

Free-to-play: A Shiny Case Study from Game Analytics

Hazel Kavılı

Wixot



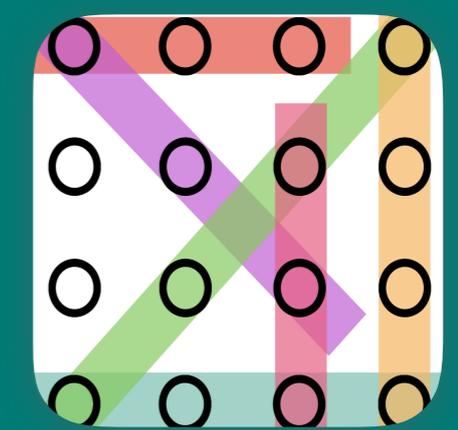
Szia!



- Data Analyst at Wixot
- PhD Student at ITU, Earth System Sciences
- Studied Statistics (BS and MS) at YTU
- Founder of R-Ladies Istanbul
- NASADatanaut, class 2017

Wixot

- Mobile game company with **more than 10M** players
- Experienced in making word search and word puzzle games

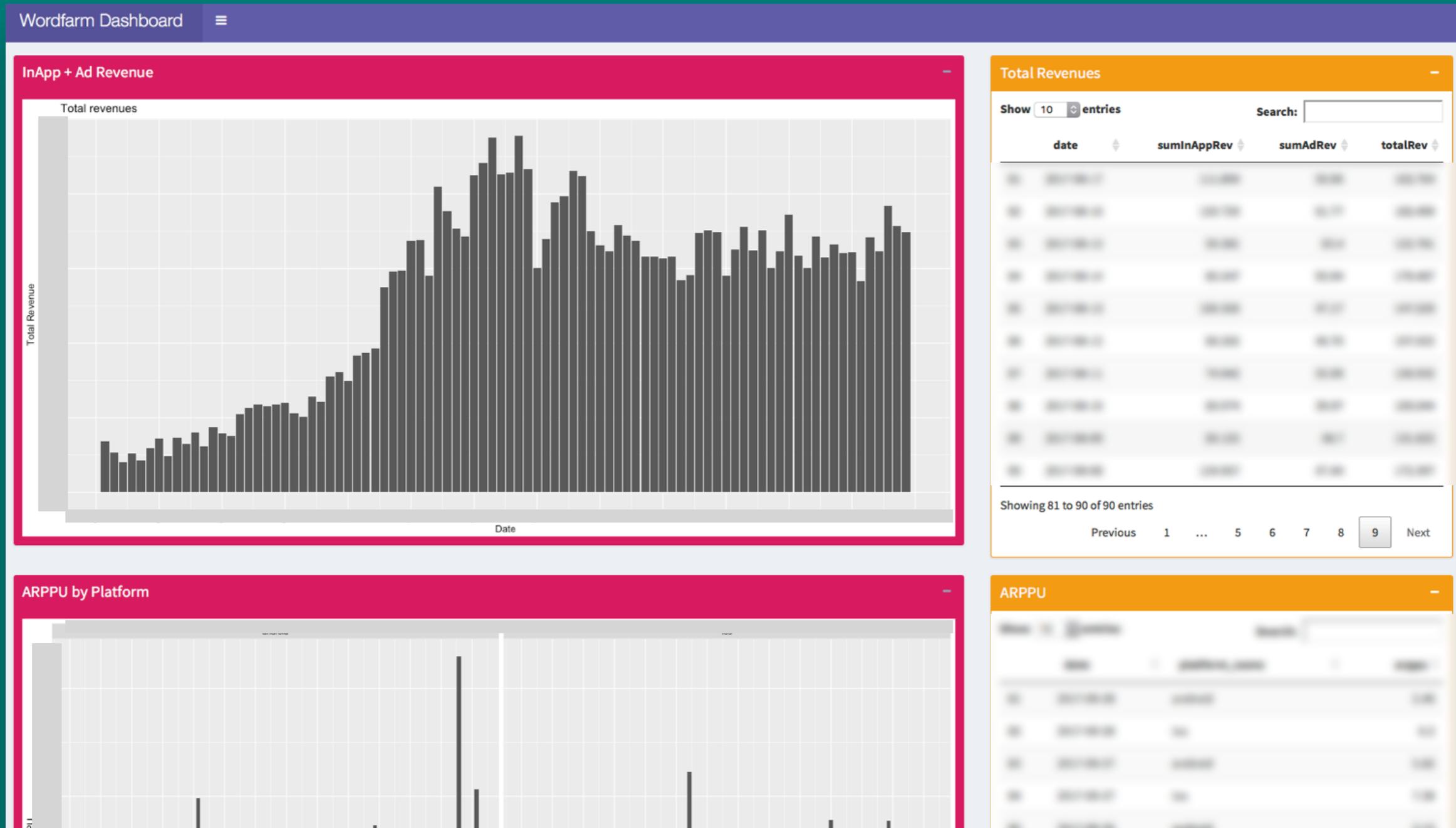


Word Farm



- Word puzzle game
- Free to play
- A lot of metrics that we wanted to track
- Goal: **better retention** and **better conversions**

Word Farm Dashboard



What is Game Analytics?

Part 1:

4 Challenges

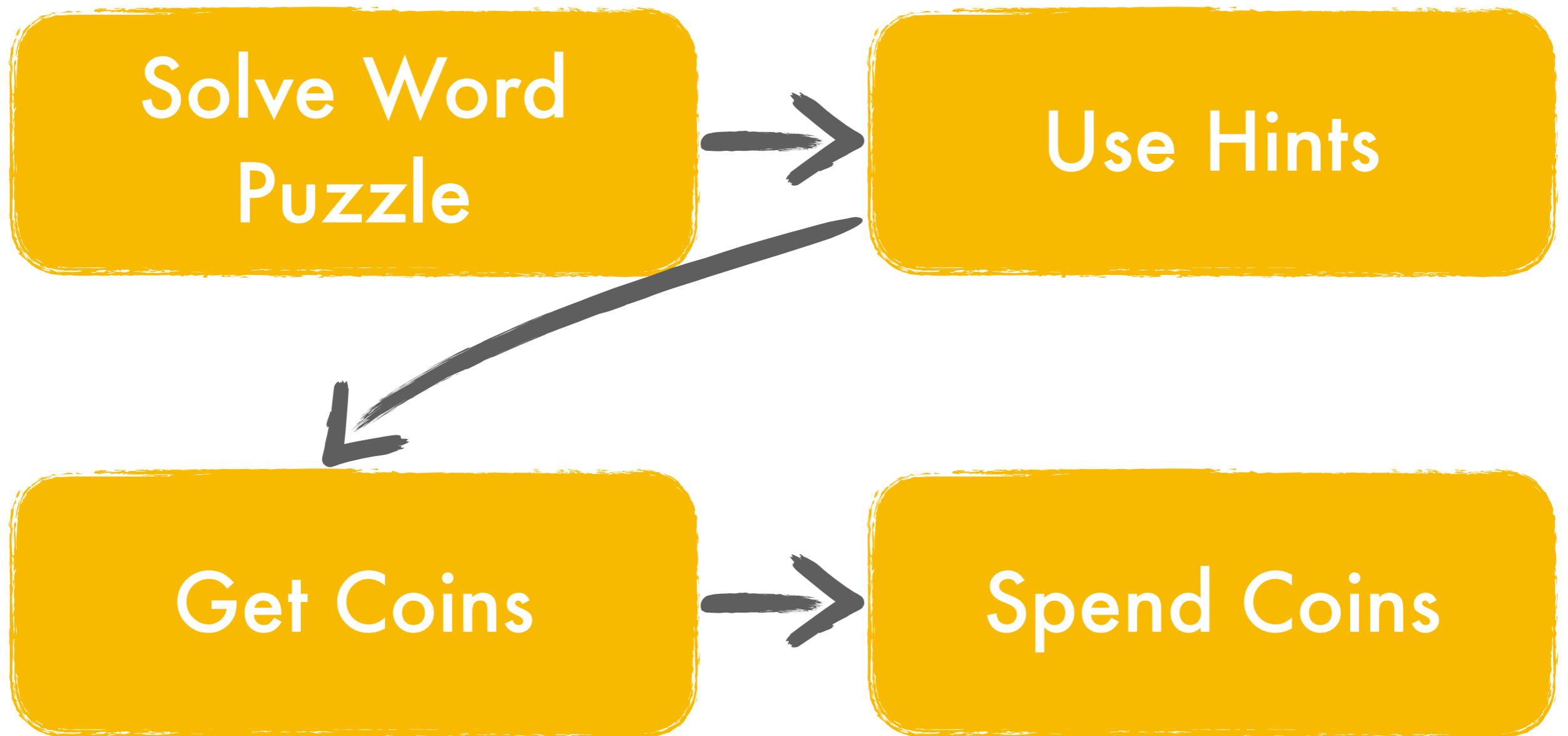
Part 2:

The Shiny Part

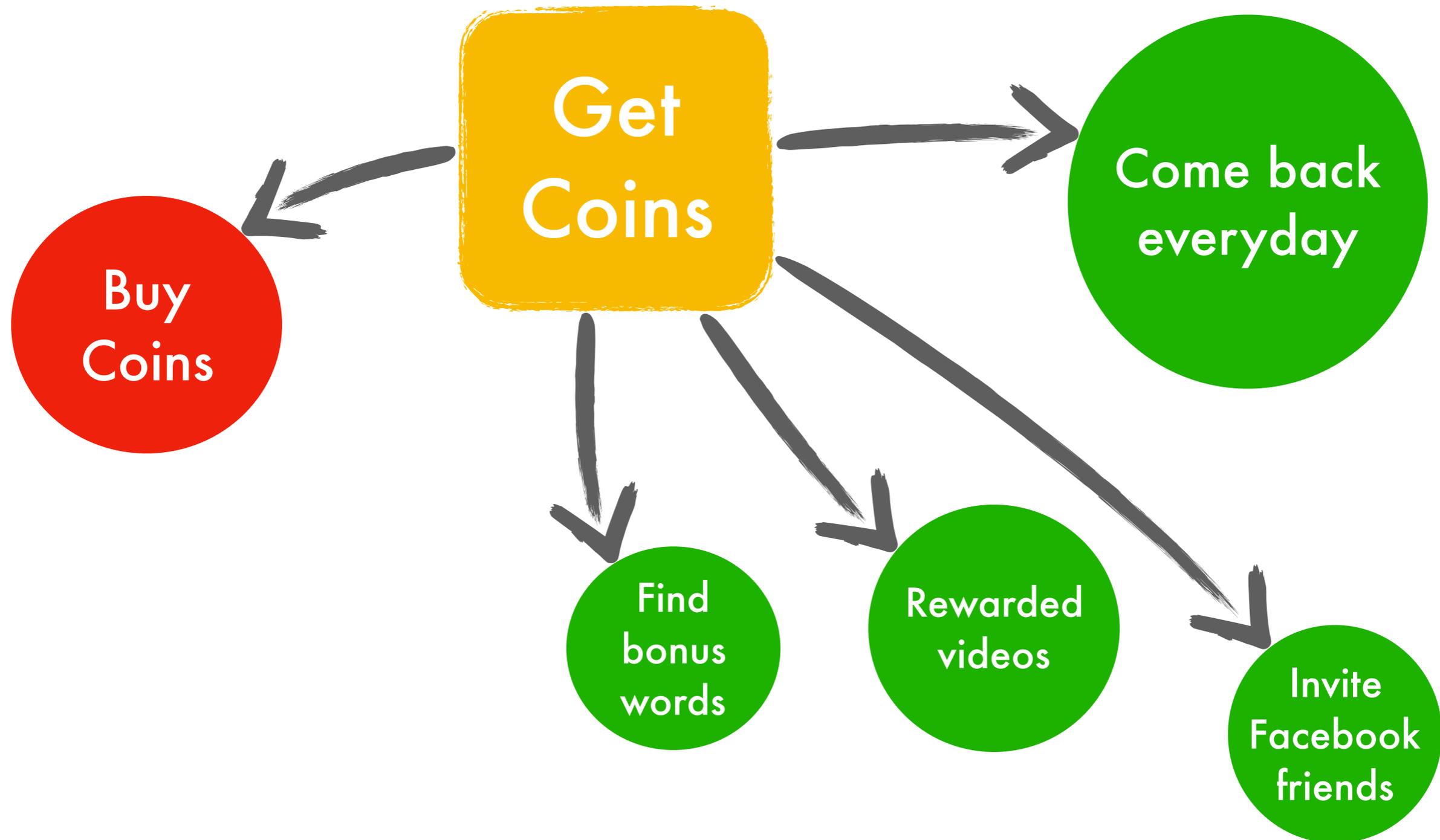
Freemium

- Game is available for **free**
- Players can buy **virtual goods** with **real money**

Freemium



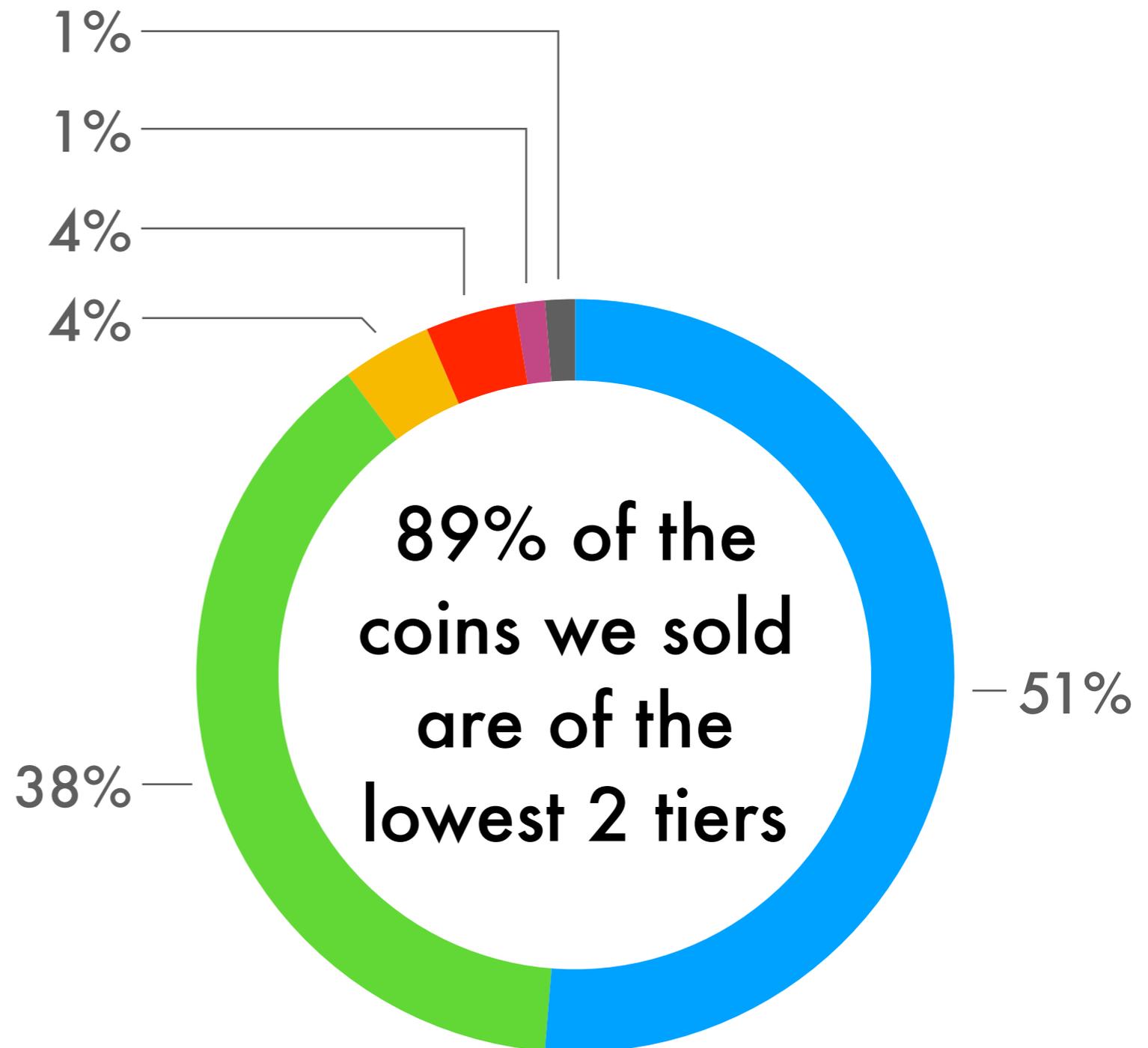
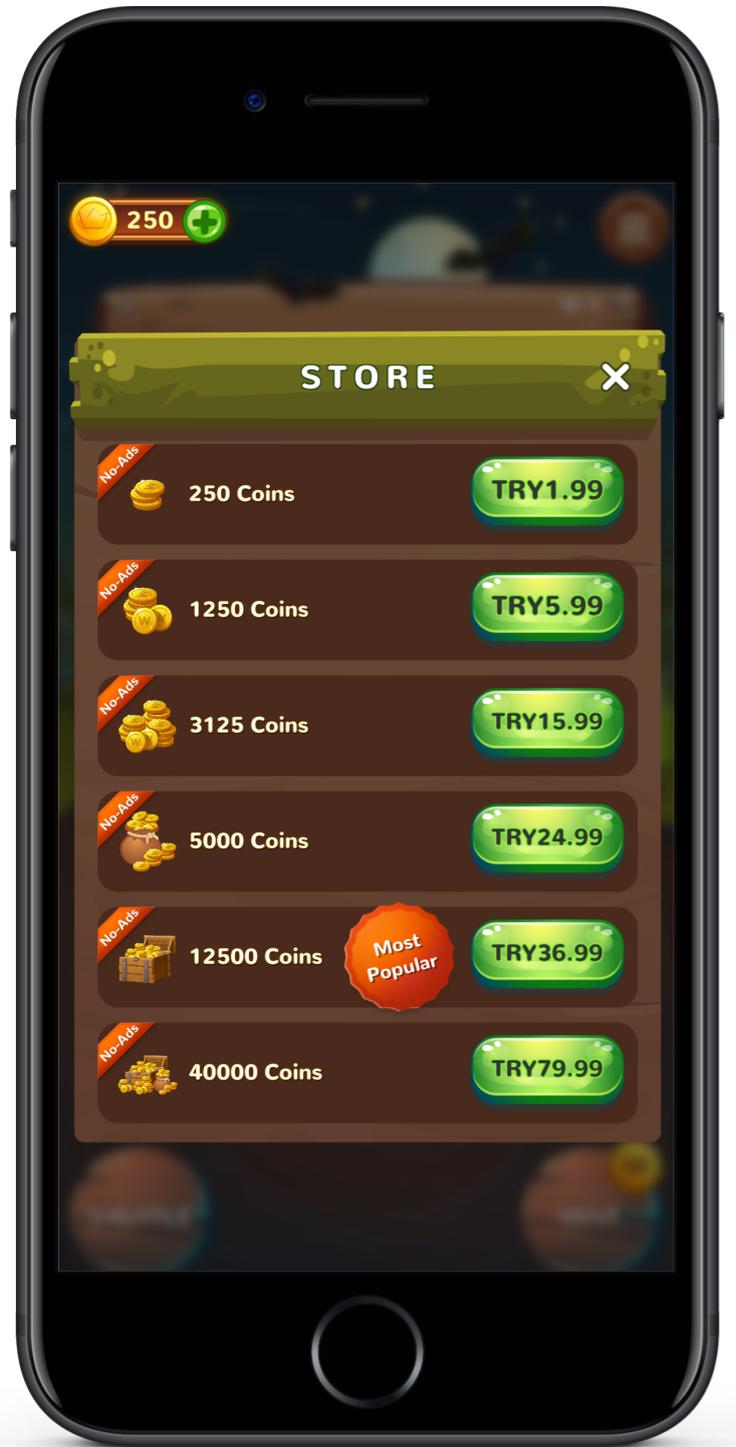
Freemium



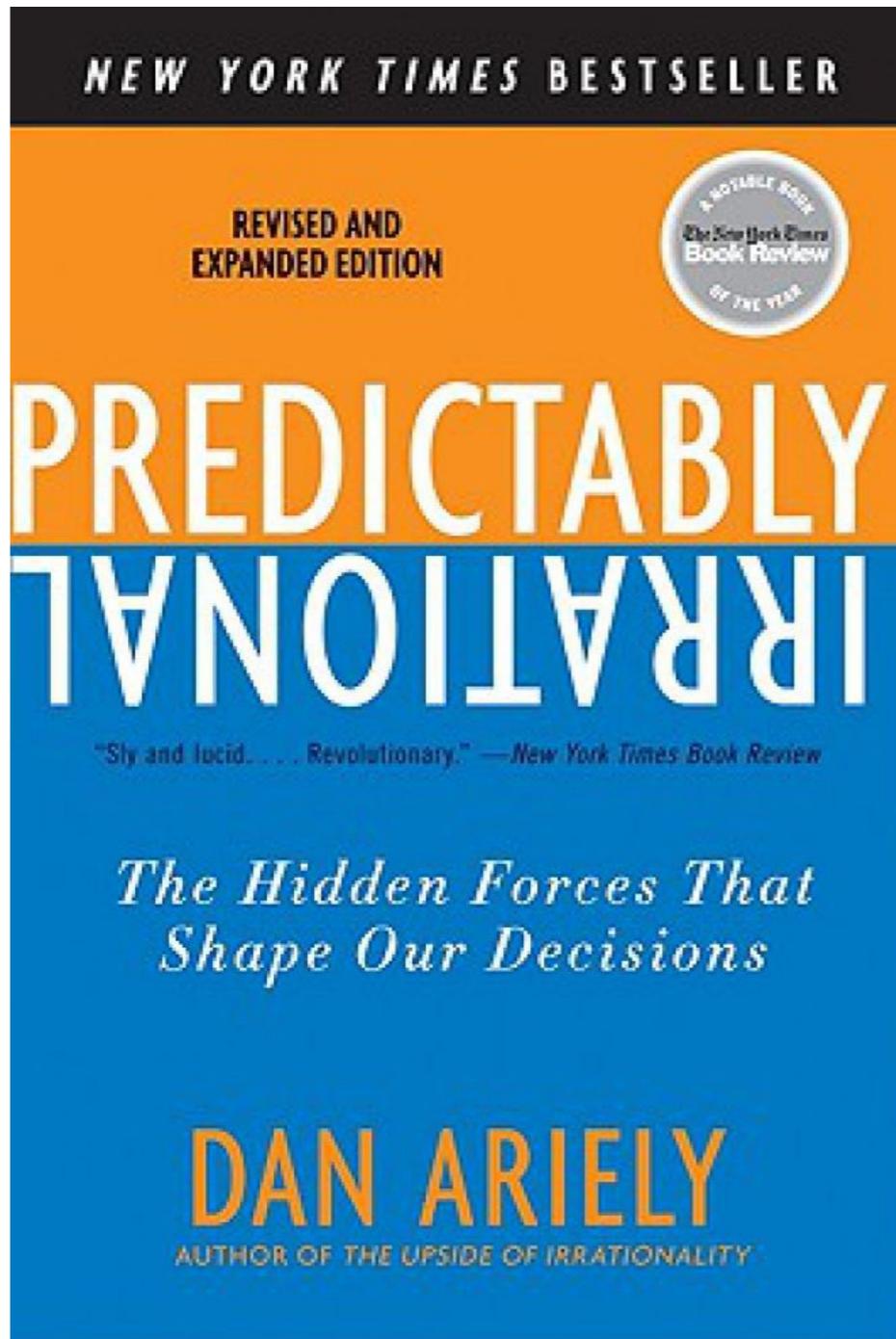
Challenge #1

In which we try
to figure out the
consumables

Challenge #1

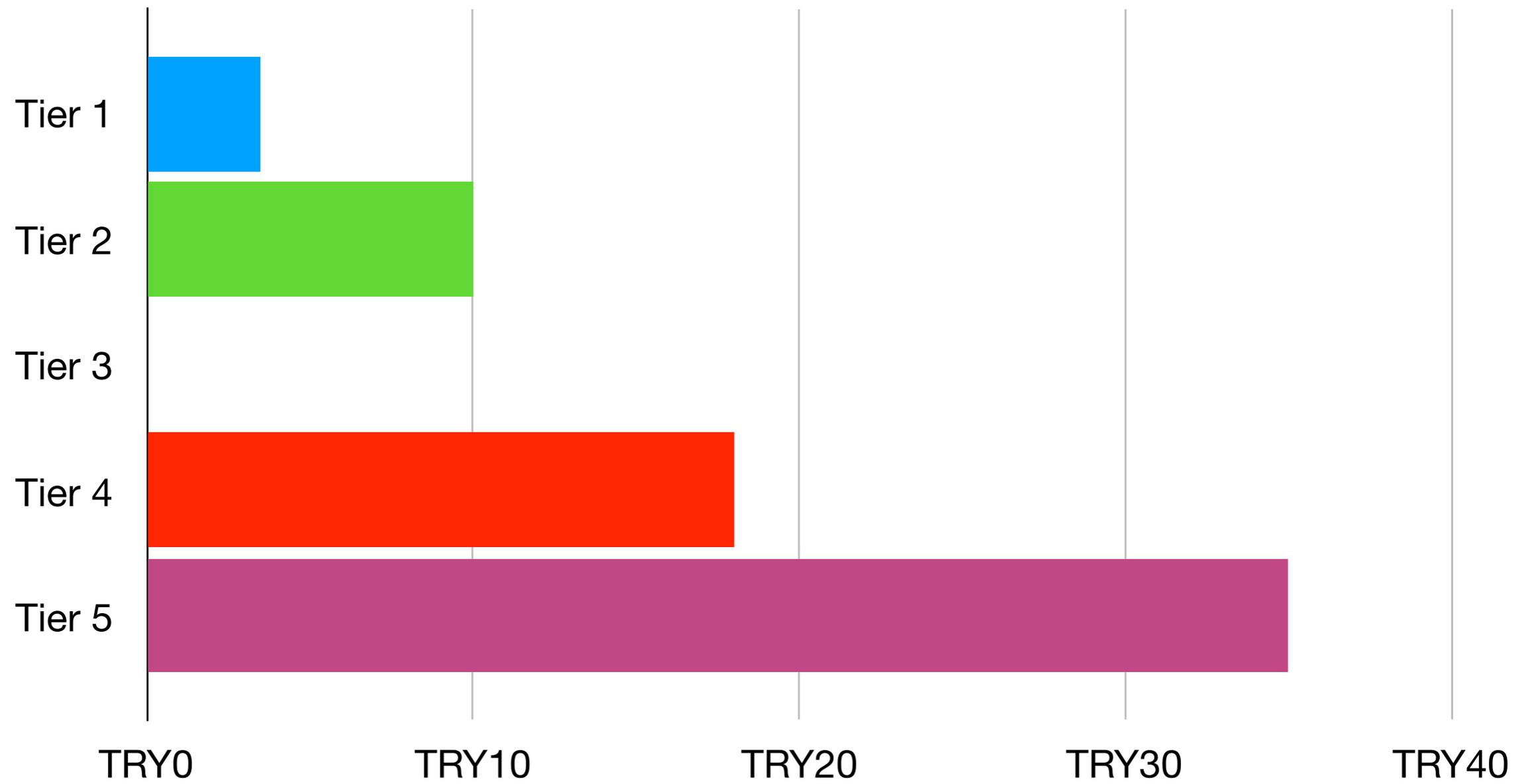


Challenge #1

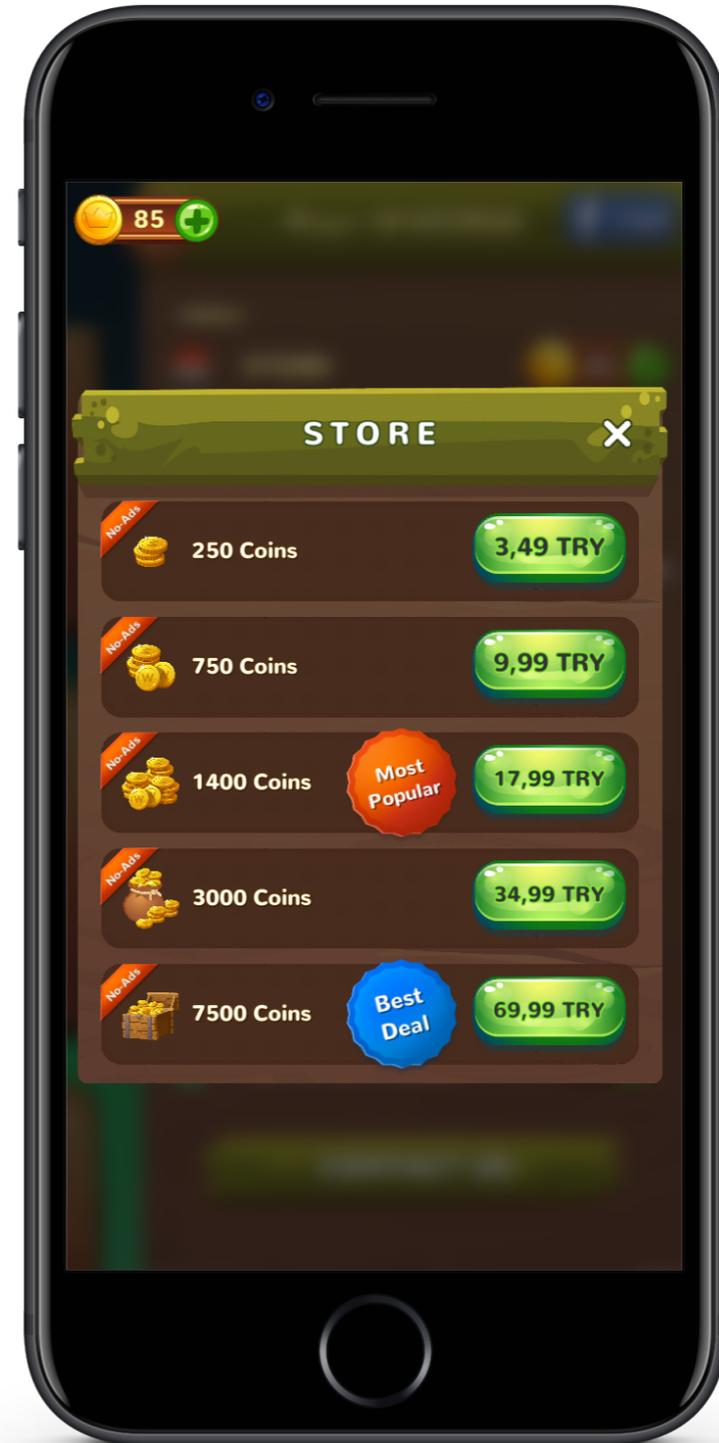
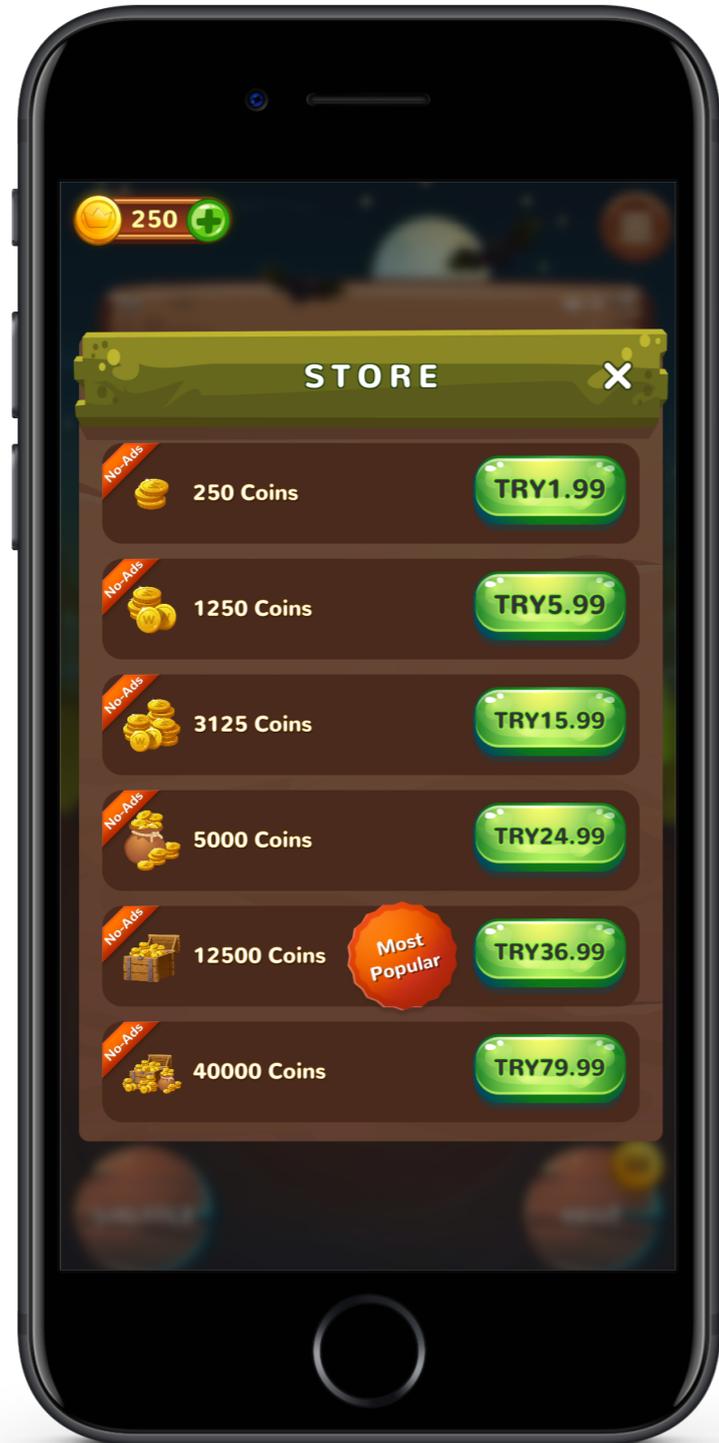


If you want to look more attractive, hang out with a slightly less attractive version of yourself

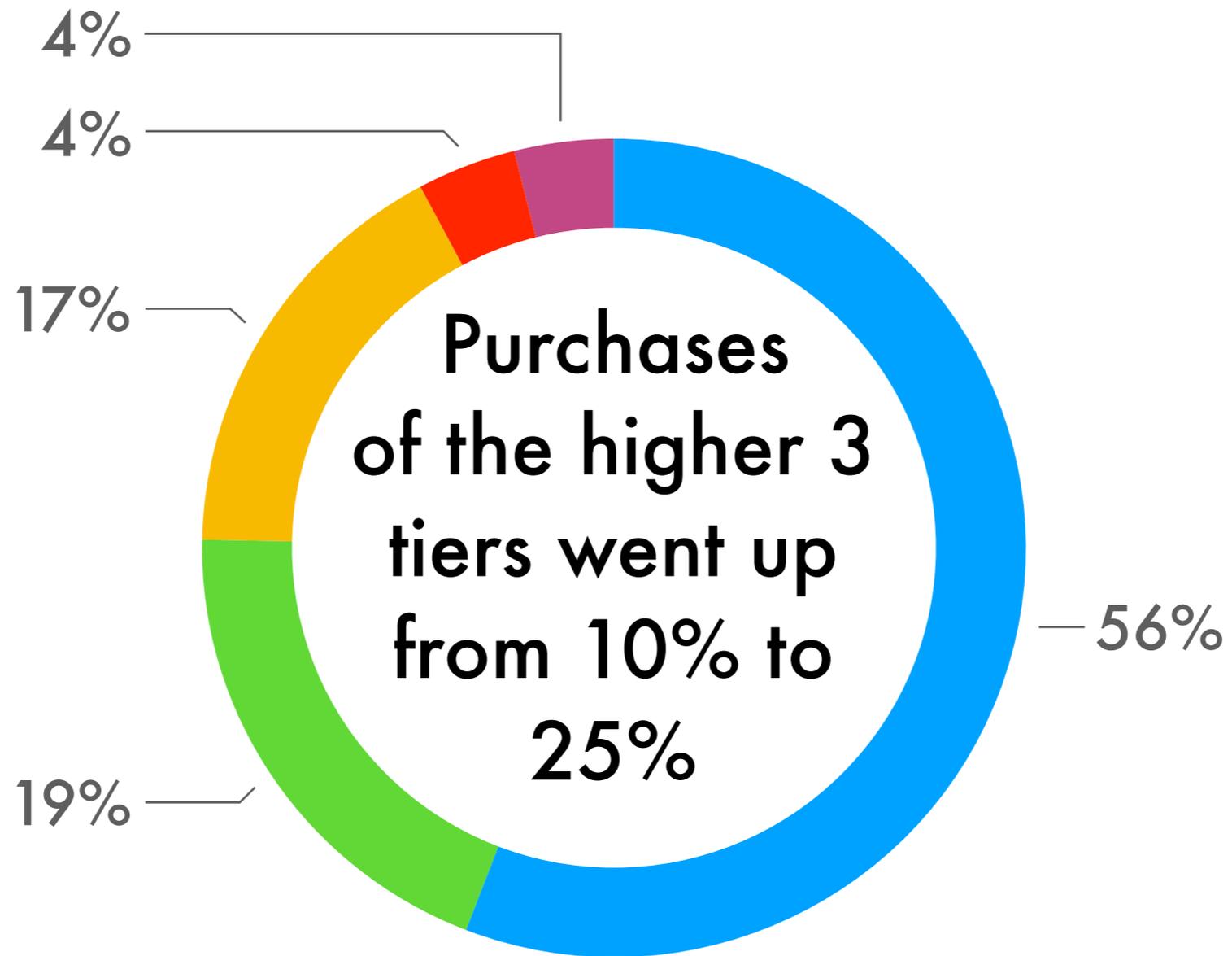
Solution



Solution

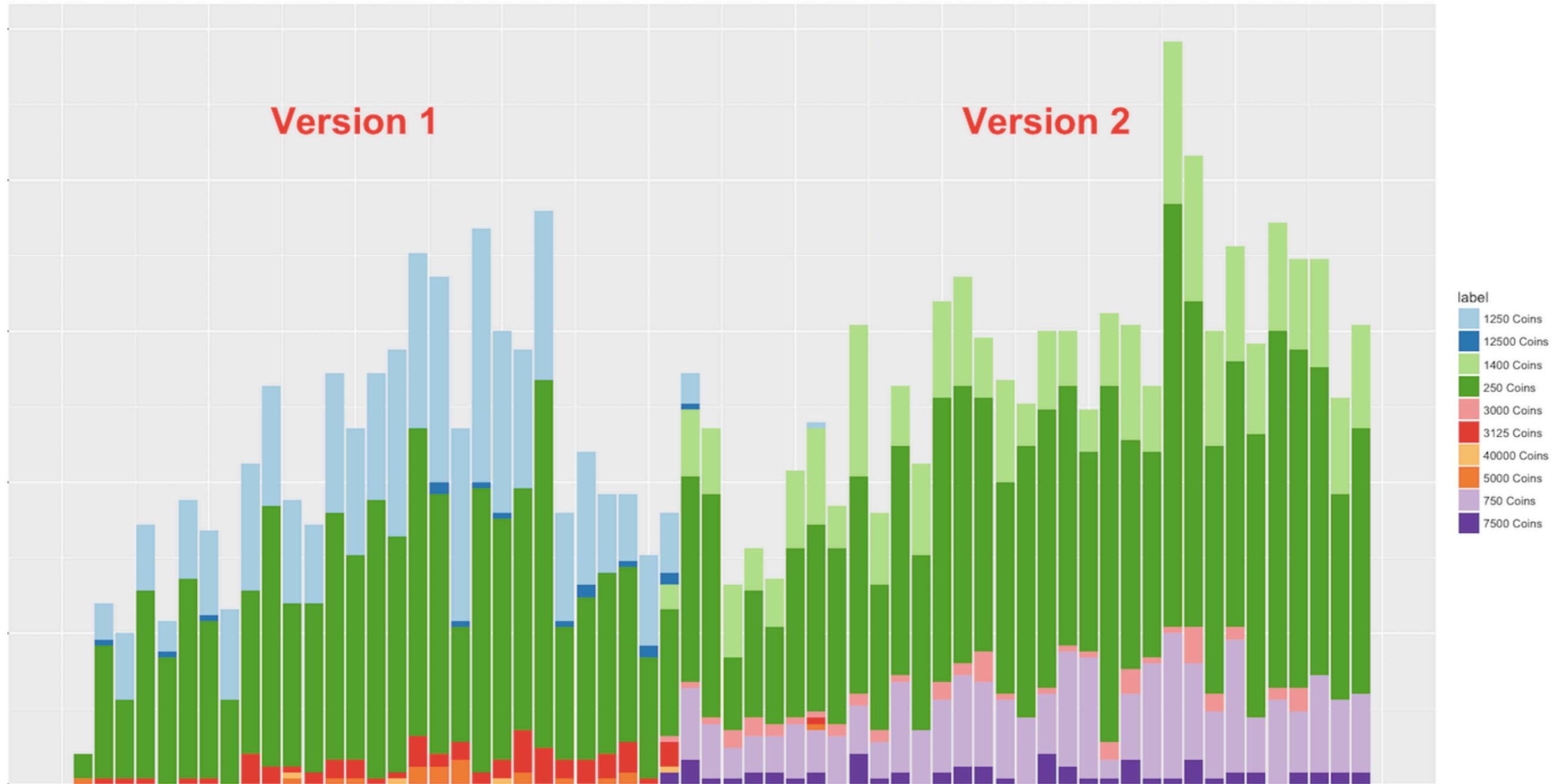


Result



Result

Transacions



Challenge #2

**In which users get
more coins for free,
while we make
more ad revenue**

Challenge #2



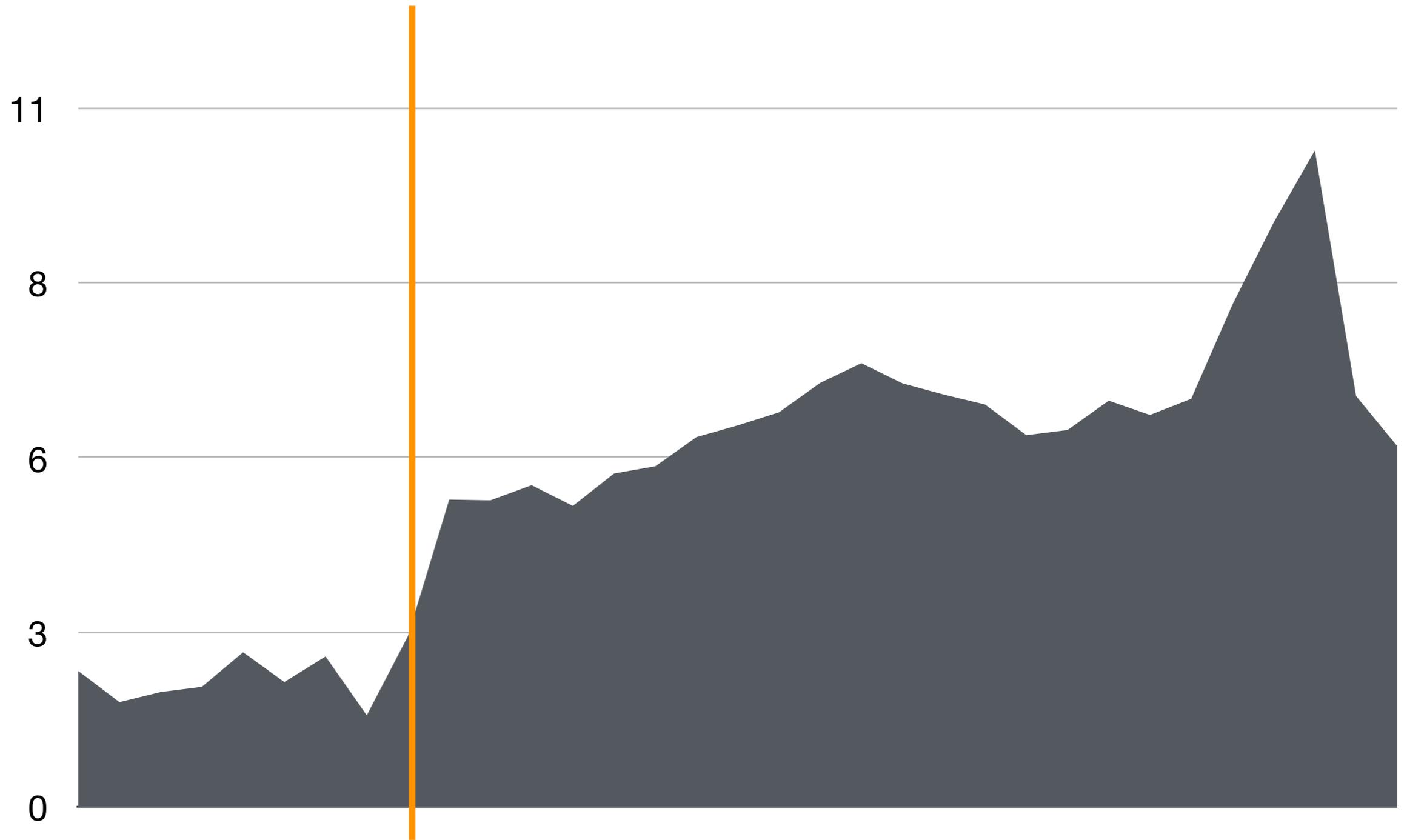
- Video ads are a great middle ground
- Players get free coins
- We still make money

Solution



**We added a
counter here!**

Result



Challenge #3

In which we
monitor where
players are stuck

Challenge #3

90 levels

10 puzzles
each

Challenge #4

In which we try
to define churn

Challenge #4

Less than 3
logins since
last week?

No
login for 2
days?

Less than 10
logins since
installation?

Only 1
login since
installation?

Left the
game at the
first level?

Solution

Login Counts

Login Count Distribution

Detailed Churn Table

Level Distribution

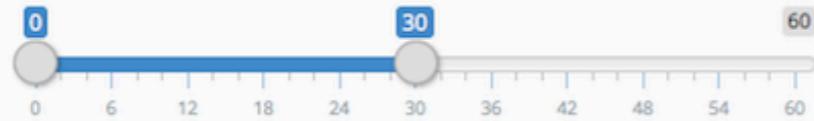
Login Count = Players' login count

Date Interval = Players have session in these intervals

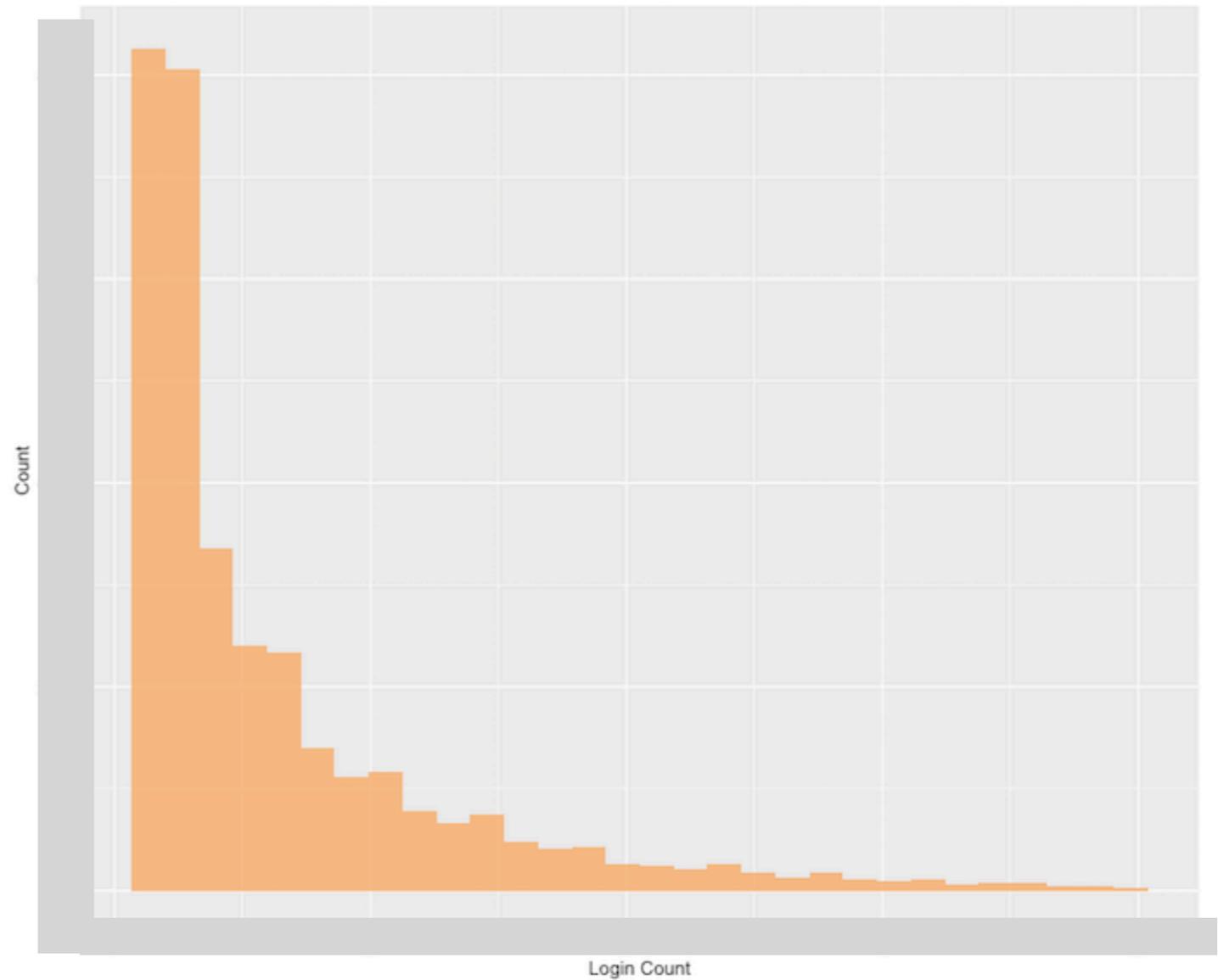
Login Count



Date Interval



Distribution of Login Count



Solution

Login Counts

Login Count Distribution

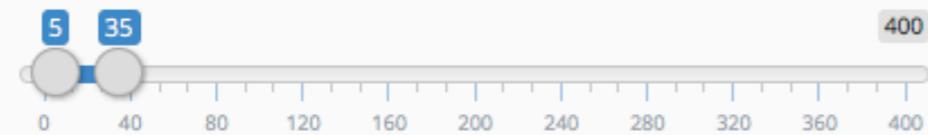
Detailed Churn Table

Level Distribution

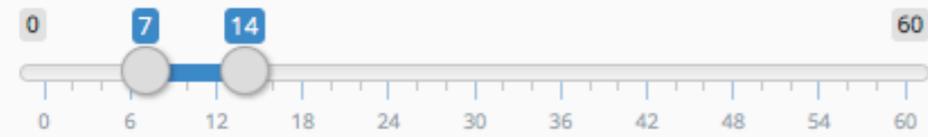
Login Count = Players' login count

Date Interval = Players have session in these intervals

Login Count



Date Interval



 Download Dataset

Show 25 entries

user_id	login
92	14
136	6
173	9
280	15
322	15
417	6
467	7
563	26
612	7
617	23
739	22
762	7

Shiny Case Study: Wordfarm

Why Shiny?

- We ♥ R
- No need to have web development skills
- Easy to write and deploy
- We can measure different metrics which some tools don't

What We Used

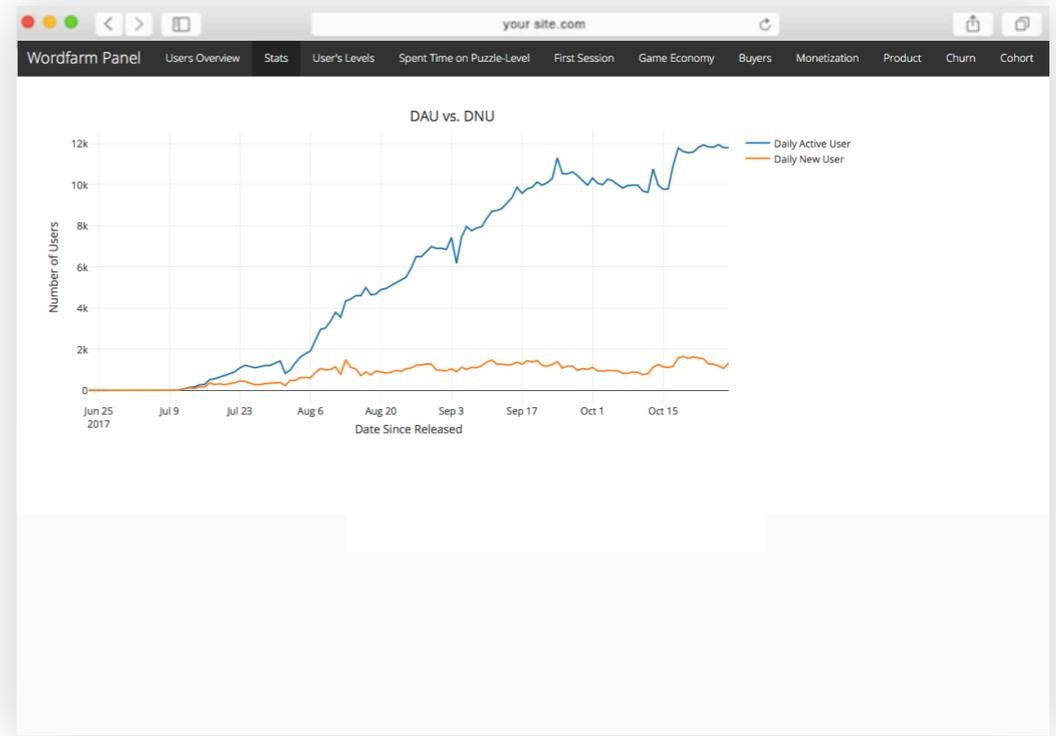
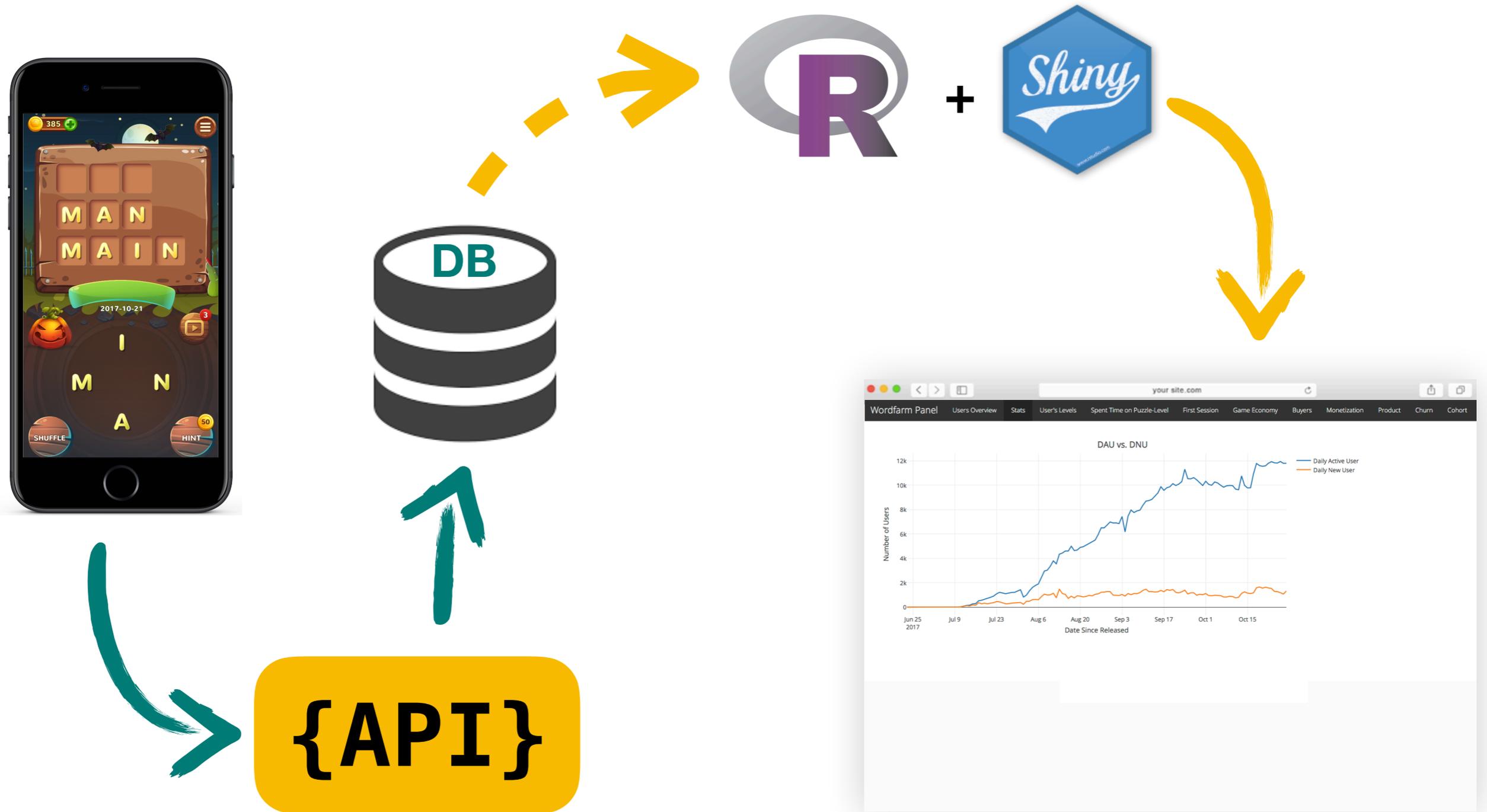
- Django
- PostgreSQL
- Git
- BitBucket



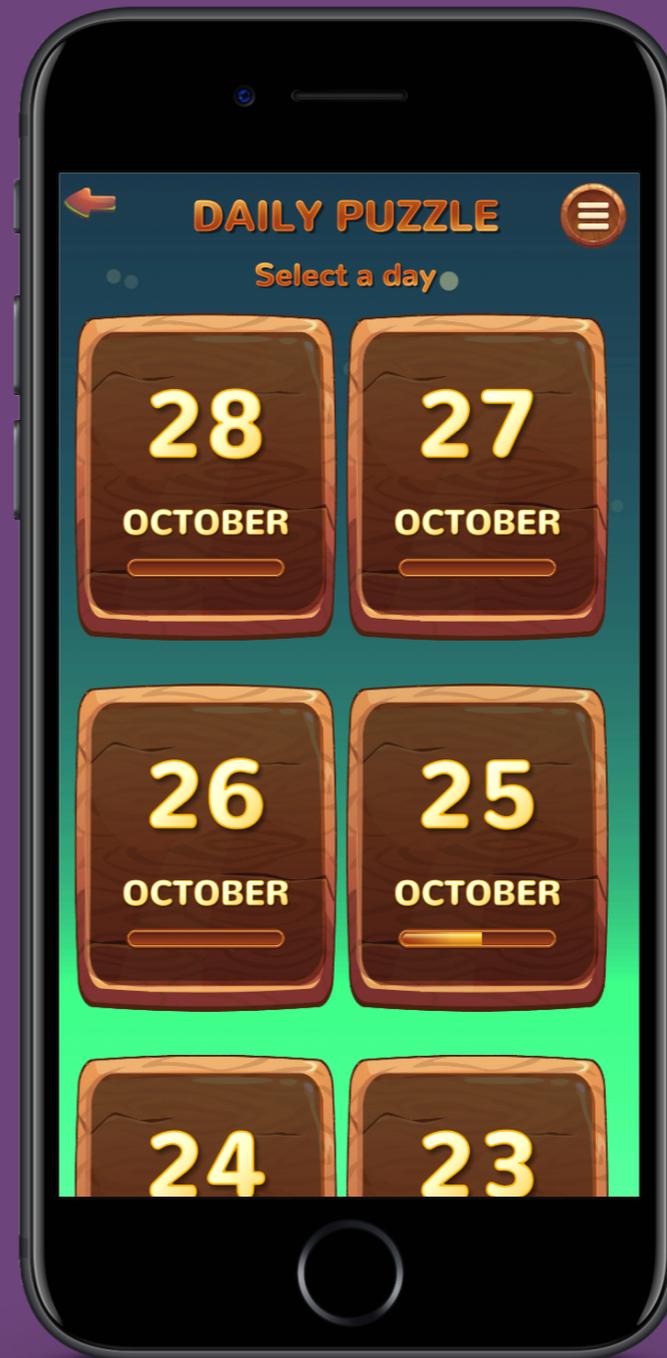
Libraries



Information Flow



Game Dynamics



Game Dynamics



Info We Gathered



demographics, device, date

level-puzzle info

in game transactions

first session duration

spent time in puzzle

order details

The Implementation

```
library(shiny)
library(tidyverse)
library(...)
```

```
source('churn.R')
source('product.R')
```

```
ui <- fluidPage(...,
  navbarPage("Wordfarm Panel", ...,
    tabPanel("Stats", ...),
    tabPanel("Game Economy", "..."),
    tabPanel("Product", "...")
  ))
```

```
server <- function(input, output, session{
  output$firstPurchase <- renderPlot({
    ggplot(...)
  })
  output$buyersTable <- renderDT({...})
}
```

The Implementation

Libraries

```
library(shiny)
library(tidyverse)
library(...)
```

UI

```
ui <- fluidPage(...,
  navbarPage("Wordfarm Panel", ...,
    tabPanel("Stats", ...),
    tabPanel("Game Economy", "..."),
    tabPanel("Product", "...")
  ))
```

Source Files

```
source('churn.R')
source('product.R')
```

Server

```
server <- function(input, output, session{
  output$firstPurchase <- renderPlot({
    ggplot(...)
  })
  output$buyersTable <- renderDT({...})
}
```

```
ui <- fluidPage(...,  
  
  tabPanel("Plots: DAU vs. DNU",  
    mainPanel(br(),  
      withSpinner(plotOutput(outputId = 'DAUvsDNU',  
        width = "100%", height = "400px")  
      )  
    )  
  )  
)
```

```
server <- function(input, output, session) {  
  output$DAUvsDNU <- renderPlot({  
  
    ggplot(data = DAUandDNU(), aes(x = date)) +  
    geom_line(aes(y=dailyactiveuser, colour = "dailyactiveuser")) +  
    geom_line(aes(y = dailynewuser, colour = "dailynewuser"))  
  
  })  
}
```

```
ui <- fluidPage(...,  
  
  tabPanel("Plots: DAU vs. DNU",  
    mainPanel(br(),  
      withSpinner(plotOutput(outputId = 'DAUvsDNU',  
        width = "100%", height = "400px")  
      )  
    )  
  )  
)
```

```
server <- function(input, output, session) {  
  output$DAUvsDNU <- renderPlot({  
  
    ggplot(data = DAUandDNU(), aes(x = date)) +  
    geom_line(aes(y=dailyactiveuser, colour = "dailyactiveuser")) +  
    geom_line(aes(y = dailynewuser, colour = "dailynewuser"))  
  
  })  
}
```

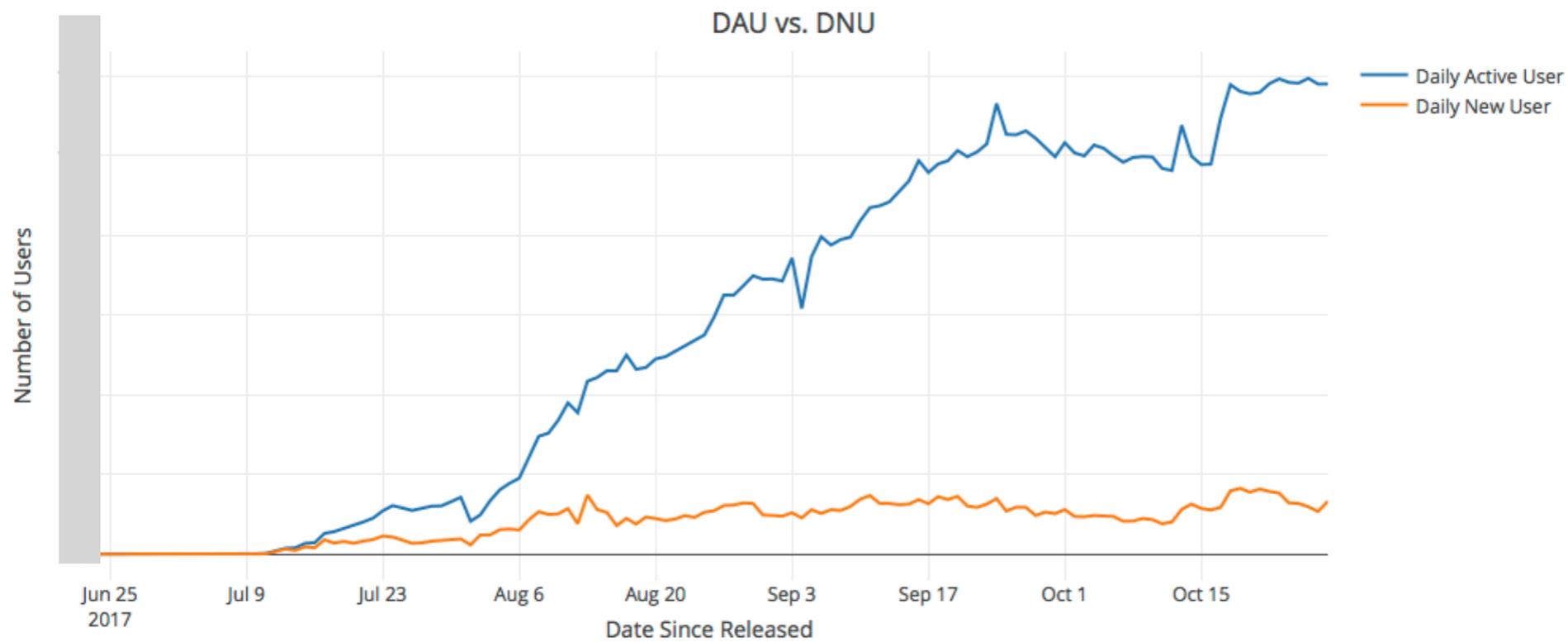
Shiny Server

- Easy deployment with Shiny Server
- Open Source
- Can host multiple applications
- Run on Linux Servers

The Dashboard

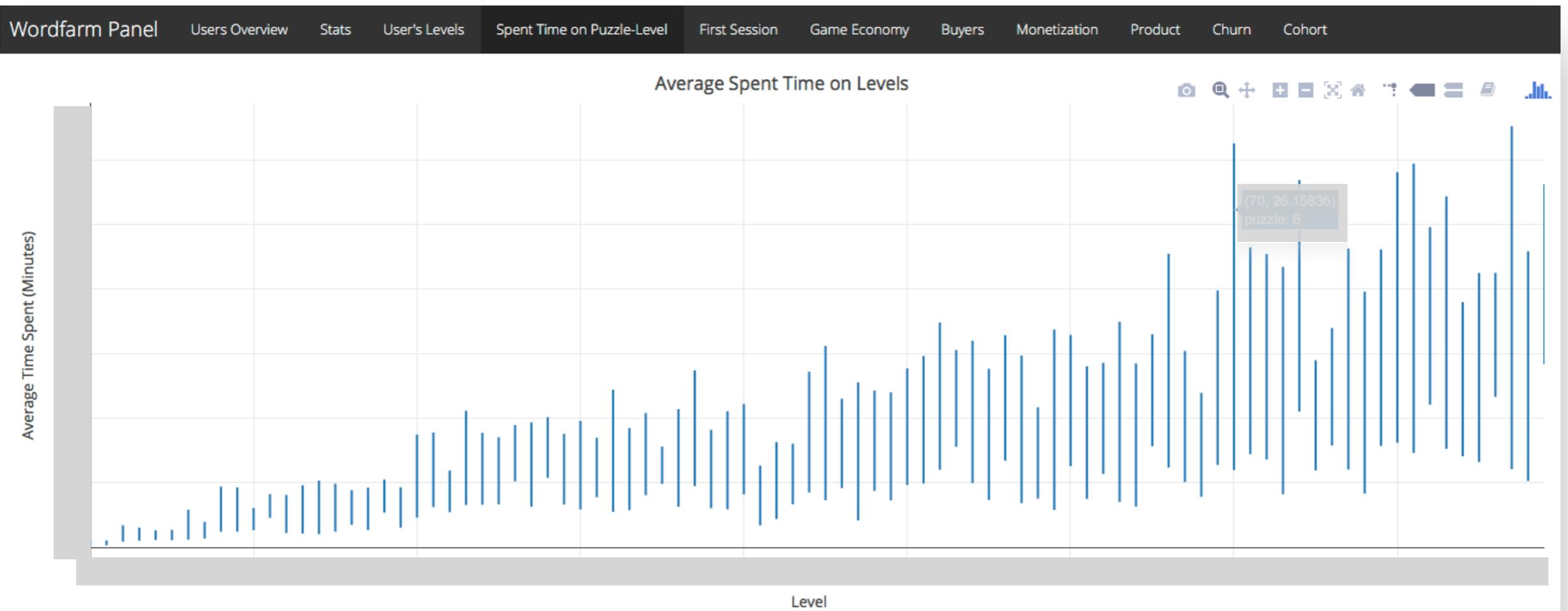
Wordfarm Panel

Wordfarm Panel Users Overview Stats User's Levels Spent Time on Puzzle-Level First Session Game Economy Buyers Monetization Product Churn Cohort



DAU vs DNU

Wordfarm Panel



Average Spent Time on puzzle-level

Wordfarm Panel

Wordfarm Panel Users Overview Stats User's Levels Spent Time on Puzzle-Level First Session Game Economy Buyers Monetization **Product** Churn Cohort

Product Filtering

Coins
250 Coins

Date Range
2017-07-15 to 2017-11-07

Download Dataset

Show 25 entries

Search:

Date	Platform	Label	Buyer	Transaction	RevenueUSD	AvgTransPerBuyer	AvgRevUSDPerTrans
------	----------	-------	-------	-------------	------------	------------------	-------------------

Product

Wordfarm Panel

Wordfarm Panel

Users Overview

Stats

User's Levels

Spent Time on Puzzle-Level

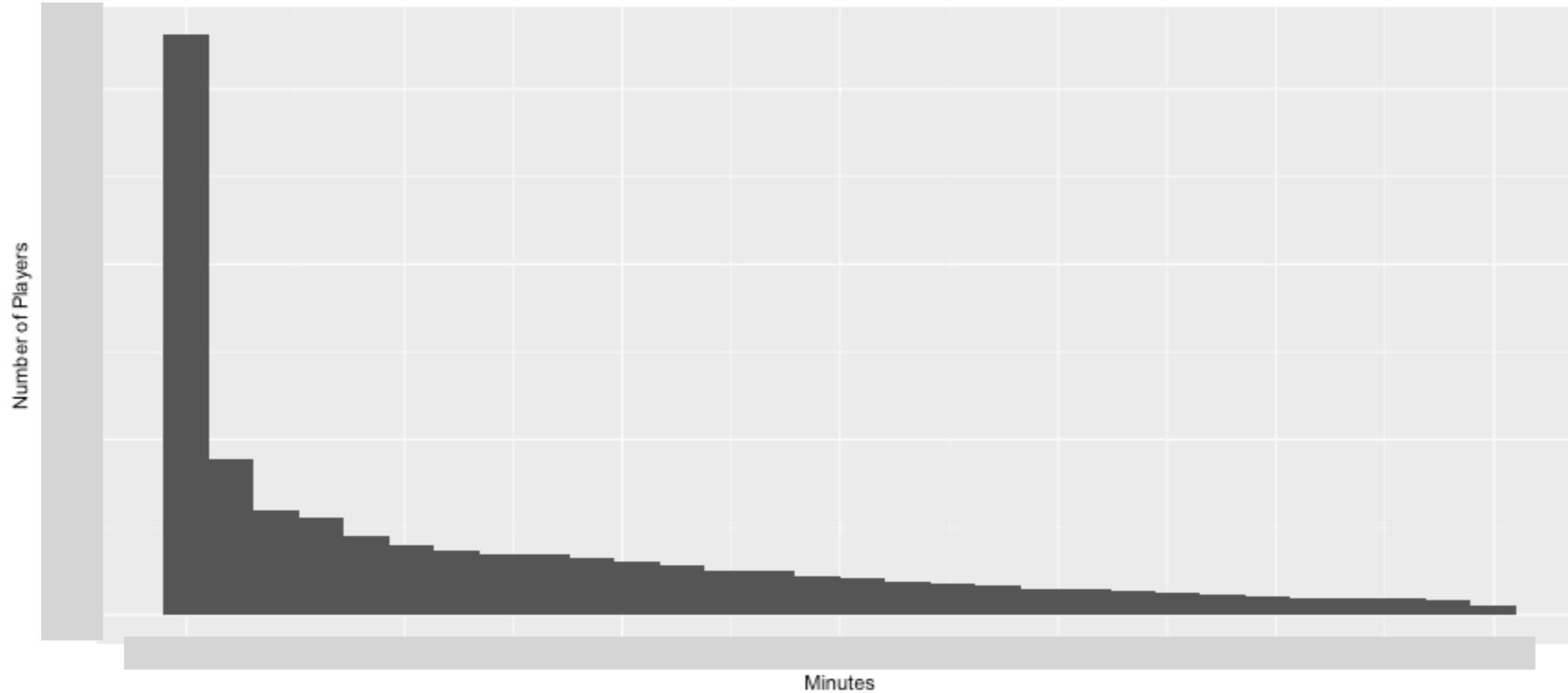
First Session

Game Economy

Buyers

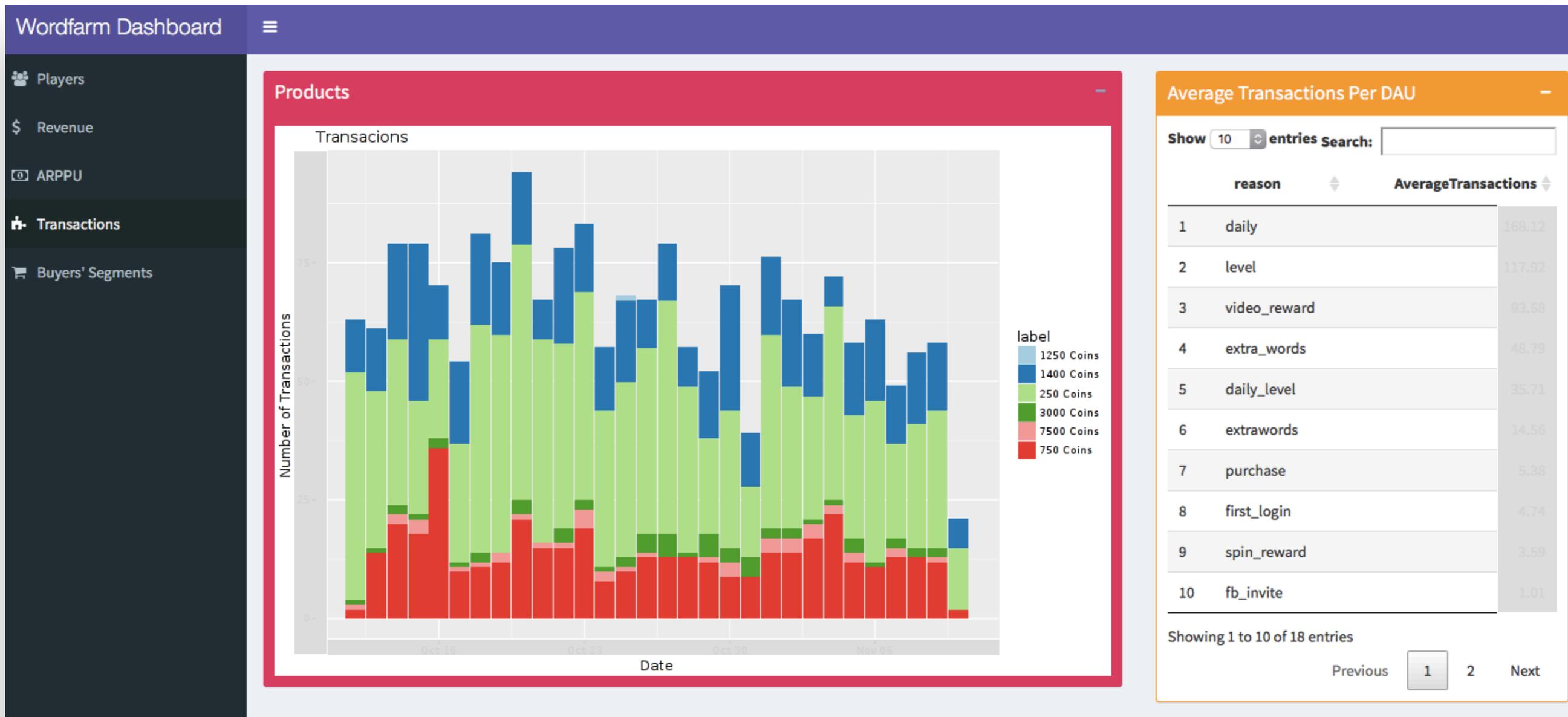
Monetization

First Session Duration Distribution



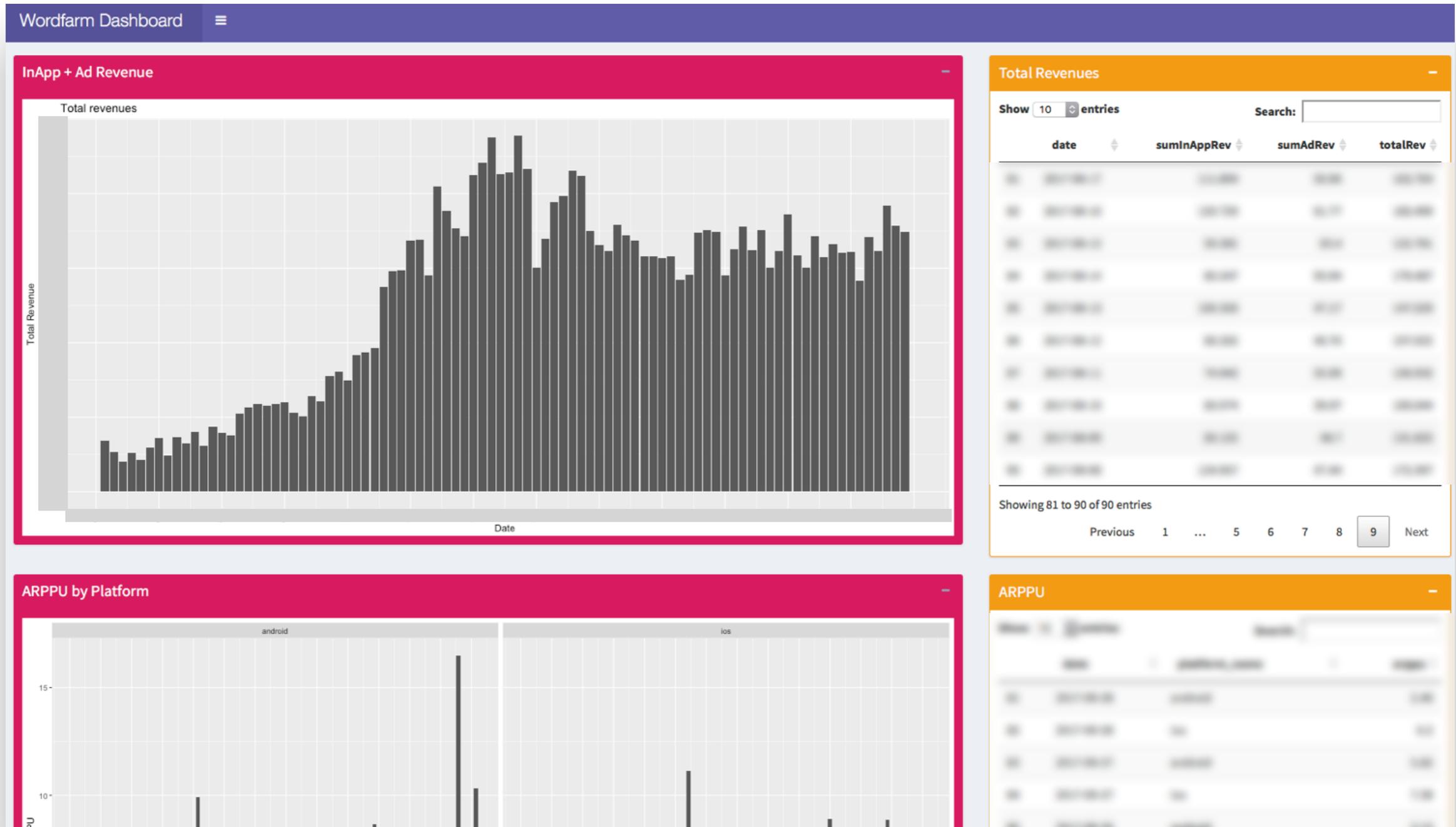
Stickiness of a game

Revenue Dashboard



Average Transactions

Revenue Dashboard



ARPPU by platform

Revenue KPIs

ARPU

Average
revenue per
user

ARPPU

Average
revenue per
paying user

ARPPDAU

Average
revenue per
daily active
user



16%

increase in

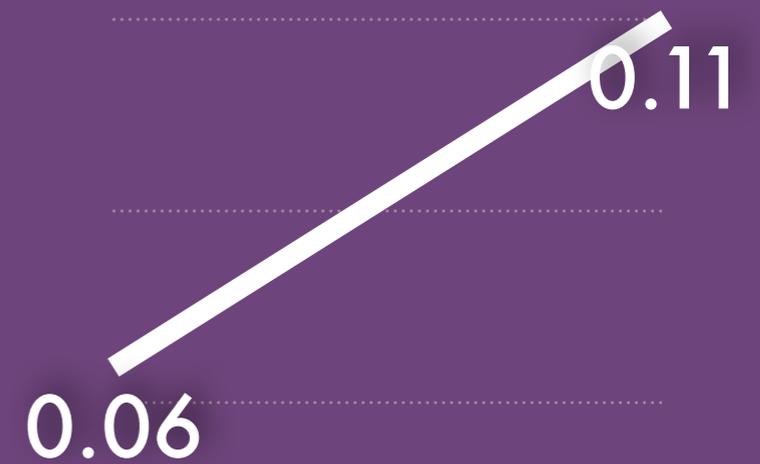
ARPU



61%

increase in

ARPPU

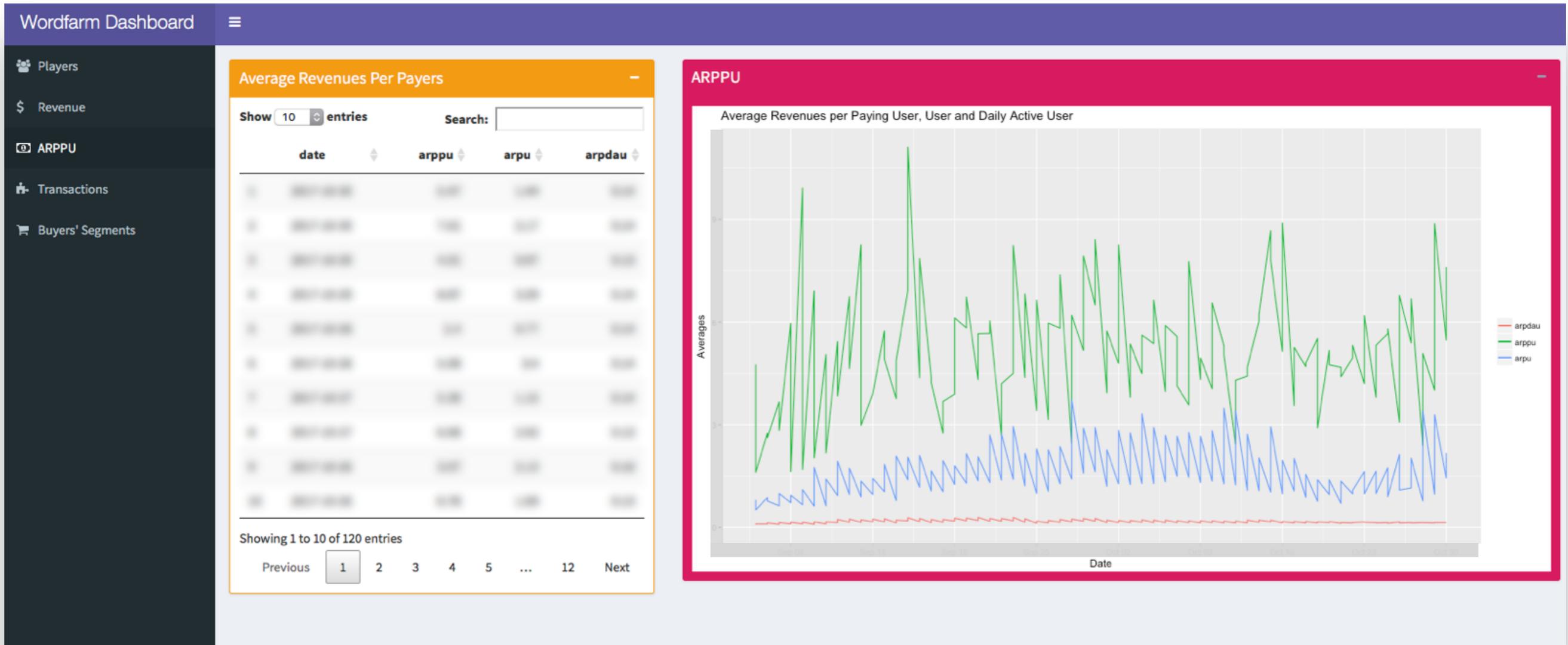


83%

increase in

ARPDau

Revenue Dashboard



Average Revenues

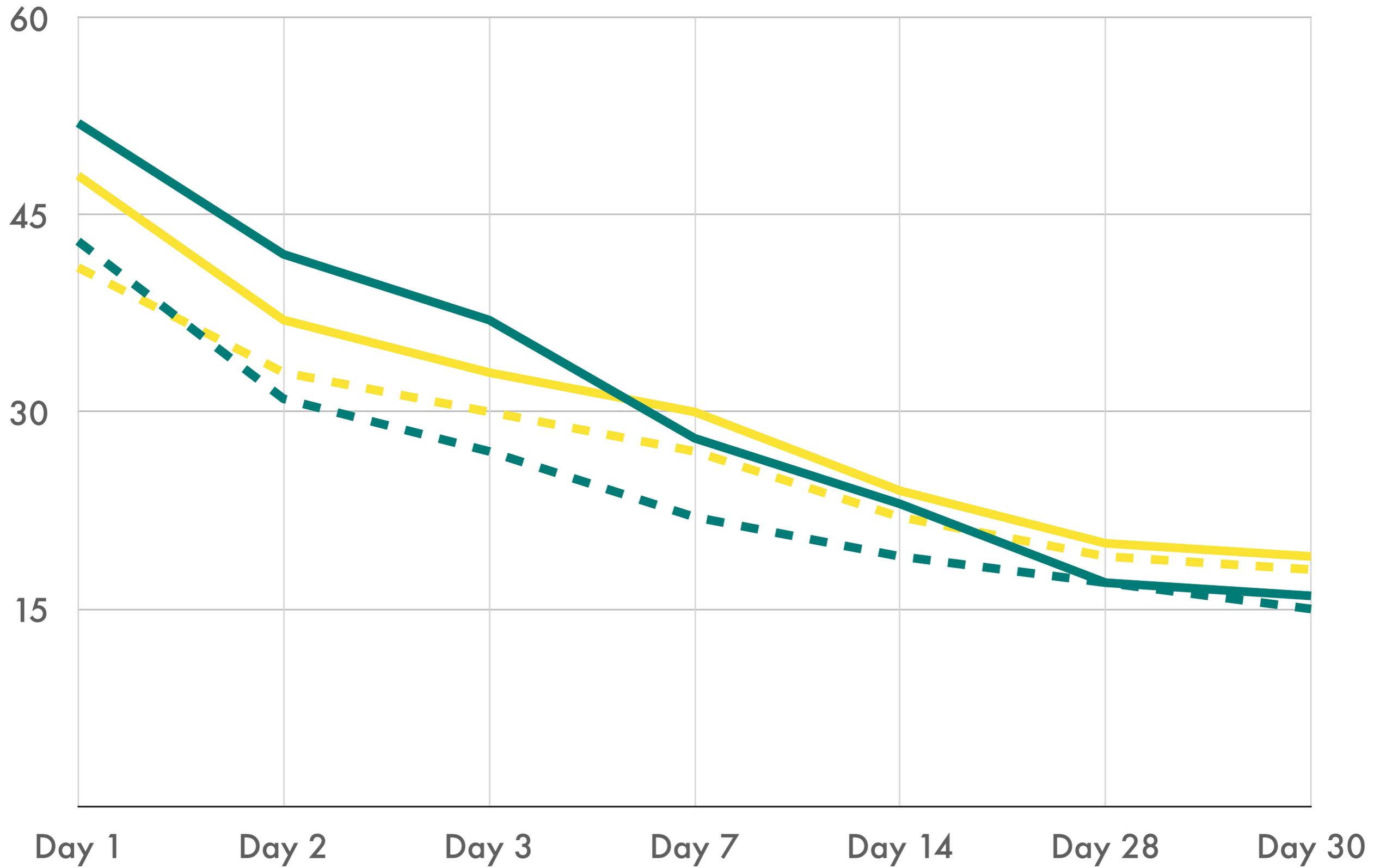
Happy

Ending: High

Retention

Rates

--- iOS_v1 — iOS_v2 - - - Android_v1 — Android_v2



Thank You!

Any Questions?



@ZofiatheWitch

hazelkavili.me

hazel.kavili@wixot.com