

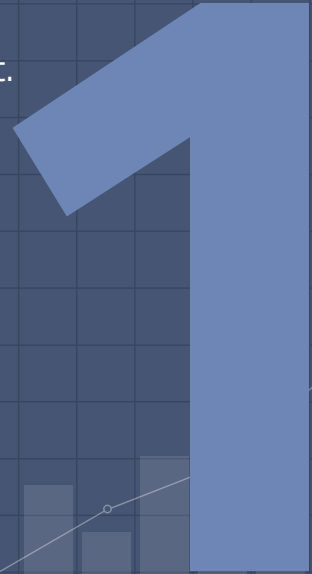


# Problem Statement

Gaming has been growing exponentially over the past few decades and games have come a long way in terms of technical advances and the way it's received in our society. Gaming is proved to be a double edged sword with problems like expensive drivers, increasing size, bad influence, addiction.

We can sum most of the problems to this:

- > Bad interface for people to buy hardware & drivers based on their requirement.
- > People not following age restrictions imposed by PEGI, other companies.
- > The growth of game size is increasing exponentially
- > Lack of publicity for indie games means people miss lot of good stuff.



# How we solve the problem

-> Giving Users an interface where it compares our current Drivers installed on PC to the driver requirements of the games on the user's "Wish-List" and providing shopping options based on:

- a) Maximum Requirements
- b) Minimum Requirements
- c) Economically Friendly

-> Analysing the size of games on PC by obtaining a list of installed games and suggesting to delete the games that are not used (or) occasionally used.

-> By checking the benchmark of games without running the game before buying it.

-> Recommending games for you with algorithm that User can customize by changing percentage of User recommendation and what their friends are playing.

Eg: 80% our recommendations + 20% friends recommendations.

-> Page dedicated for Indie Developers.

# PREMIUM FEATURES

## DRIVERS

Checks the drivers of PC and suggests new drivers, checks whether you can run a game in your PC.

## LIVE WALLPAPERS

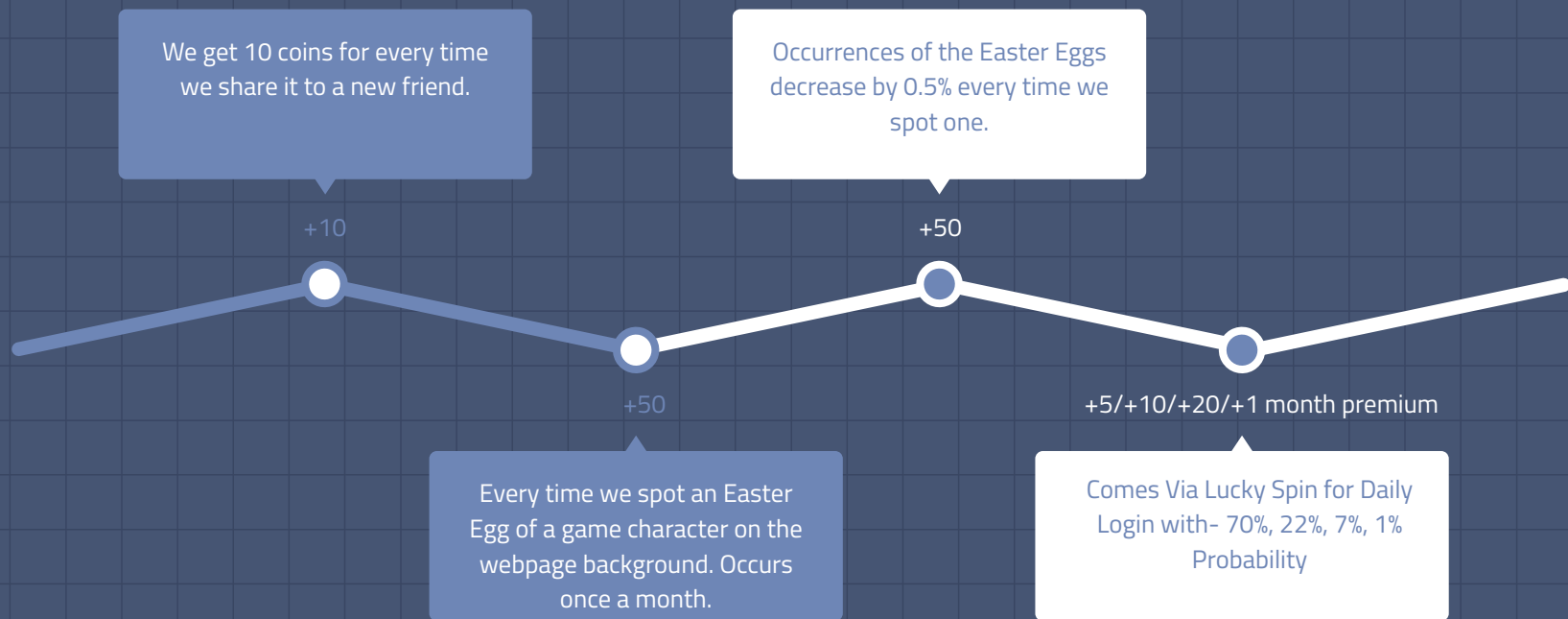
Gives links to Live Wallpapers, 4K wallpapers of that game to download.

## FPS\*

Obtains the Frames Per Second of a game from database of websites to determine performance of game in its lowest and highest graphical settings.

\*only for laptops

# HOW USERS EARN COINS



# Target Audience:

FORMULA = 70%gamers+30% Discord, Reddit users.

Age	15-24	25-34	35-44	45+
Men	54%	45%	46%	37%
Women	47%	41%	46%	41%
GameByte	52%	24%	17%	7%

Amount spent in our marketing,  
Free marketing is the best marketing.

# Marketing Strategy

8

## Earn Coins

What works about our interface is that we use “Premium” feature as a way to earn USERS rather than MONEY. This system is explained properly in next page.

## Platform for Indie

We are focusing on Indie games, making this interface ideal for indie game developers. Indie developers get to PUBLICIZE their Indie Game.

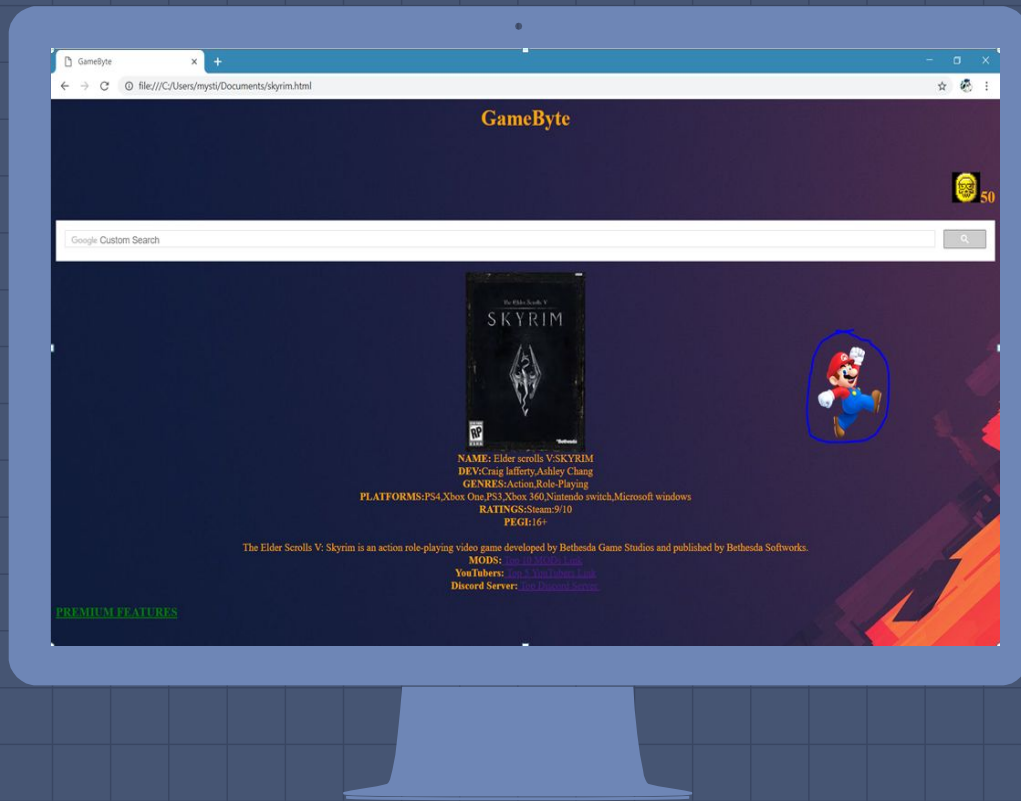
## Gamble Coins

A feature where we can Gamble with CPU and other players to earn or lose coin. With the win fetching 2X the amount of initial coins. This increases the positive competitive nature among peers. Eg: discord server bot game, which is addictive.



# EASTER EGGS

If we are able to spot a game character in our website, we get 50 coins as a reward.



# Earning Money:

## COINS:

We include coins as one of the in-app purchases with the incentive of bringing more players, this is explained in our "How to earn coin" slide.

## ADS:

- We start implementing ADS in the player recommendation page, this works without interrupting the essence of our UI.

The background features a dark blue grid. A white line graph with circular markers is positioned horizontally across the middle. Below the text, there is a faint bar chart with vertical bars of varying heights.

Let's see how some  
Python Backend codes  
work....

## Earning coins by sharing

```
RESTART: C:/Users/Nimish Mittal/AppData/Local/E

Current no of shares: 1
Current no of shares: 2
Current no of shares: 3
Current no of shares: 4
Current no of shares: 5
Current no of shares: 6
Current no of shares: 7
Current no of shares: 8
Current no of shares: 9
Current no of shares: 10
Current no of shares: 11
Current no of shares: 12
Current no of shares: 13
Current no of shares: 14
Current no of shares: 15
Do you want to buy premium? Yes
Current no of shares: |
```

## Gambling Coins

```
#read current number of coins, suppose its 100
coins=100
import random

bet=int(input())

a = random.choice([1,2,3,4,5,6,7,8,9,10])

if a=="1":
    print("You Win")
    coins=coins+bet
else:
    print("You Lose")
    coins=coins-bet

|
```

# Premium Store

```
coins=50
premium=0
oldshares=0
newshares=0

while True:
    if newshares==oldshares+1:
        coins = coins+ 10
    newshares=oldshares
    #taking the number of shares info from website
    #oldshares=int(input("Current no of shares: "))

    if coins >=200:
        a=input("Do you want to buy premium? ")
        if a=="Yes":
            coins=coins-200
            premium=1
```

# Alternate way for drivers

```
#take input from search
model_of_your_pc = input("Enter the model of your pc: ")
if model_of_your_pc=="Dell Inspiron 7577":
    a=open("Dell Inspiron 7577.txt",'r')
    print(a.read())
```

# DETECTING PROCESSOR USING PYTHON.

```
>>> import platform
>>> platform.platform()
'Windows-10-10.0.17134-SP0'
>>> platform.processor()
'Intel64 Family 6 Model 158 Stepping 9, GenuineIntel'
>>> platform.system()
'Windows'
```

# IMPORTING DATA FROM URL USING BEAUTIFULSOUP

```
In [1]: import urllib
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
from urllib.request import urlopen
```

```
from bs4 import BeautifulSoup
import requests
```

```
In [2]: import urllib
from bs4 import BeautifulSoup
```

```
In [5]: url='https://www.actiongame.com/'
```

```
In [6]: html=urlopen(url)
```

```
In [8]: soup = BeautifulSoup(html, 'lxml')
```