



Theming Configuration Redesign

TABLE OF CONTENTS

Overview

Broader Initiatives

Research

Design Process

Learnings



OVERVIEW

Background

Skillz provides a library of **developer tools** to make creating games easier

Objective

The aim of this project was to revamp our **Theming Configuration** experience

Role

I worked as the **sole designer** on the developer experience team



PREVIOUS DESIGNS

The previous designs was a dropdown lists nested within a dropdown list of every single element you could update on the mobile SDK

The screenshot shows the Skillz mobile SDK theming tool interface. On the left is a dark sidebar with the Skillz logo and navigation links: Dashboard, Games, Analytics, Statements, Documentation, Downloads, Support, Account, and Log Out. Below these is a 'SPECIAL EVENT' section for the 'NFL Challenge'. The main content area is titled 'Test' by 'Untitled Publisher' with 'Game Id 16725'. It has tabs for Overview, Optimize (selected), Game Builds, and Progression. A text block explains that users can customize the Skillz UI design using a theming tool. Below this is a section titled 'Select an out-of-box theme for your game' with a dropdown menu currently set to 'Light'. A list of theme categories follows: 'Create a custom theme to better match your game', 'Core Colors', 'Accent Colors', 'Buttons', 'Text', 'Artwork', and 'Advanced'. The 'Advanced' section is expanded, showing 'Cell Alternate Background Color' with a color picker set to '#f6f6f6' and a description: 'Alternative cell background color, for tables with alternating colors ie: taunts/leaderboards.' Below that is 'Opacity None' with a value of '1'.



PREVIOUS DESIGNS

These theming fields mapped to the mapped to the mobile elements annotated below

Core Colors

Background Gradient

#ccd3da

App wide default background gradient if no background image is supplied (and while supplied background image)

Main Color

rgba(255, 255, 255, 1.0)

Primary view background color, ie: tab bar background.

Main Color Opacity

0.9

Used where opacity is required (will replace given opacities for colors, not multiply)

Main Divider Color

#cfd9df

Used for background of Pending/Placeholder Avatar, and a few different dividers. (darker divider)

Secondary Color

rgba(255,255,255,.90)

Secondary view color, ie: Match Select cell background

Secondary Color Opacity

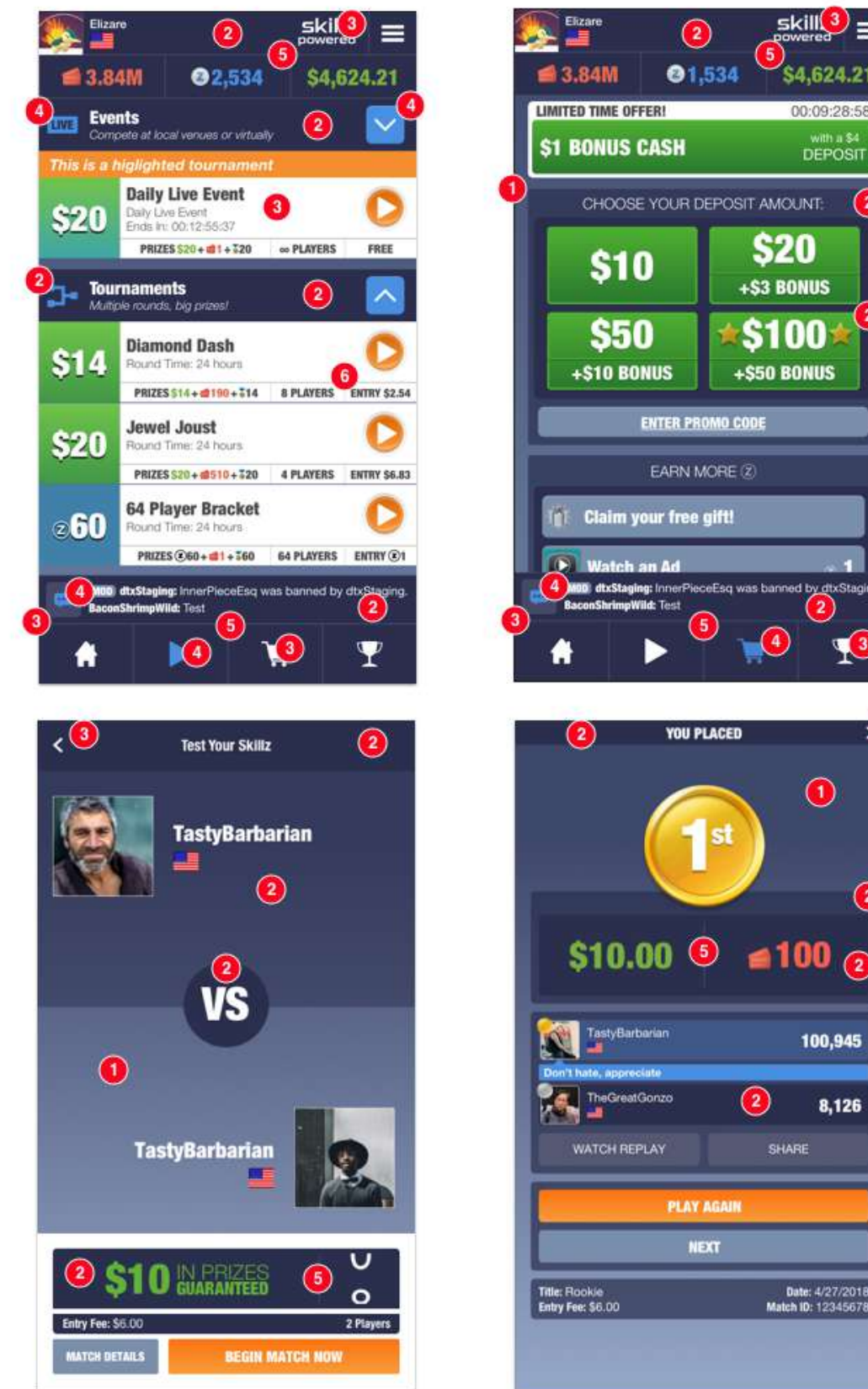
0.8

Used where opacity is required (will replace given opacities for colors, not multiply)

Secondary Divider Color

#bfc0c1

Used for borders of avatars, flags and checkboxes. (lighter divider)



Core Colors

- 1 Background Gradient
- 2 Main Color
- 3 Secondary Color
- 4 Tab Highlight Color
- 5 Main Divider Color
- 6 Secondary Divider Color

Notes:
There are several areas where the opacities are hardcoded resulting in a lighter color. This especially applies to the "Main Color."



PREVIOUS DESIGNS

Once developers finished updating their designs they would need to download a json to preview their changes on a local environment, then repeat the whole cycle if they wanted to make any other changes.

Save your theme

Once you're happy with your theme, click "Save." We'll compile a JSON file containing your theme for you to import into your Unity/Cordova project below.

Save

Sync theme to your live game

If you are ready to sync your saved changes to your live game, click 'Request Sync'. A member of our integrations team will send you an email confirmation within 5 business days once your theme has been synced to production.

Request Sync



Tue Jun 22 2021 10:54:09 GMT-0700 (Pacific Daylight Time) - Your request has been sent to our integrations team! A member of our team will email you within 5 business days to let you know when your theme has been synced to your production game.

Any changes saved, after you have requested sync, will be included in the synced theme.

Download your theme

You may now download your updated theme **Light** below and follow these instructions to add it to your game. Please note that it may take 30 or more minutes for custom images to appear in your theme.

[Theming instructions](#)

Platform	Last Updated	Download
iOS	2021-06-22 17:55:15 UTC	
Android	2021-06-22 17:55:15 UTC	

SYNC



PROBLEMS

Problem #1:

No Visual Feedback

As developers updated their designs, there was no way to see what changes they actually made.

Problem #2:

Too Many Fields

Another glaring issue was the sheer number of choices developers needed to make in order to fully reskin their game.

Problem #3:

No help for beginners

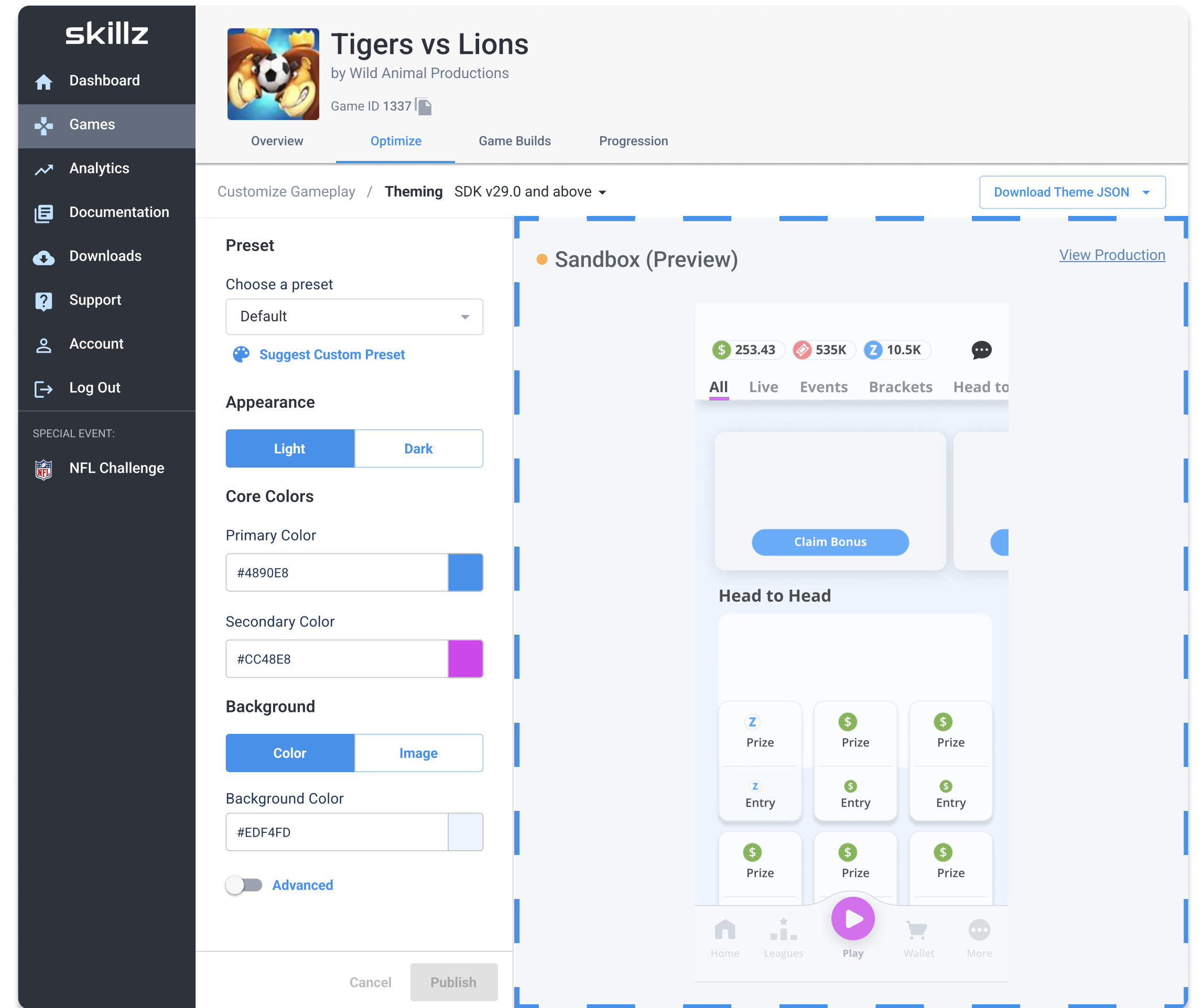
Making a game is already hard enough – doubly so for new developers who don't know how to best design their games.



SOLUTION

Preview Window

In the newly designed theming configuration page, the biggest update was adding in a new **preview window**, so developers could see what they were changing in real time.



PROBLEMS

Problem #1:

No Visual Feedback

As developers updated their designs, there was no way to see what changes they actually made.

Problem #2:

Too Many Fields

Another glaring issue was the sheer number of choices developers needed to make in order to fully reskin their game.

Problem #3:

No help for beginners

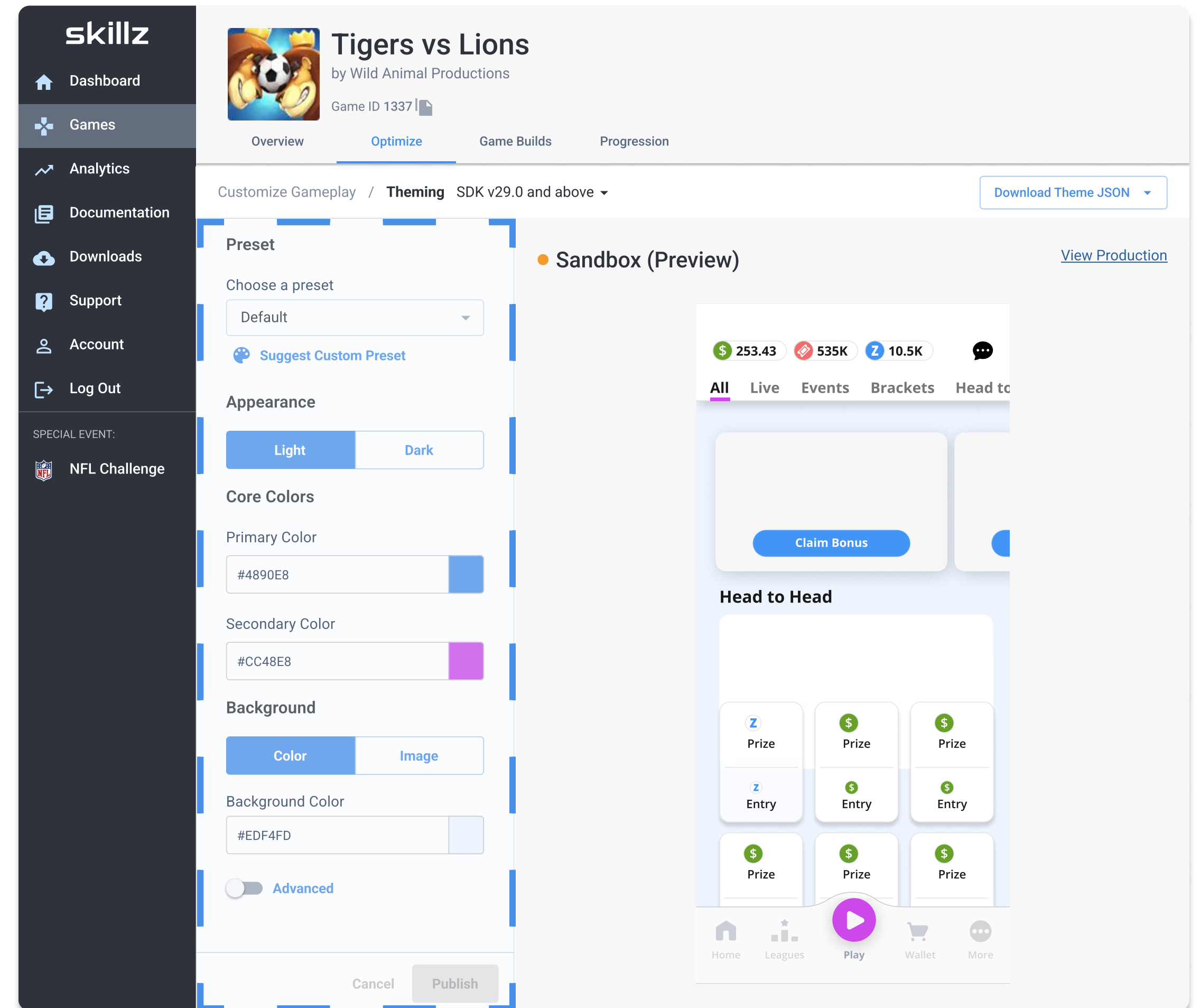
Making a game is already hard enough – doubly so for new developers who don't know how to best design their games.



SOLUTION

Reduced Fields

I also reduced the number of choices developers needed to make to just the **3 core colors** – the rest of the options would be auto generated based off of these selections. For more advanced developers that would want to override this selection, they could toggle on the advanced mode.



PROBLEMS

Problem #1:

No Visual Feedback

As developers updated their designs, there was no way to see what changes they actually made.

Problem #2:

Too Many Fields

Another glaring issue was the sheer number of choices developers needed to make in order to fully reskin their game.

Problem #3:

No help for beginners

Making a game is already hard enough – doubly so for new developers who don't know how to best design their games.



SOLUTION

Guided Color Recommendation Experience

There was also a new feature added that allowed developers to input their brand color, and get palette recommendations for their game.

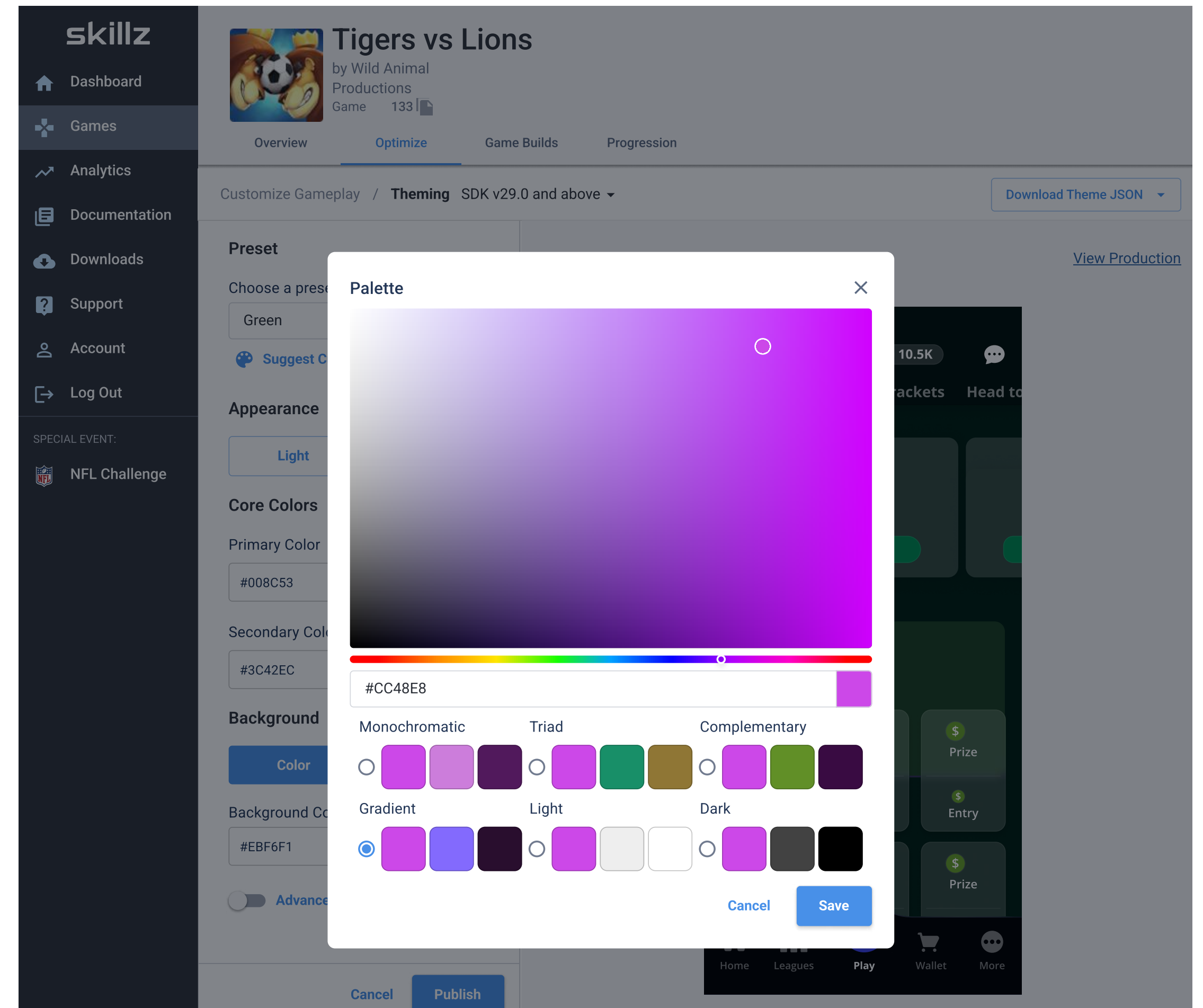


TABLE OF CONTENTS

Overview

Broader Initiatives

Research

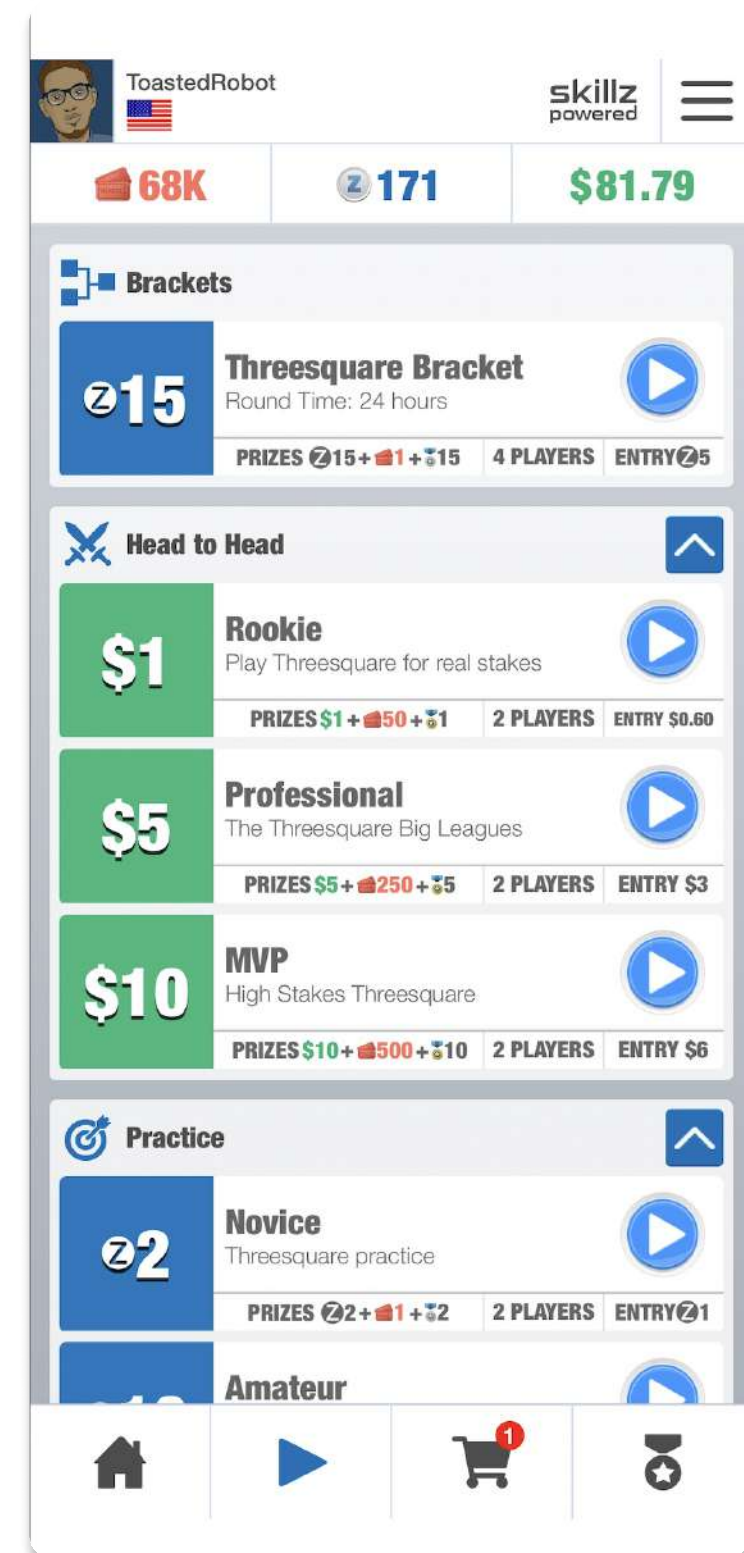
Design Process

Learnings

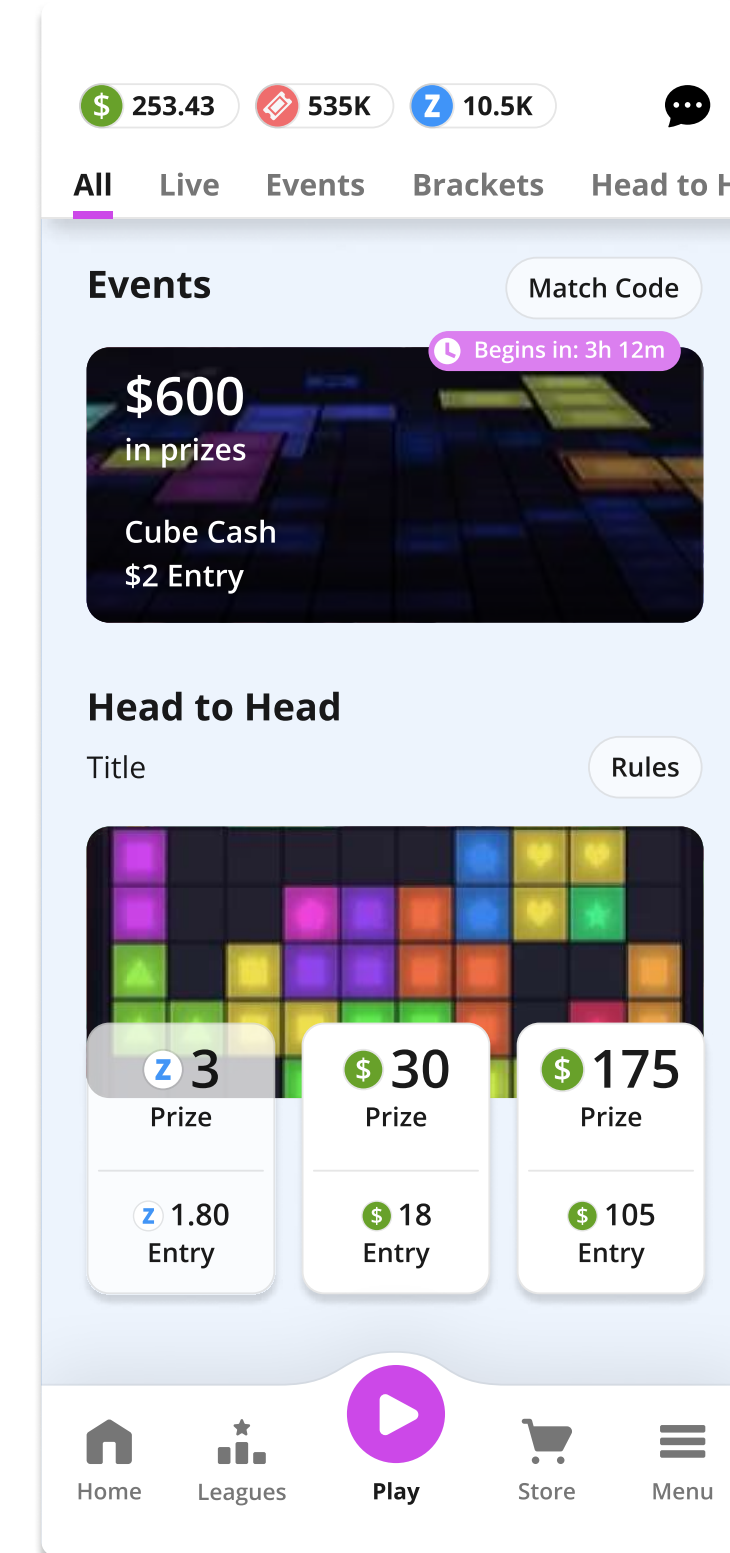


MOBILE DESIGN CHANGES

So you might've noticed the jump between mobile screens designs – that's because this project was part of a broader initiative to completely redesign the mobile SDK.



Before

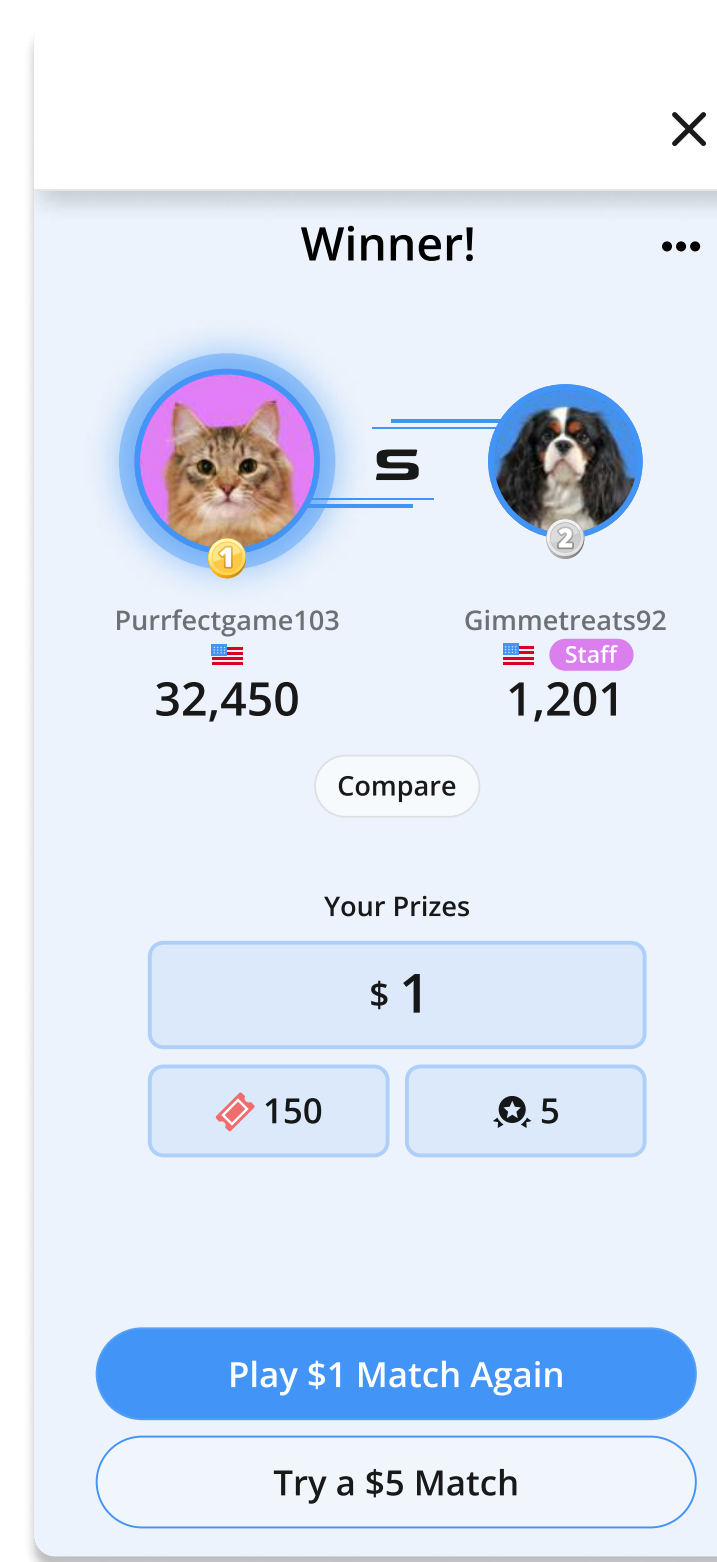
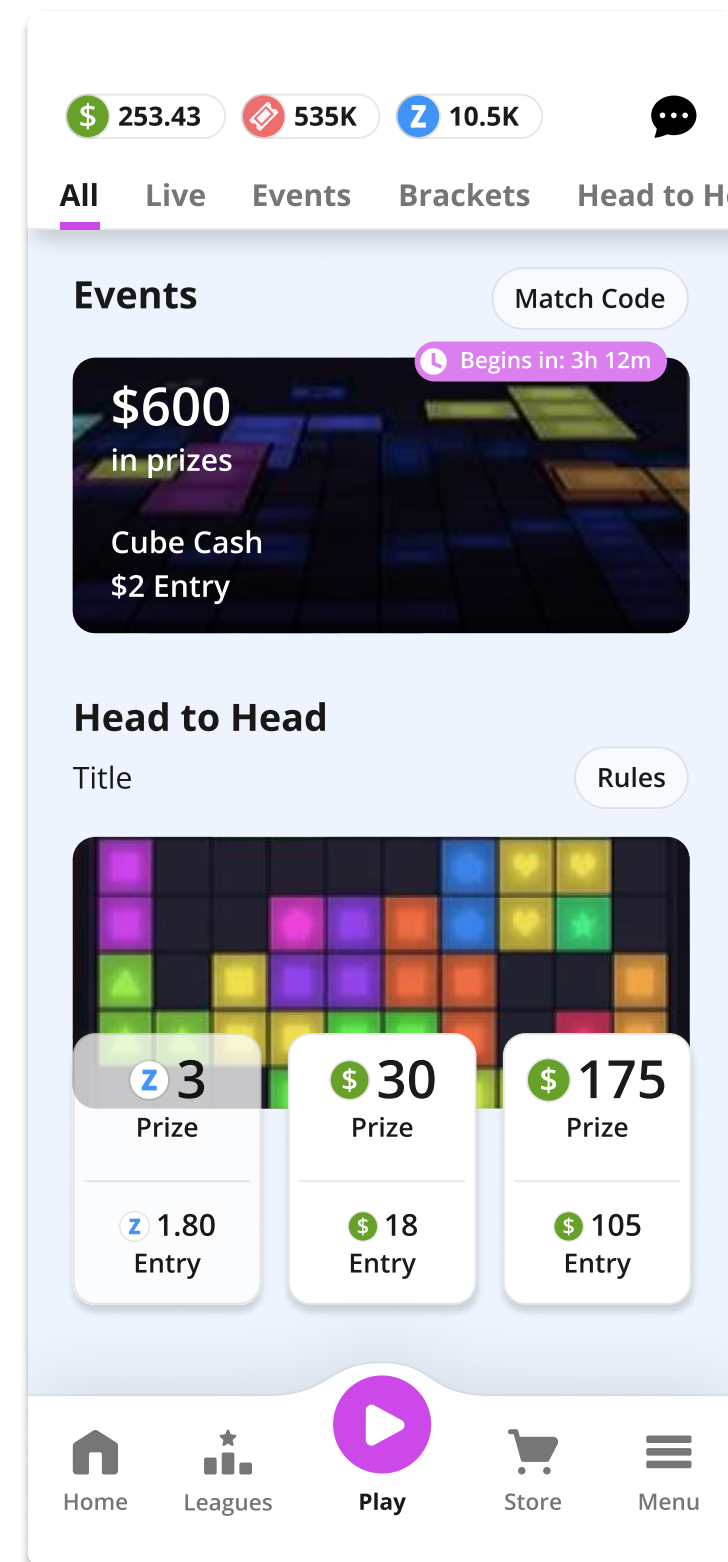


After

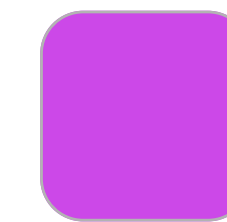


COLOR SYSTEM CHANGES

This effort involved completely changing how our mobile systems could be defined with just three colors: **Primary**, **Secondary**, and **Background**. The rest of the theme could then be inferred by the system to simplify the previous 50+ fields that needed to be edited.



Primary Btn Color



Secondary Btn Color



Labels (Auto Generated)
Step up from secondary



Main Background Color



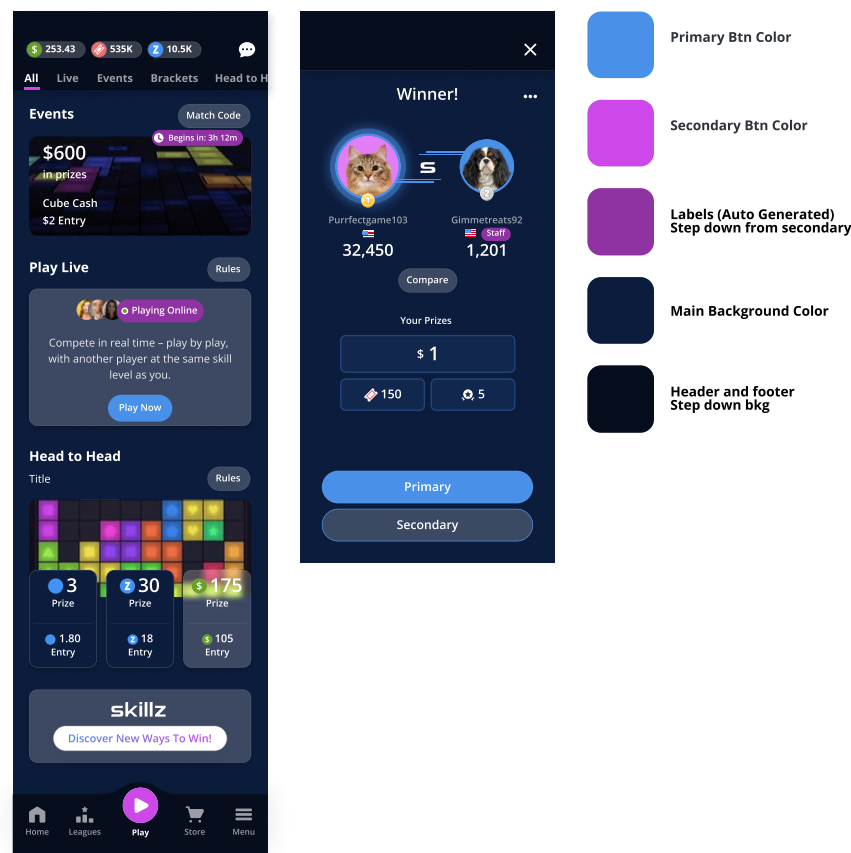
Header and footer
Step down bkg



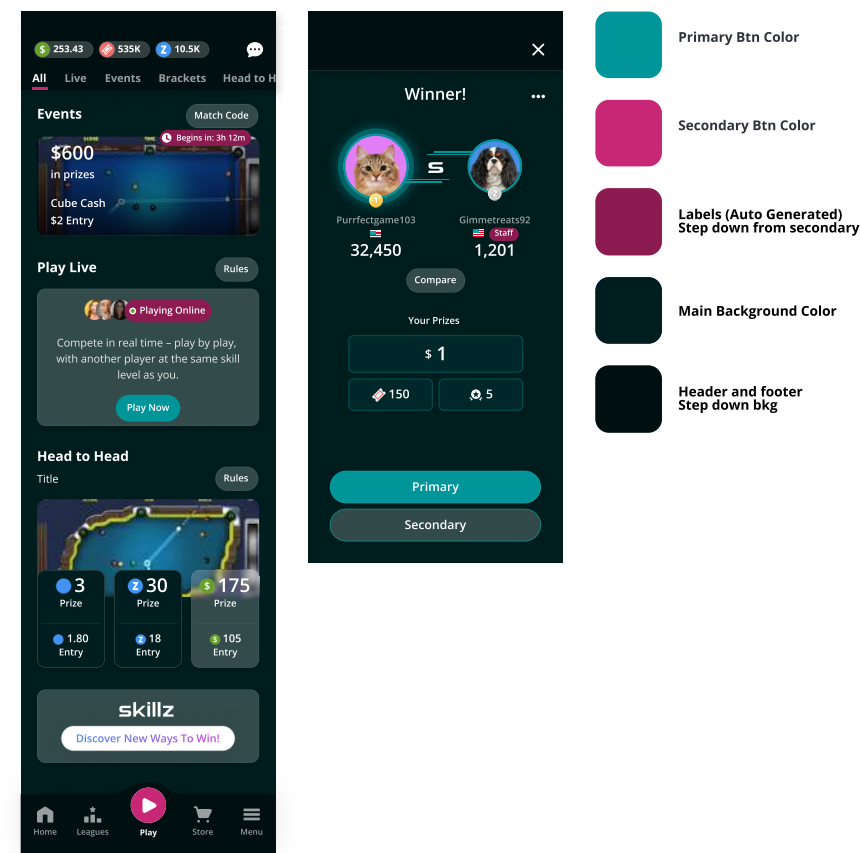
COLOR SYSTEM CHANGES

This part of the project involved Christa, the lead of the mobile design team, to help define out what the mobile designs needed to look like, and how it mapped to the new mobile designs.

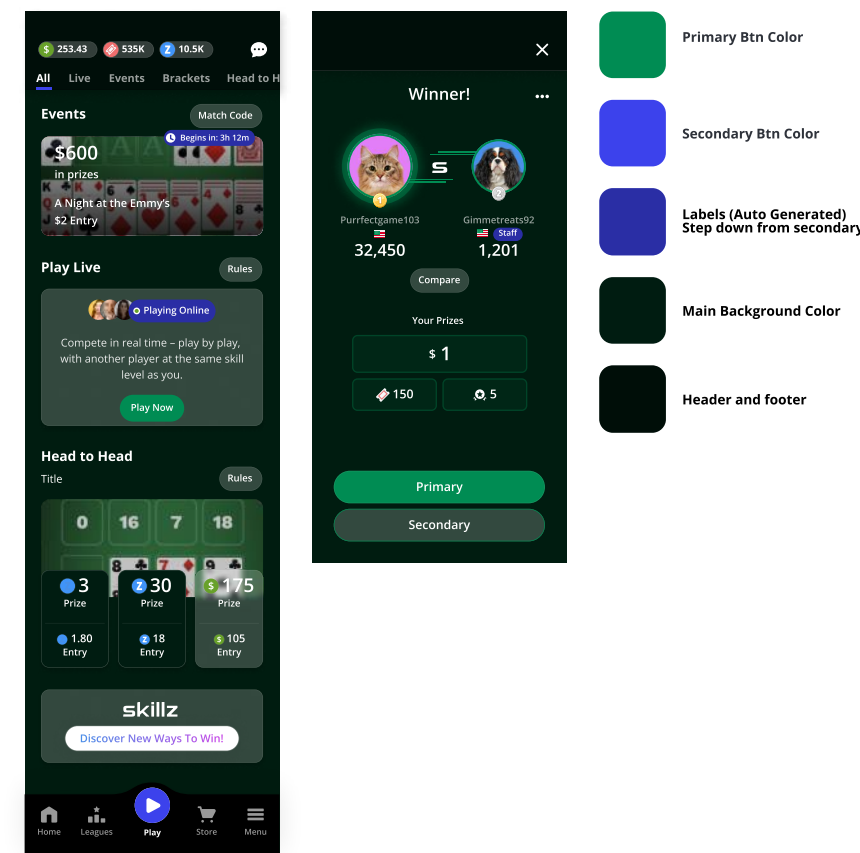
Default (Blue)



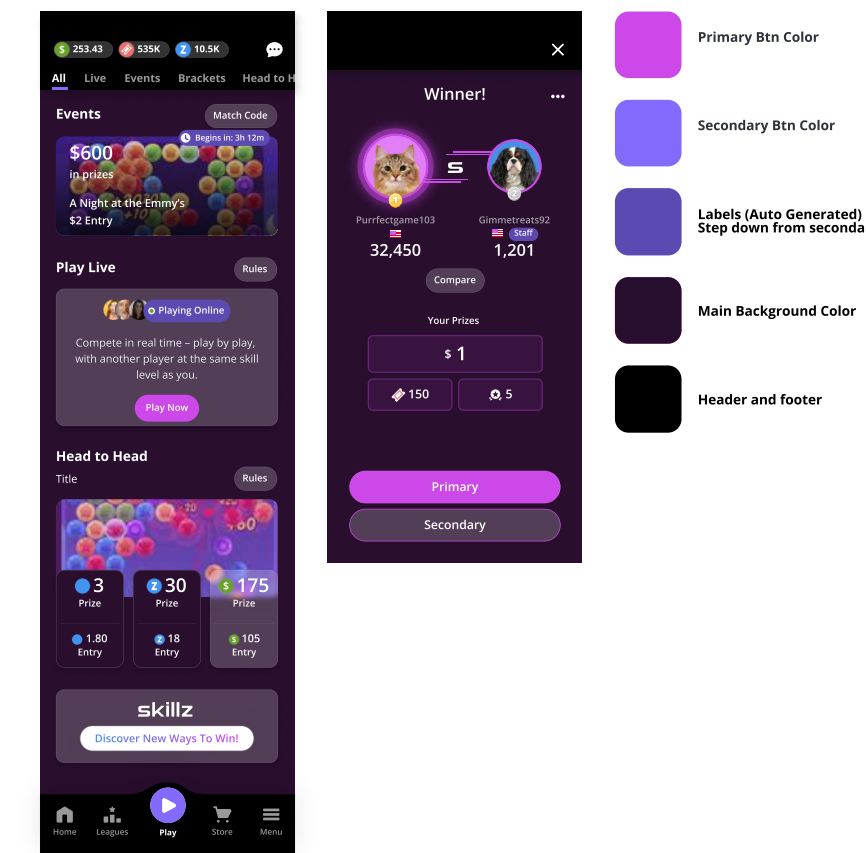
Teal



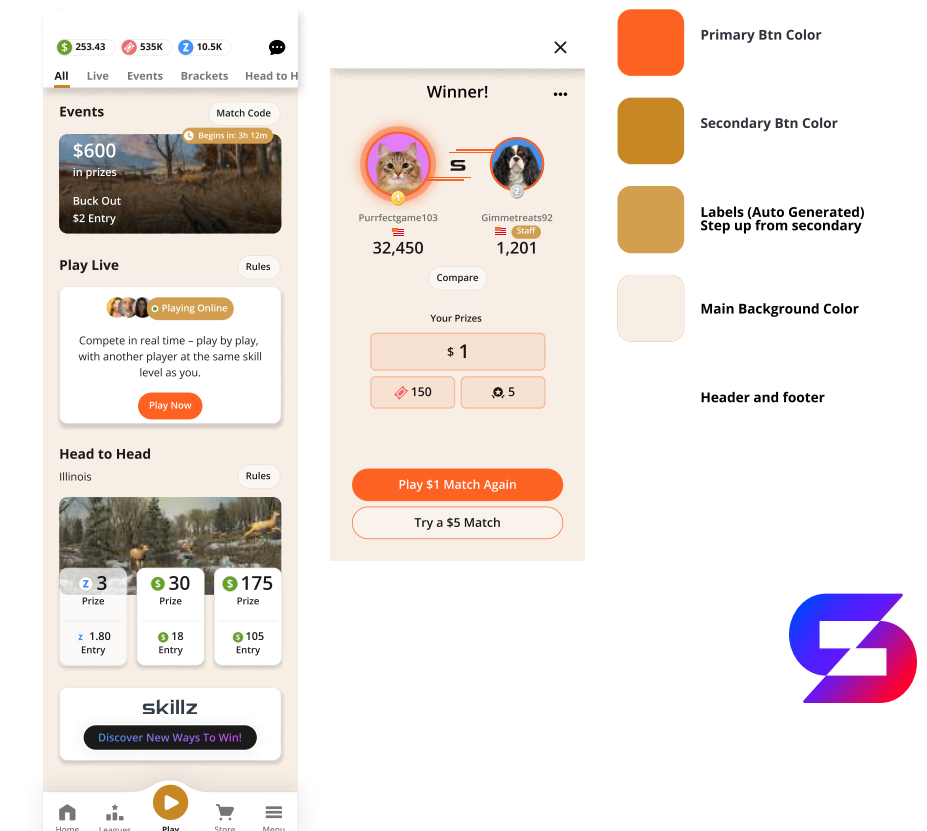
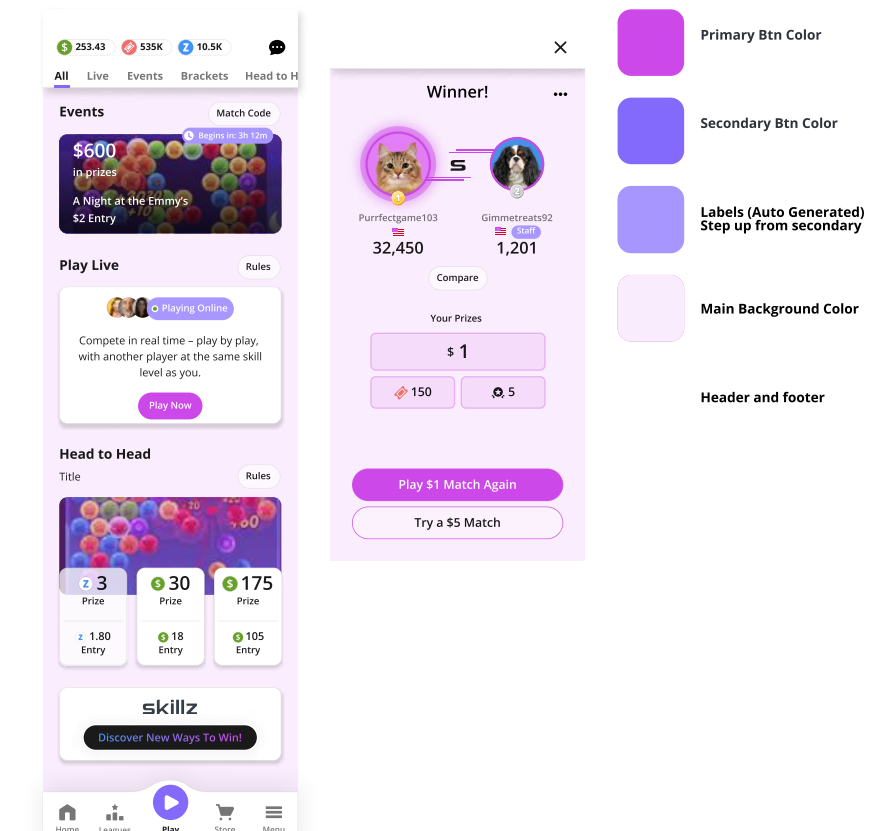
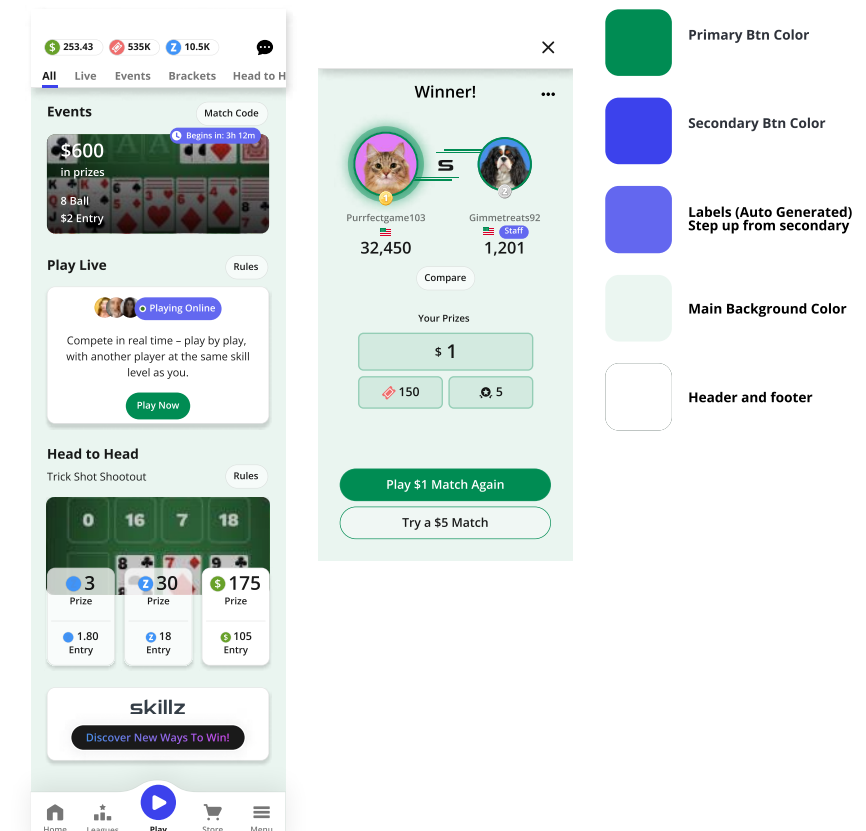
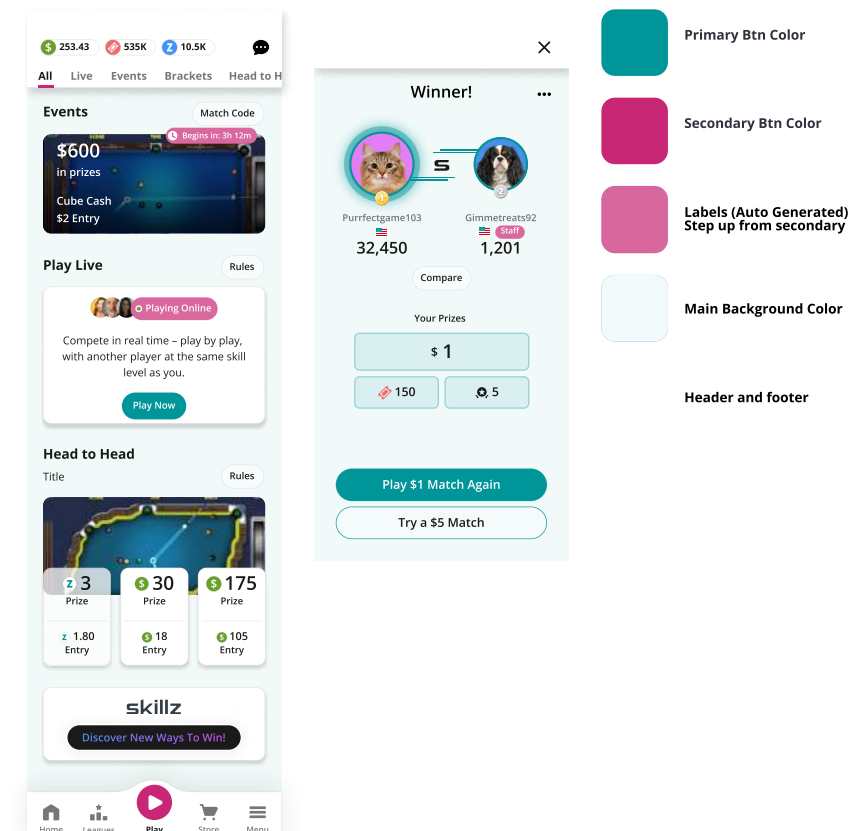
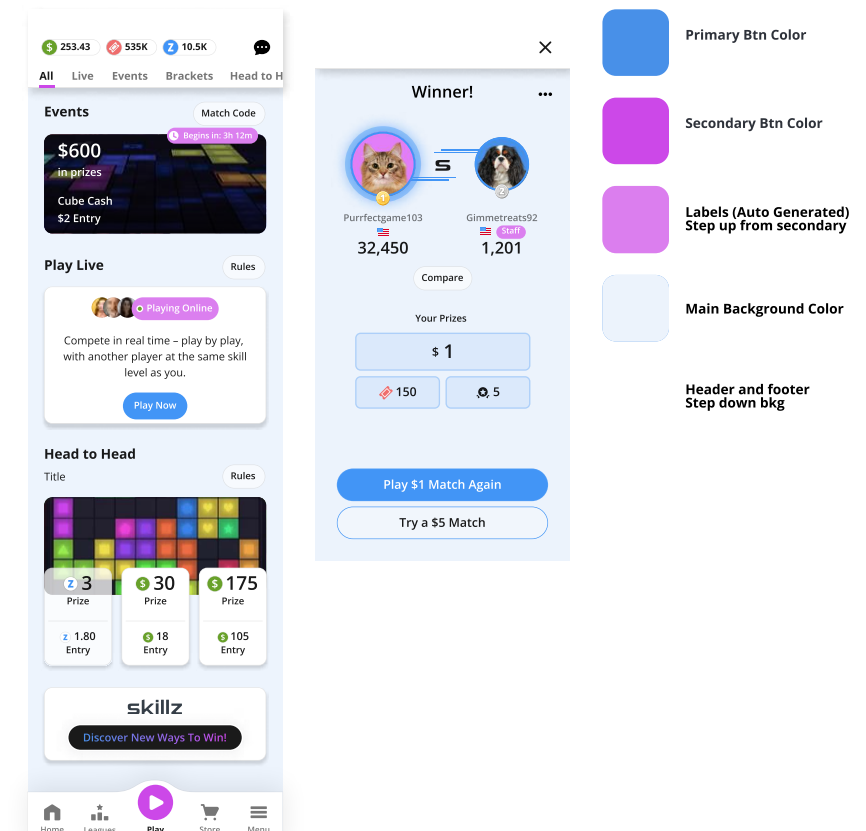
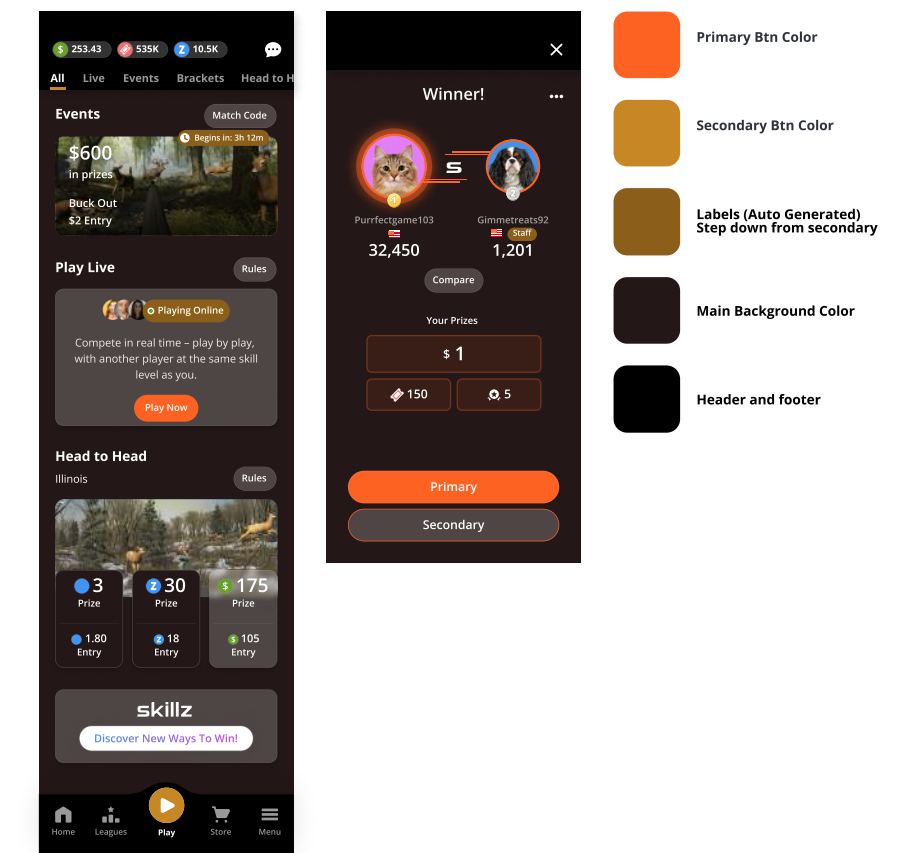
Green



Purple



Orange



TRANSLATING TO DEVELOPER EXPERIENCE

This involved mapping each of the current colors we used in the old color definition system and how it mapped to the code:

Core Colors	
<pre>"backgroundViewGradient": ["#ccd3da", "#ccd3da", "#a7bcd7"],</pre>	Background Gradient App Wide default background gradient if no background image is supplied (and while background image is loading)
<pre>"viewPrimaryColor": { "backgroundColor": "rgba(255, 255, 255, 1.0)" },</pre>	Main Color Primary view bkg color. Tab bar background
<pre>"opacityLevels": { "opacityPrimary": 0.9, "opacitySecondary": 0.8, "opacityTertiary": 0.8, "opacityNone": 1.0 },</pre>	Main Color Opacity Used where Opacity is required (will replace given opacities for colors, not multiply)
<pre>"viewDividerPrimaryColor": { "backgroundColor": "#cfd9df" },</pre>	Main Divider Color Used for background of pending/placeholder avatar, and a few different dividers (darker divider)
<pre>"viewSecondaryColor": { "backgroundColor": "rgba(255, 255, 255, .90)" },</pre>	Secondary Color Secondary view color, ie: Match Select cell background

Background Gradient

#ccd3da

App wide default background gradient if no background image is supplied (and while background image is loading)

Main Color

rgba(255, 255, 255, 1.0)

Primary view background color, ie: tab bar background.

Main Color Opacity

0.9

Used where opacity is required (will replace given opacities for colors, not multiply)

Main Divider Color

#cfd9df

Used for background of Pending/Placeholder Avatar, and a few different dividers (darker divider)

Secondary Color

rgba(255, 255, 255, .90)

Secondary view color, ie: Match Select cell background



ELIMINATING EXCESS FIELDS

And then remapping it to the new system, as well as deciding which fields should be editable, system generated but overridable, or not editable at all to the developers:

A Light

9:41 | 10,000 | 3.2M | 10.5K

All | Events | Play Live | Head to Head

Head to Head
Mexican Cantina | Rules

Prize: \$2, \$30, \$30 | Entry: 1, 18, 18

Regular Mix of Tiles | Rules

Prize: \$2, \$2, \$30 | Entry: 1, 1, 18

Prize: \$30, \$30, \$30 | Entry: 18, 18, 18

Home | Leagues | Play | Shop | More

Core Colors

Editable	Editable (System Generated)	Not Editable (System Generated)	Not Editable (Removed still in code)
Main Color Primary view bkg color. Tab bar background	Background Gradient App Wide default background gradient if no background image is supplied (and while background image is loading)	Main Divider Color Used for background of pending/placeholder avatar, and a few different dividers (darker divider)	Main Color Opacity Used where Opacity is required (will replace given opacities for colors, not multiply)
Secondary Color Secondary view color, ie: Match Select cell background		Secondary Divider Color Used for borders of avatars, flags, and checkboxes (lighter divider)	Secondary Color Opacity Used where opacity is required (will replace given opacities for colors, not multiply)
Tertiary Color Tertiary view background color, ie: Trophy Cell Background		Cell Background Color Primary cell background color	Tertiary Color Opacity Used where opacity is required (will replace given opacities for colors, not multiply)
		Profile Picture Background Color Used for Profile section icons	



END RESULT

By doing this exercise, in the designs I would later create, it helped the UX immensely by allowing developers to focus on those main 3 color choices, and allow the system to auto generate complementary colors and accents.

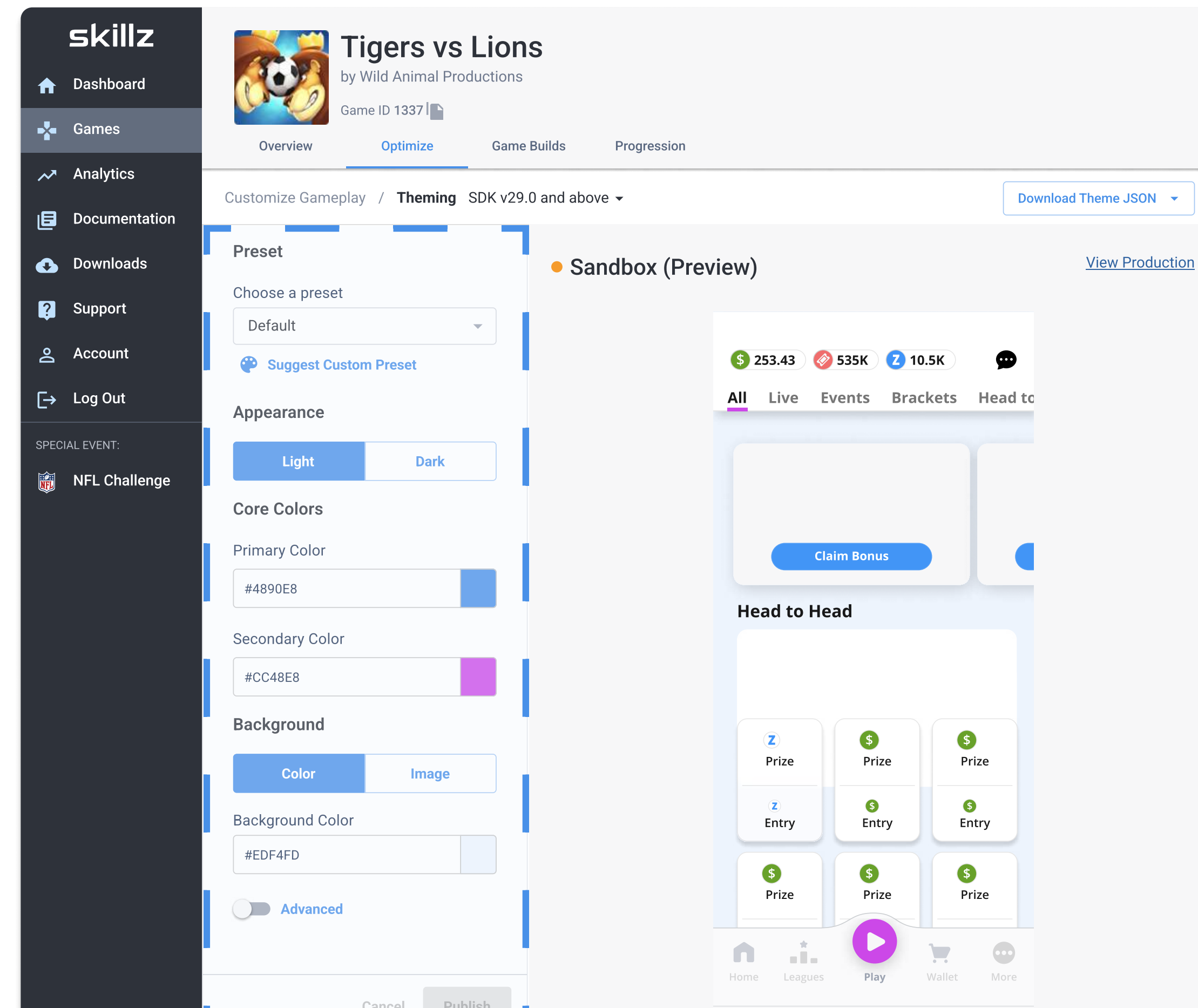


TABLE OF CONTENTS

Overview

Broader Initiatives

Research

Design Process

Learnings



PERSONAS

One of the biggest challenges with this project was the need to design for developers of different experience levels. After chatting with our developers there were 3 personas that surfaced.

Persona #1:

Beginner Indie Developer

For the beginner developer, there was a knowledge gap on how to get started with tackling their theme.

Persona #2:

Experienced Indie Developer

For developers who have previously already created games, there needed to be enough tools to bring their ideas to life, while being easy enough to use so they aren't spending too much time on retheming their game.

Persona #3:

Game Studio

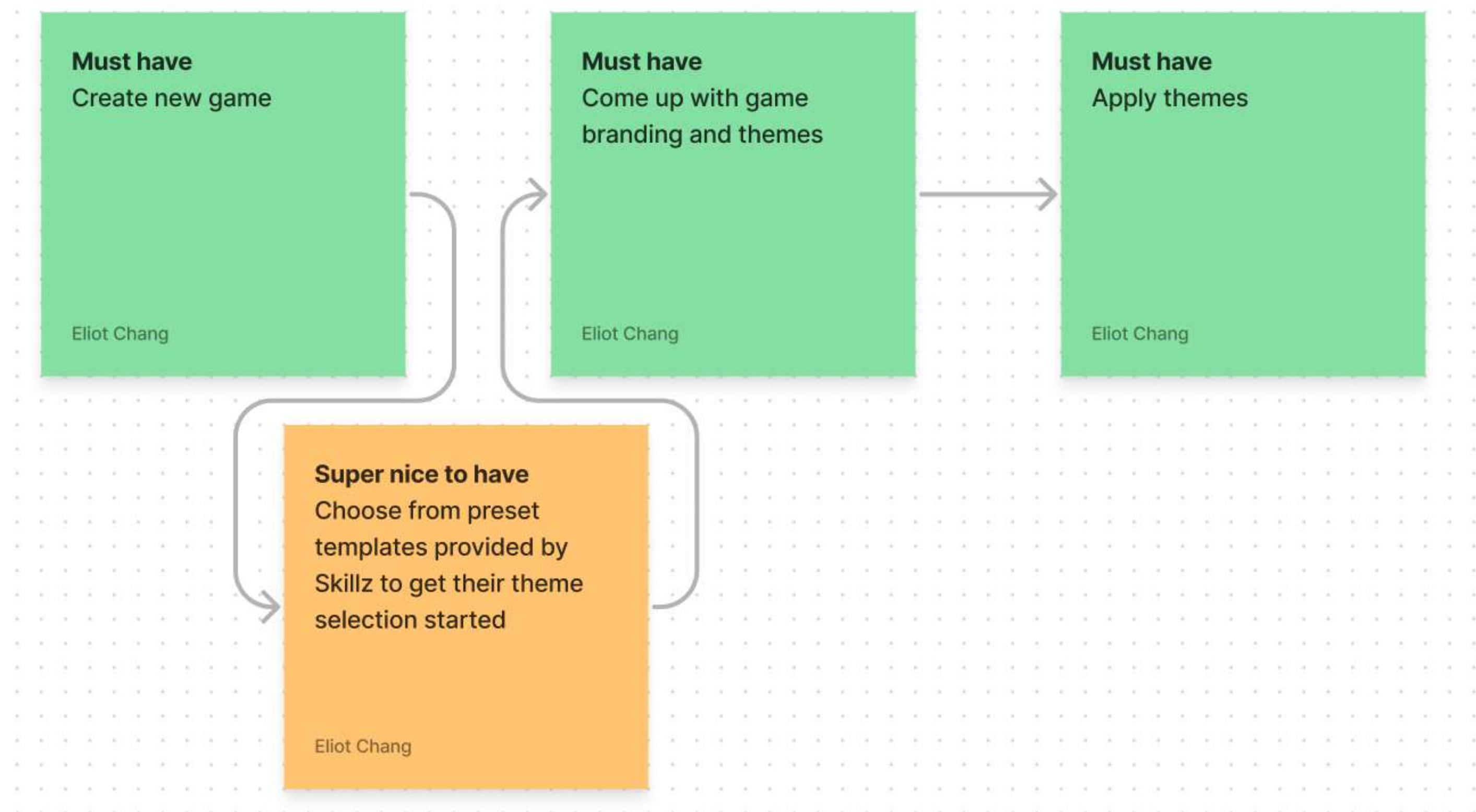
For game studios that might have someone dedicated to the branding and retheming of their games, there needed to be features that allowed them full control of exactly how their games could look and feel.



DEVELOPER JOURNEY

Beginner Indie Developer

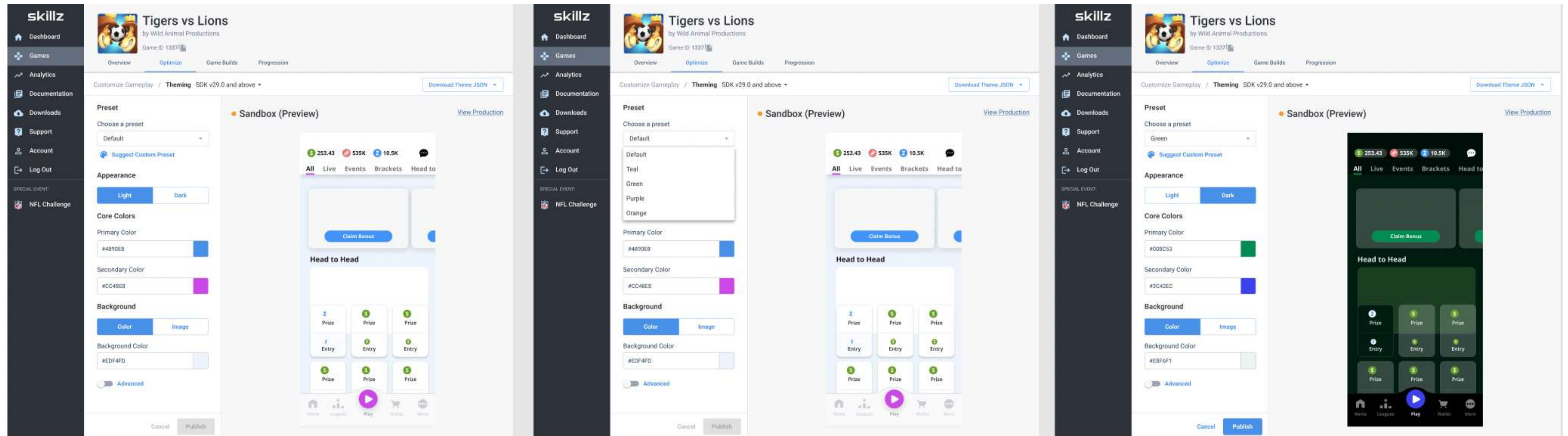
For people just dipping their toes into game development, there needed to be features that helped guide them in the initial color selection process, and the subsequent complementary colors to choose.



DEVELOPER JOURNEY

Preset Templates

For beginner developers, the solution for not having any idea where to start was creating presets for them to apply to their games.

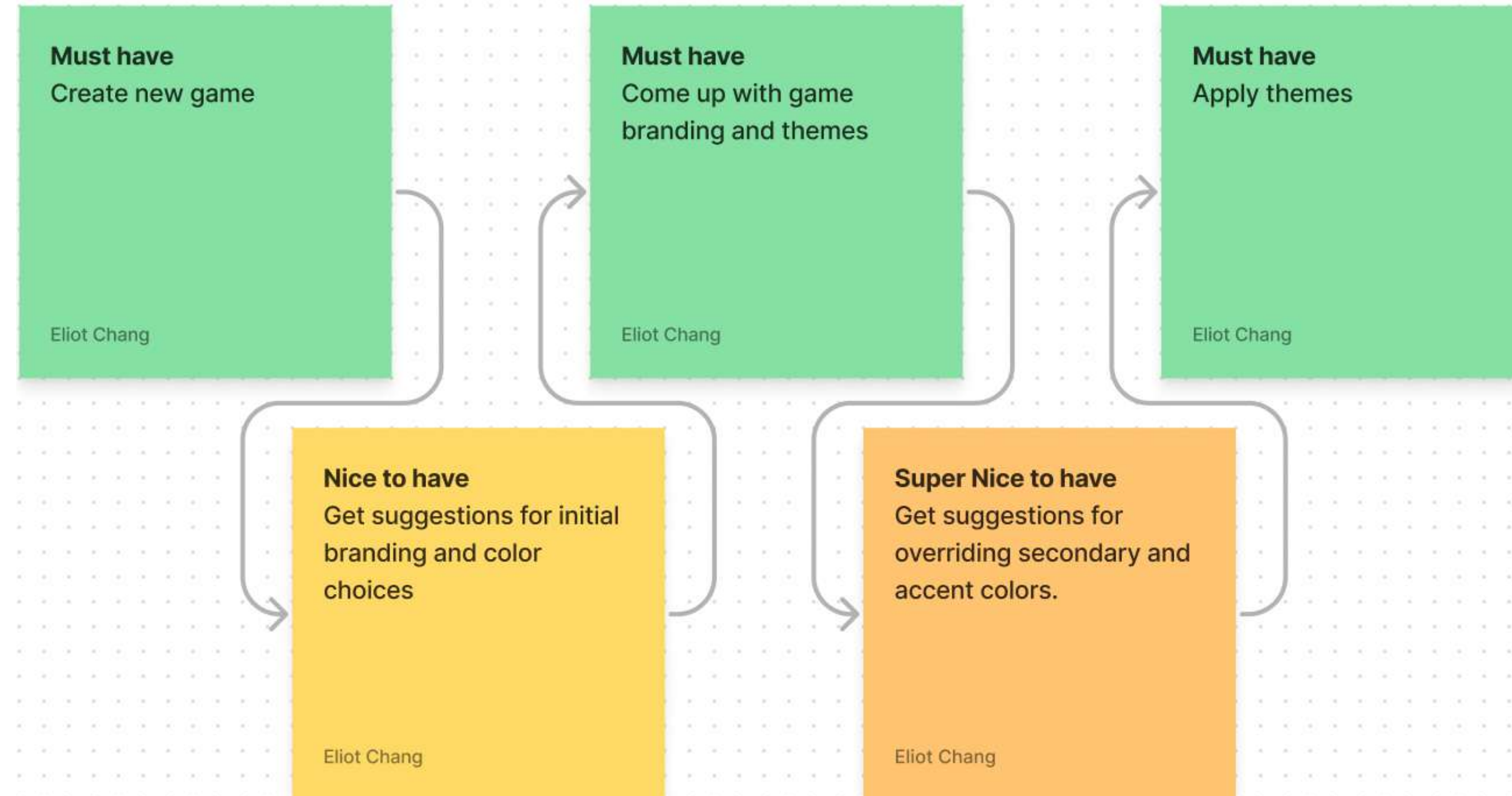


DEVELOPER JOURNEY

Experienced Indie Developer

On the other hand, for experienced indie developers who might already know what their initial brand colors are, this feature wouldn't be as useful.

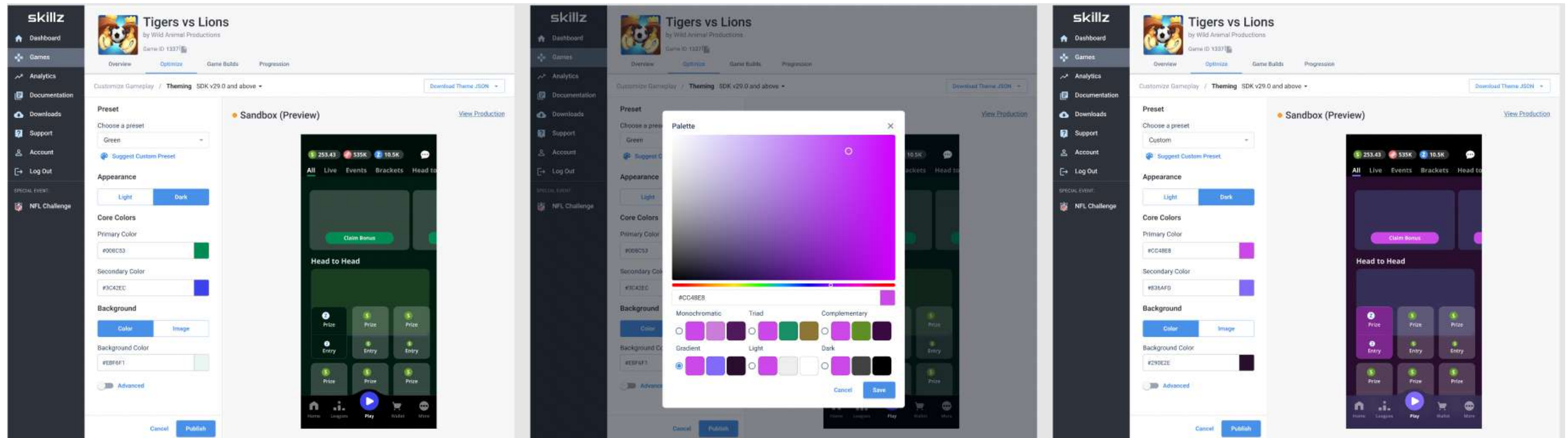
However, for these developers who might not know what colors best complement their theme, having a feature that suggests secondary and accent colors would be very useful.



DEVELOPER JOURNEY

Suggesting Secondary + Accent Colors

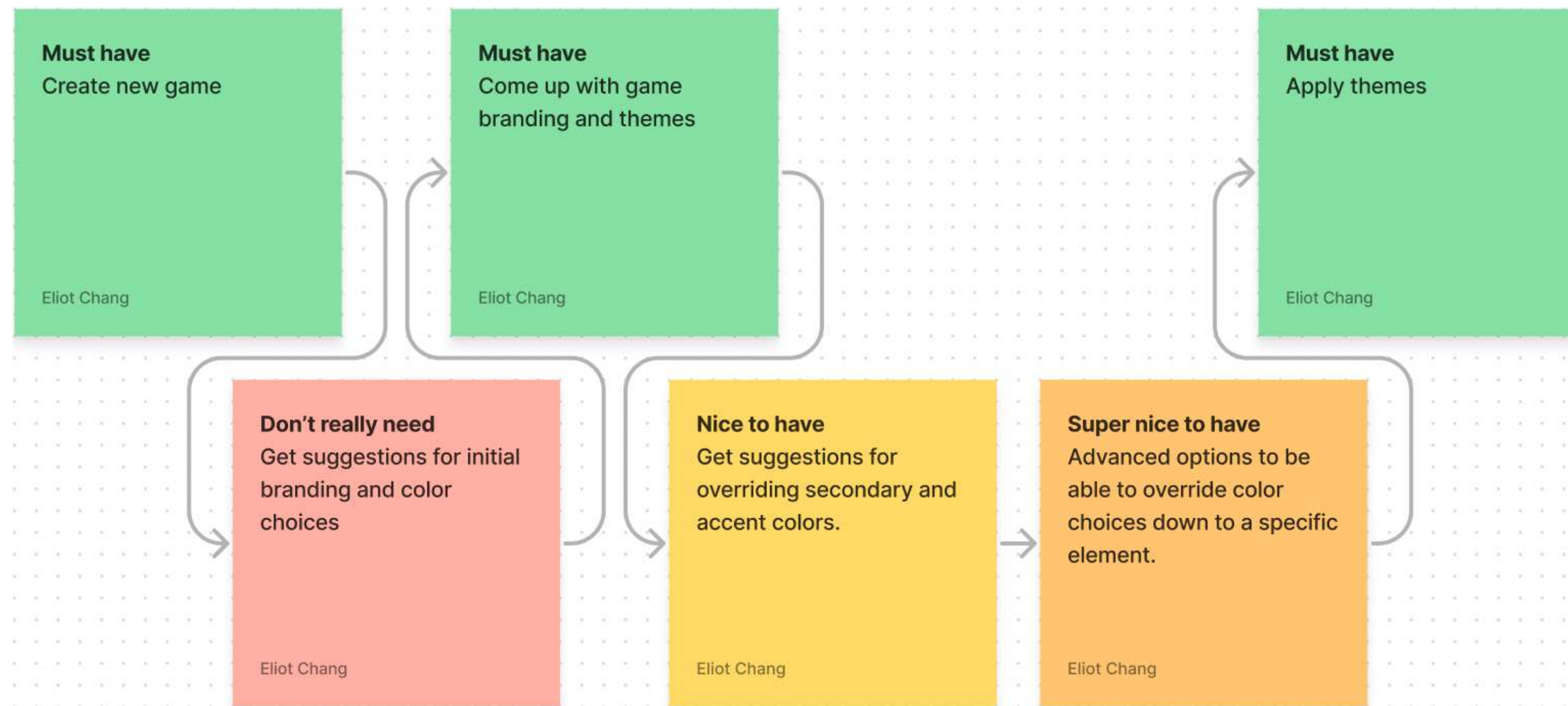
For experienced developers, the solution for suggesting secondary and accent colors was a palette recommendation tool built into the experience



Game Studios

For game studios who already know the exact branding they want, neither of the recommendation systems would be super useful to them.

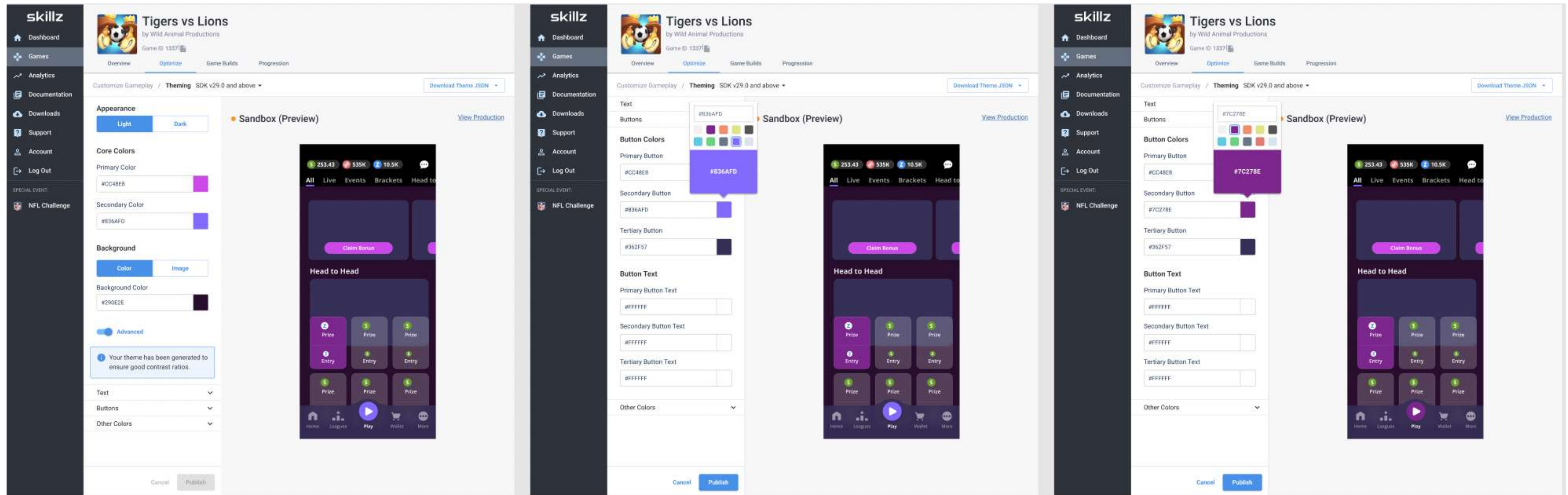
However, because they already fully fleshed out their games' designers, we need to have a feature focused on an advanced feature that allows overriding system generated suggestions.



DEVELOPER JOURNEY

Advanced overrides

For game studios, the solution for giving them free rein of their color choices was to create an advanced section where they could override the individual elements.



DEVELOPER JOURNEY

Notes on Advanced overrides

It should be noted that the advanced override section was tucked away because for beginner developers there is the headache of creating contrast issues on their games that we wanted to reduce.

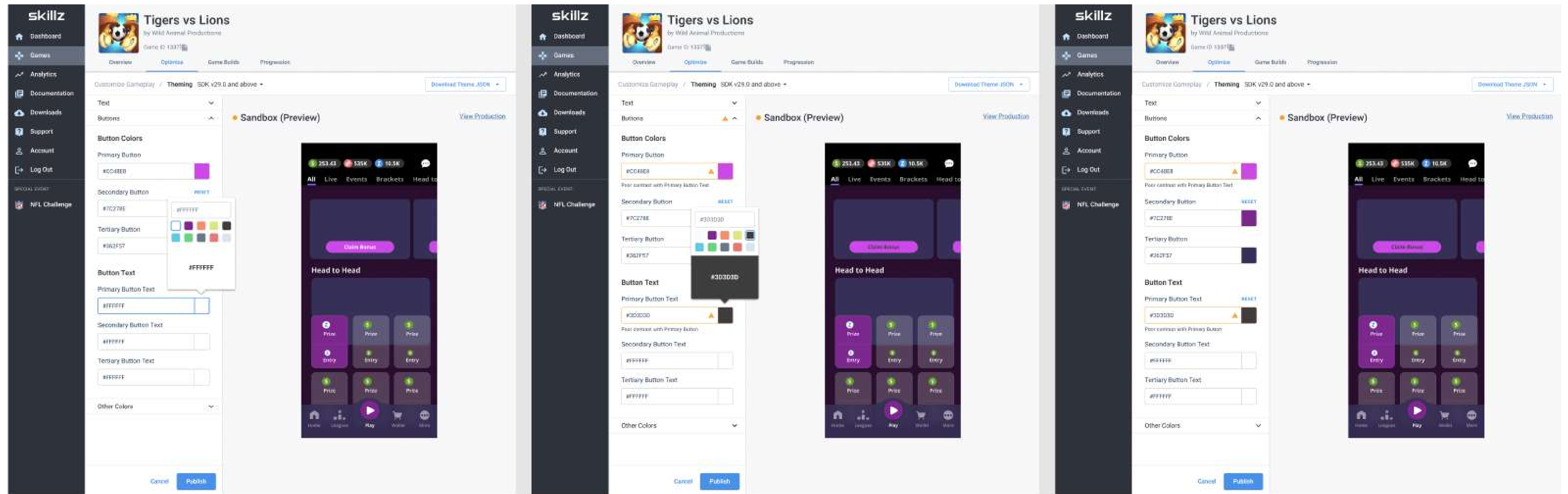


TABLE OF CONTENTS

Overview

Broader Initiatives

Research

Design Process

Learnings

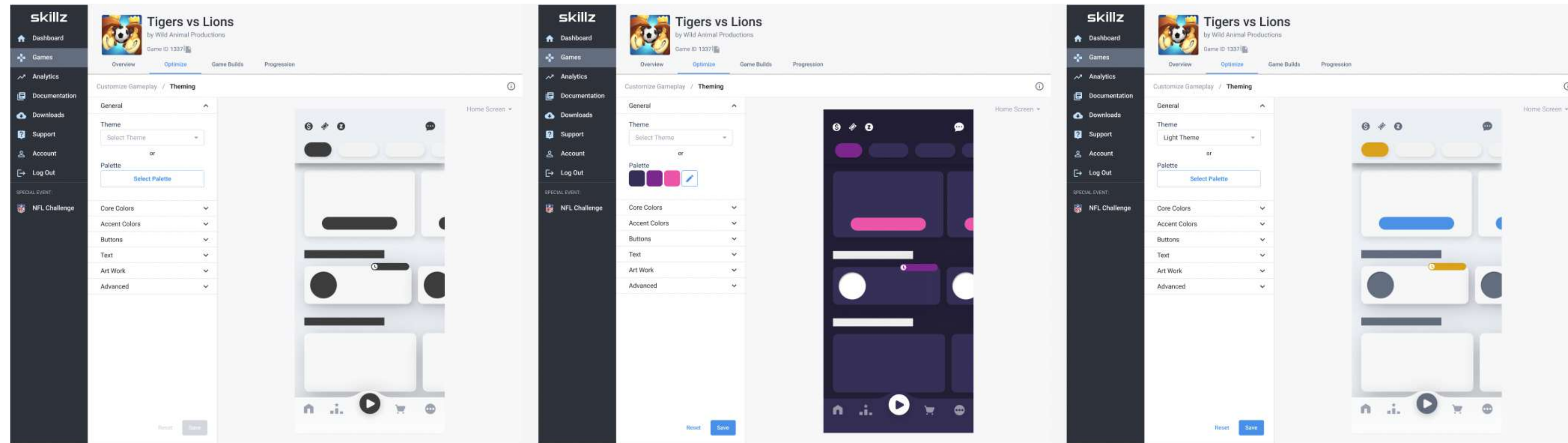


DESIGN PROCESS

Original Designs

From the get go I wanted to create a preview for the redesign – without it is would be impossible to do quick iterations.

The biggest changes in the design can be seen on the left hand editing section where the original designs were just a list of categories to choose from, without too much thought put behind the different experiences for beginner vs advanced developers.



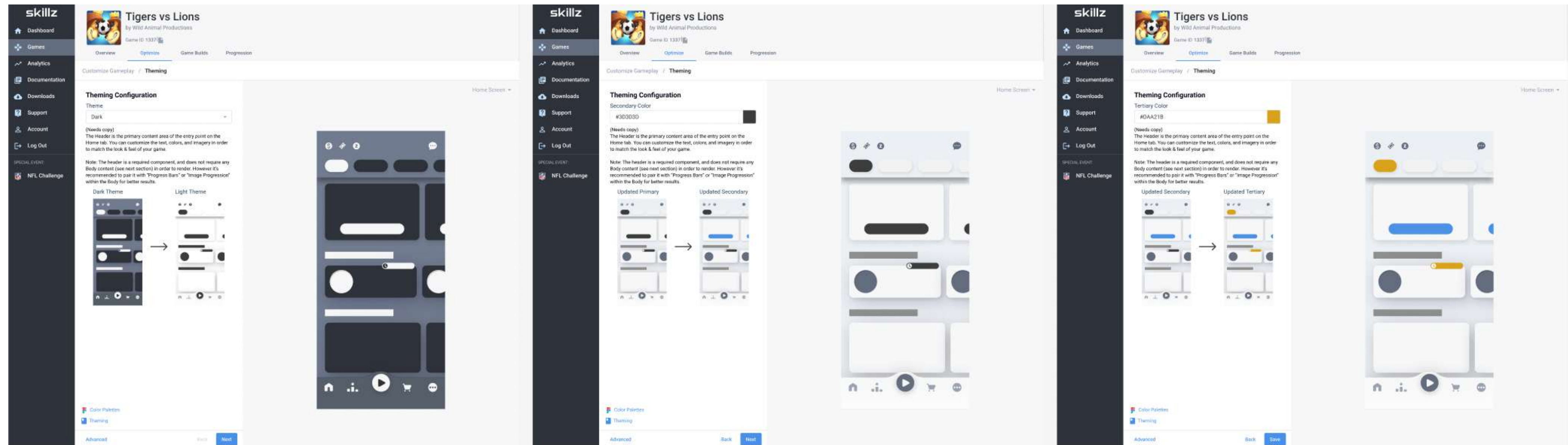
DESIGN PROCESS

Guided Experience

The next iteration I worked on focused on having a guided experience throughout the redeeming journey.

This would take you step-by-step through the flow as you added on each layer of color.

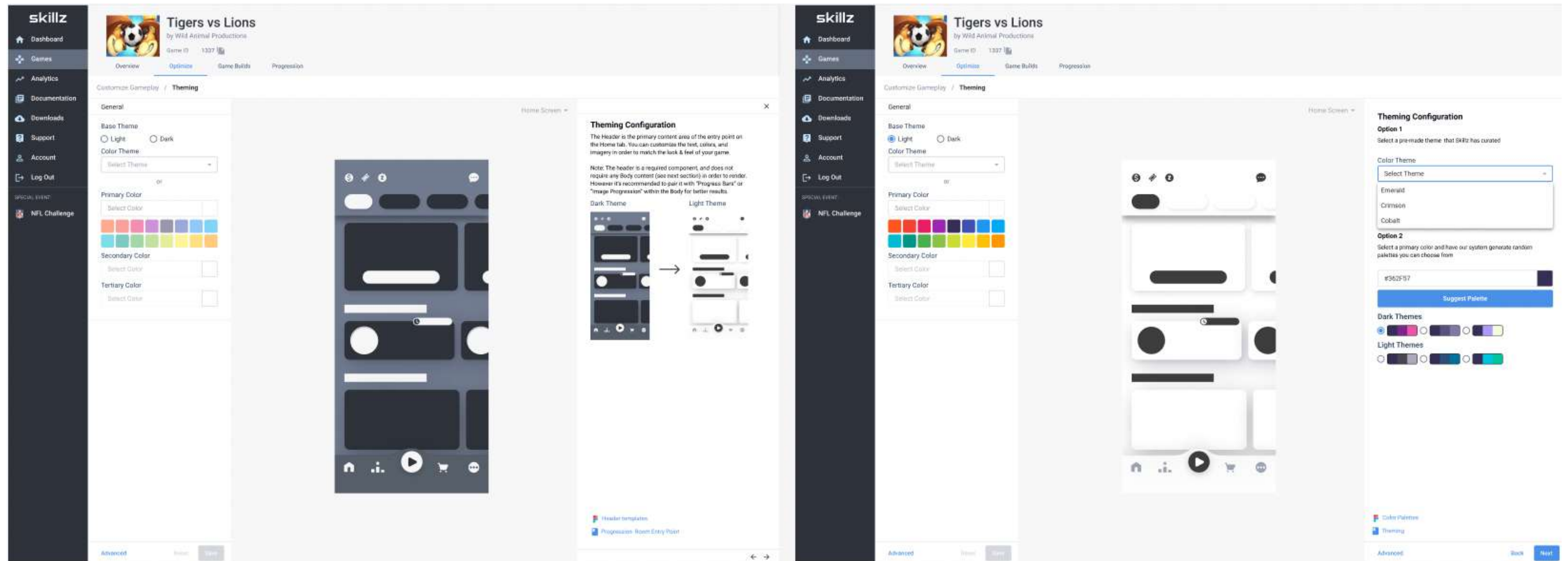
The issue with this design was it accommodated primarily for the beginner developer and the first time redesign experience. It largely ignored features for advanced developers and developers looking to retheme a pre-existing game.



DESIGN PROCESS

Guided Experience II

The second iteration of the guided experience had it optionally appear on the right-hand side, but this resulted in using the help bits as a crutch and the UX itself starting to get less intuitive at a glance.



DESIGN PROCESS

Advanced Tabs

This iteration of the designs focused on a tabbed experience for the advanced mode. I ended up deviating away from this design since the prominence of the tabs made developers more likely to explore the advanced section before they were ready to apply the advanced options.

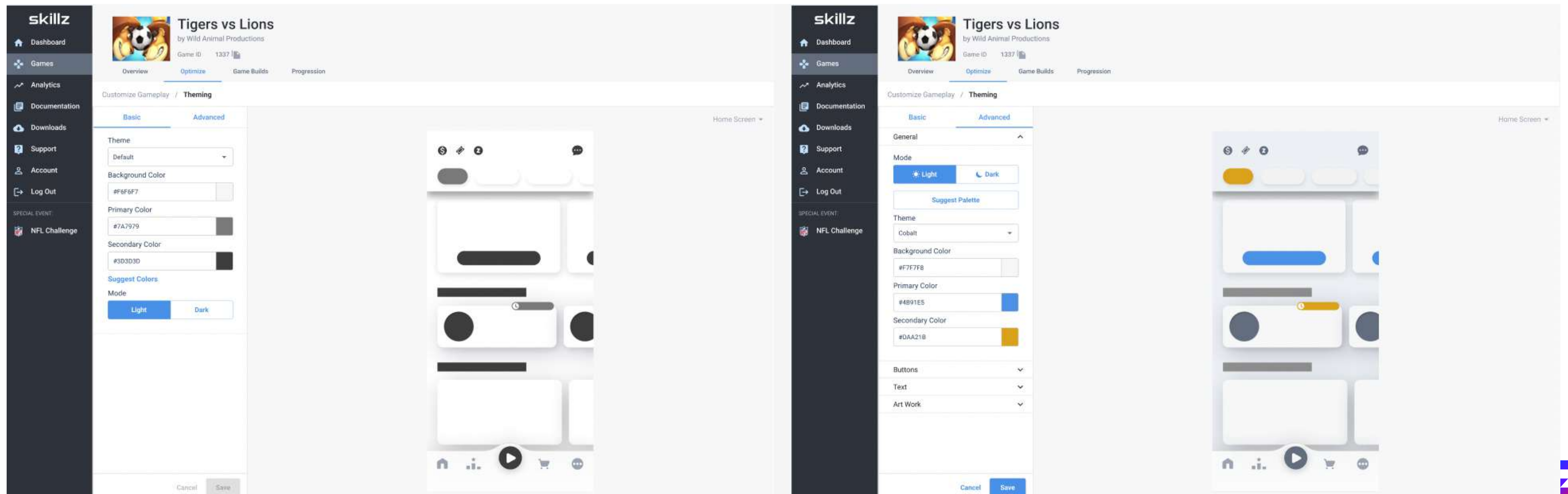


TABLE OF CONTENTS

Overview

Broader Initiatives

Research

Design Process

Learnings



LEARNINGS

Lesson #1:

One size fits all tradeoffs

The balancing act of making an experience easy enough for anyone to use, but flexible enough for game development veterans to have full control of their designs was tough to manage - but not impossible to pull off.

Lesson #2:

Focusing on the big picture

My initial hiccups with the design process for this project was focusing too much on selecting individual colors. Once I saw the color application process as an entire system, it was much easier to conceptualize how to tackle the problem.

Lesson #3:

Creating complex experiences

This was my first project working on a design with this level of complexity and showed me the number of edge cases needed to be considered to roll out a feature like this.

