
| Remove Defensive Diving | x1AA0E | SET0(0x1AA0E, 0xEAEAEA) |
| Remove Offensive Diving | x1A76E | SET(0x1A76E, 0xEAEAEA) |
| Remove Defensive Jumping | x1A9FC | SET(0x1A9FC, 0xEAEAEA) |
| Remove Offensive Jumping | x1A752 | SET(0x1A752, 0xEAEAEA) |
| Use player attributes for INT return speed | x167E44 | SET(0x167E44, 0xE100E206E300E4DF) |

**Remove Grapple Hack**

SET(0x18d83,0xa01db163c902)

SET(0x18dc3,0xa01db103d163)

SET(0x1d010,0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffdfffffff6ffffffffbffeff)

SET(0x18dac,0x20b0c3)

SET(0x1d068,0xffff7fffffffffefffffffffffffffffffffffffffffffffffffffffffffffffffffff)

**Variable OL/ DL HP Hack**SET(0x18BB8,0xB16338F1039163AD0002293F187163300BC90E9007C93490064C8E8C4C158D20B0C3EAEAEAEAEA)

If you lower the value in red, the defender will be thrown more often. If you raise the value in blue closer to the value in red, the offensive player will be thrown more often. The wide the gap in the two values will cause a normal grapple to happen more often.

**Quickness for Incompletions**

SET(0x1ABC8,0x4C00D0EA)

SET(0x1ABDB,0x4C10D0EA)

SET(0x1ABEE,0x4C20D0EA)

SET(0x1D070,0x860A2A600C6A298F22268483A0874CD3AC4CF5AC)

SET(0x1D010,0xA692A5A02060D0A4A0C95090234CCCABA692A5A12060D0A4A1C95090134CDFABA69

<http://tecmobowl.org/forum/viewtopic.php?p=125865#p125865>

x18794, this is the game speed slider. There are 16 bytes it starts with 10 ends with 1F.

Bumping these numbers up will increase all the running attributes (rushing speed, rushing power, and maximum speed all at once.). All 3 have their own slider, but this one increases them all at the same time. I added 10 (making it 20-2F) and it gets very fast. So tinker with it to your liking.

Tested different speeds and "18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27" plays exactly like the NES speedwise. I haven't tested any passing speeds yet though. (8 notches)

It seemed a bit too fast for me at these settings (could just be my game)? I lowered them down one notch each and ran the NES and SNES side by side. It seemed identical at one notch lower. I could be imagining things though.

If you want to increase the passing speed to adjust it to fit the new game speed that starts at x18764. Again 16 bytes, starting with 44 and ending with 62.

1st screen (game start) starts at hex- x162D00  
2nd screen (preseason, season, etc) - x162C00

**Small helmet palettes**  
158740 IND/MIA/CIN/CLE/TEN/RAI/SEA/PHX/DET/TB  
158760 BUF/NE/KC/WAS/GB  
158780 PIT/DEN/SD/DAL/GIA/CHI/MIN/ATL/RAMS  
1587A0 JETS/PHI/NO/SF

**Large helmet palettes**

The first offset is the helmet, the 2nd is the logo and facemask. The letter in () is for teams that share palettes. Example: IND(A) and PHX (A) share helmet colors.  
  
BUF 15AA60 159388  
IND 159560(A) 1595A0  
MIA 15ABA0 159600  
NE 15AB60 1595C0  
JETS 15AA00 159260(B)  
CIN 1592E0 159320  
CLE 15AA40 159300  
HOU 15ABC0 1595E0  
PIT 15AB00 159480  
DEN 159420 159440  
KC 159340 159380  
RAI 15AB40 159540 (C)  
SD 1593E0 159400  
SEA 159520 159500  
DAL 15AB20 159540 (C)  
GIA 15AAE0 159460 (D)  
PHI 159240 159260 (B)  
PHX 159560 (A) 159580  
WAS 15AA80 159360  
CHI 15AAA0 159460 (D)  
DET 15AB80 1594E0  
GB 1593A0 1593C0  
MIN 159200 159220  
TB 15ABE0 159620  
ATL 15AAC0 159460 (D)  
RAMS 1594A0 1594C0  
NO 159280 1592C0  
SF 15AA20 1592A0  
  
SNES colors are 2 hex values (FF7F is white, for example). Each team logo uses a 16 color string with the last 3 colors used for the facemask. By default, the first 2 values are 0000's, the first one is transparent & allows the helmet color to show though, the 2nd one is pure black.  
Each team's helmet shell also uses a string of 16 values, starting with two 0000's.

**Team name logo palettes**

3D402-3D4DA

**City name palette**

2B532-2B60A

To edit the team name logos, the graphics are compressed and need to be viewed in 2BPP or game boy format in a tile editor.

**Converting an RGB color to hex:**

(Red) + (Green \* 32) + (Blue \* 1024)  
  
RGB would be the number / 8  
so white.  255, 255, 255  
is 255/8 = 31  
  
31 + 31\*32 + 31\*1024 = 32767  
  
32767 = x7FFF  
and then reverse it xFF7F

<http://tecmobowl.org/topic/10512-applying-sim-data/>