

This is where I dump ideas, so there may be some redundancies.

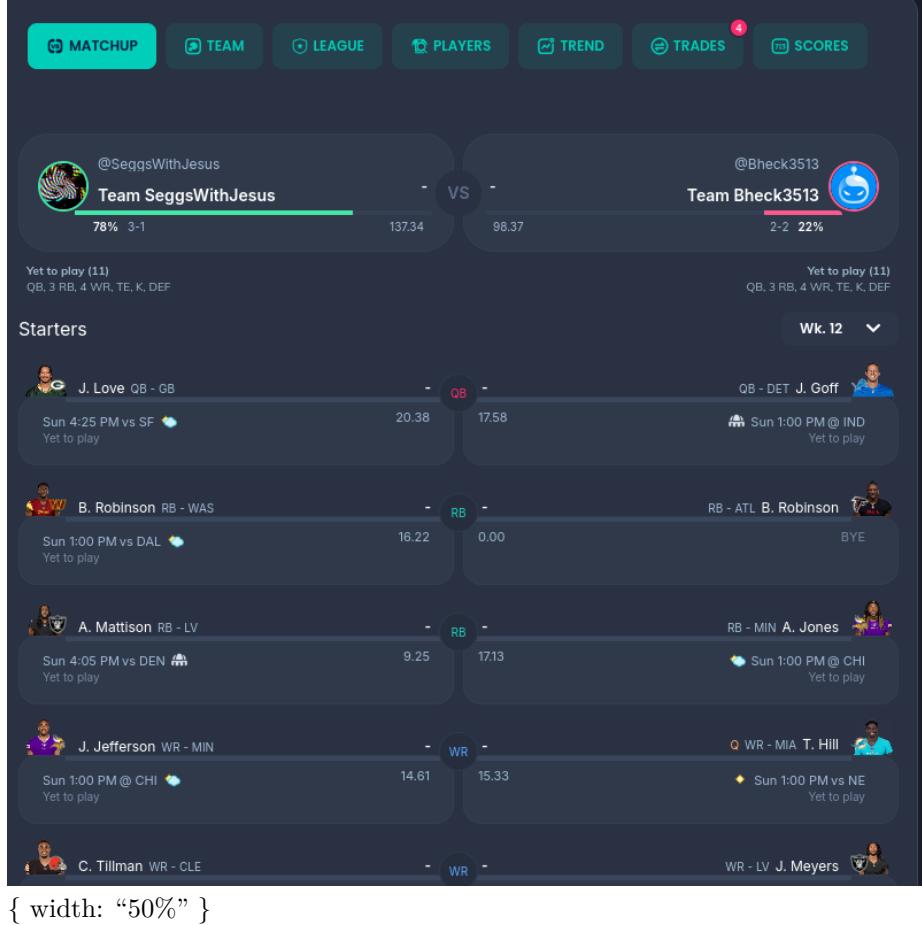
Use-Case Driven UIs

Mid-Season Draft Tool

- The Player List UI 1 (Fig. 1), but modified so that each row allows some extra info to be displayed, including one or more of
 - some summary representation of several player ranking measures
 - a small message or number conveying extremely salient contextual information (injured: projected return date, teammate projected return date)
 - would-be-if-drafted team measures, or
 - the difference between $m(T)$ and $m(T - \{p\})$ (where T is user's team and p is the player) for currently-selected measure m , or
 - a summary representation of the above-defined difference for multiple measures.
- Clicking on a player will display a pop-up with details
 - performance stats for season games to date
 - additional contextual fields in each row of game data - roster, quarterback, target share
 - Potentially also some creative “timeline” display against it, showing team states and their duration.

Compare		SELECT team	to	SELECT team	<input type="checkbox"/> Highlight Waivers	<input type="checkbox"/> Filter	
*Indicates Starter.							
QB	RB	WR		TE			
1. Lamar Jackson 2. Jalen Hurts 3. Josh Allen 4. Joe Burrow 5. Jayden Daniels 6. Kyler Murray 7. Baker Mayfield 8. Justin Herbert 9. Jordan Love 10. Patrick Mahomes 11. Brock Purdy*	2379 7240 6055 5641 4633 3606 3218 2761 2479 2468 2384 2379	1. Saquon Barkley 2. Christian McCaffrey* 3. Bijan Robinson 4. Derrick Henry* 5. Jahmyr Gibbs 6. De'Von Achane 7. Joe Mixon 8. Alvin Kamara* 9. Breece Hall 10. Kyren Williams 11. Jonathan Jacobs 12. Jared Off	10527 9623 9334 8774 8043 7571 7532 7439 7880 6441 5839 5838 5796 5078 4964 4646 4195 3394 3357 3120 3085 3023 2861 2703	1. Ja'Marr Chase* 2. Justin Jefferson 3. Amari'Co St. Brown 4. Nico Collins* 5. A.J. Brown 6. Puka Nacua* 7. Cooper Kupp 8. Drake London* 9. Tyreek Hill 10. CeeDee Lamb* 11. Malik Nabert 12. Garrett Wilson 13. Kenneth Walker 14. David Montgomery 15. Marvin Harrison Jr.* 16. DK Metcalf 17. Terry McLaurin 18. Tee Higgins 19. Zay Flowers 20. Debo Samuel 21. Davante Adams 22. Jayden Reed 23. Jaxon Smith-Njigba 24. Brian Thomas	10402 9488 8614 7826 7648 6681 6490 6091 6078 6005 5316 5030 4710 4423 4359 4147 4143 3805 3576 3260 3018 2942 2890 2421	4866 4773 4052 3947 1939 1819 1710 1501 1274 1100 1081 1055 865 718 675 652 603 394 378 308 261 246 233 233	1. Travis Kelce 2. Brock Bowers 3. George Kittle* 4. Trey McBride 5. T.J. Hockenson* 6. David Njoku 7. Sam LaPorta 8. Cade Otton 9. Mark Andrews 10. Kyle Pitts 11. Evan Engram 12. Dallas Goedert 13. Dalton Kincaid 14. Taysom Hill 15. Jake Ferguson 16. Tucker Kraft 17. Jonnu Smith 18. Hunter Henry 19. Zach Ertz 20. Dalton Schultz 21. Cole Kmet 22. Pat Freiermuth 23. Will Dissly 24. Mike Gesicki
19. Anthony Richardson 20. Drake Maye* 21. Caleb Williams 22. Matthew Stafford 23. Sam Darnold 24. Aaron Rodgers	687 550 535 525 447 378	10. Islan Pacheco 21. D'Andre Swift 22. Aaron Jones 23. Brian Robinson 24. Tyrone Tracy	2975 3085 3023 2861 2703	2975 3085 3023 2861 2703	2410 2410 2410 2410 2410	2410 2410 2410 2410 2410	

width: "50%" }



Trade Tools

Trade Manual Analysis Could be integrated into previously-mentioned roster UI styled after the typical “Matchup” UIs (Fig. 2). We want to show comparison of two values: $m(t_0)$ and $m(t_1)$, where m is the currently-selected measure, t_0 and t_1 are user’s team before and after the trade, respectively.

Trade Finder Search league for trade that would optimize some user-specified objective function $f(X)$ and satisfy a set of user-specified constraints $g_1(X) < c_1, \dots, g_n(X) < c_n, X \in D$. We could define the feasible set D in multiple ways, but for the moment let X be some matrix representation of both teams’ before and after states. (Obviously, the UI will enable the user to specify these in a more intuitive way. No mathematical optimization experience required.) For example

- “mutually advantageous”
 - objective function: $f(X) = gain(t_{user}, \text{trade})$
 - constraints: $gain(t_{opponent}, \text{trade}) > 0$,
 - where $gain(t, \text{trade})$ is how many more fantasy points team t is projected to score, rest-of-season, after making **trade**. (where we consider **trade** to be a function over the collection of sets of players, $gain(t, f) = p(f(t)) - p(t)$ where f is the trade and p is projected rest-of-season fantasy points.)
 - i.e. maximize our benefit whilst still being beneficial to the other team.
- “maximizing difference between apparent and actual opponent-team gain”
 - e.g.
 - * apparent: free rankings provided online that don’t take into account league settings
 - * actual: projected points using league settings
 - * this wouldn’t be a great thing to optimize, but would be interesting to see. A better set up would be “maximize actual user-team gain while keeping apparent opponent-team gain over a minimum threshold”

We’ll have to show the above defined optimization problem can be solved, or approximated to some degree of tolerance, reasonably efficiently. On second thought, it might be best to think of the criteria/constraints ourselves and give the user a set of pre-defined options. Or just present the trades we find using our hard-coded optimization problems to the user. I do think an easy-to-understand (potentially simplified) explanation of what was optimized would be very beneficial.

Trade Retrospective This would be a *great* way to draw people to site. Go through past trades and enable evaluation of their fairness/quality based on

- information available at the time
- how they actually affected teams afterwards

General Entity-Representation UIs

Player

We want to have a page that conveys a lot of information about a player at one time, including

- how that player's projections have been trending over the last week
- their performance history
 - raw performance metrics
 - fantasy points, using league settings
- context for past performances - Look into timeline UIs
 - their injuries (obviously)
 - teammates' injuries
 - team roster changes
 - strength of matchups
- performance projections
 - stats projections
 - ScoringConfig (stats projections) -> fantasy projections
- context for future
 - returning teammates
 - strength of matchups

NFL Teams

- fantasy points and stats allowed - to position - drill down into wide receiver types
- fantasy points and stats allowed - to play type - short/long pass, qb rush
- fantasy points and stats allowed - to (passing, rushing, receiving)
- defensive roster changes

League Teams

- 2-Team Comparison of current players' past performance
- 2-Team Comparison of current players' projected performance
- Able to change ranking measure for each of listed UIs
 - ranking measures
 - * (past/projected performance) by (points/stats)
 - * projected performance add'l inputs
 - can use raw-rankings projections for standard-scoring (PPR/HALF/STD), as there are *a lot* more opinions available than with stats projections
 - next week/ROS
 - UIs
 - * PlayerListUI1 (Fig 1.)
 - * "Matchup"-style UI (Fig 2.) with players' values next to them, and totals at op
 - some sort of summary comparison