CS4080 Final Presentation:

NBA Statistic Web Scraper

Caleb Ng, Darrel Chang, Andy Nguyen, Jonah Lysne

Introduction to Golang

- Created by Google and released as open-source in 2009
- Procedural, strictly typed language
- Designed for concurrency with garbage collection
- Fast and simple thanks to its compiled nature
- Uses packages for managing files and dependencies

The Golang Gopher

```
GO test.go > ...
1   package main
2
3   import "fmt"
4
5   func main() {
6       fmt.Println("Hello, World!")
7   }
8

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

(base) [redacted]-aruba-authenticated-10-110-249-81 Go % go build test.go
(base) [redacted]-aruba-authenticated-10-110-249-81 Go % ./test
Hello, World!
```

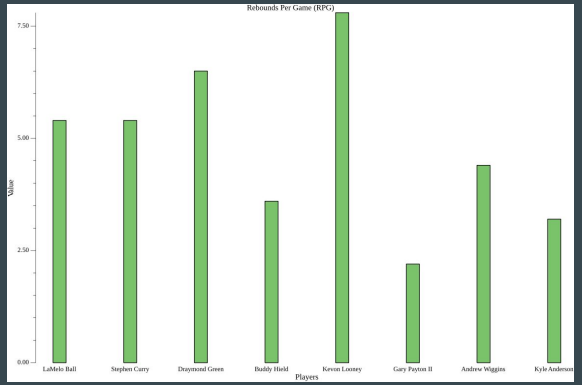
Project Goal

The goal of the project is to create a console application that allows users to...

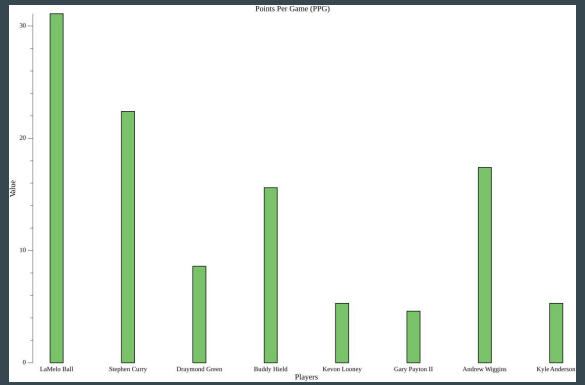
=> Input NBA player names via their NBA.com URL

=> Web scrape the the player's profile and ge their basketball game statistics

=> Generates the graphical representations of the data



rebounds per game

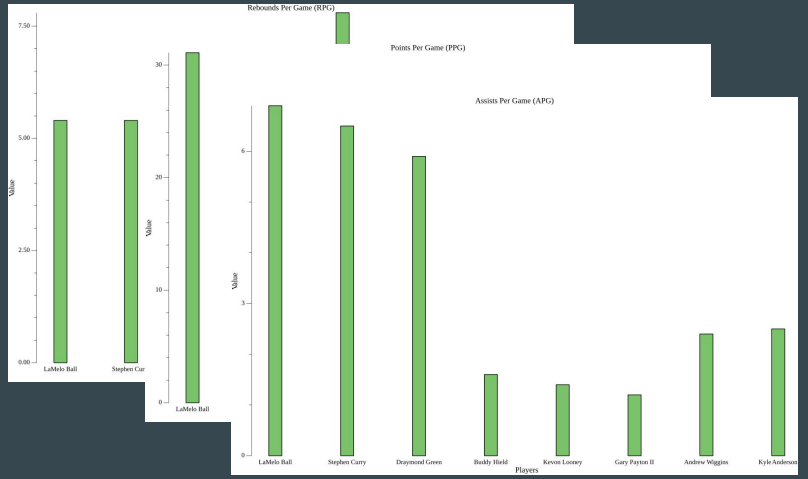


points per game

Project Features

The application aims to provide users with a dynamic visualization tool for analyzing the selected amount of player performance

- fantasy sports
- performance tracking
- sports analytics



PNG graphs

NBA website

First Name	Last Name	Team	PPG	RPG	APG
LaMelo	Ball	Charlotte Hornets	31.1	5.4	6.9
Stephen	Curry	Golden State Warriors	22.4	5.4	6.5
Draymond	Green	Golden State Warriors	8.6	6.5	5.9
Buddy	Hield	Golden State Warriors	15.6	3.6	1.6
Kevon	Looney	Golden State Warriors	5.3	7.8	1.4
Gary	Payton II	Golden State Warriors	4.6	2.2	1.2
Andrew	Wiggins	Golden State Warriors	17.4	4.4	2.4
Kyle	Anderson	Golden State Warriors	5.3	3.2	2.5

CSV file

Key Languages Features [Part 1]

```
module GoScrapper

go 1.23.3

require (
	gioui.org v0.2.0 // indirect
	gioui.org/cpu v0.0.0-20220412190645-f1e9e8c3b1f7 // indirect
	gioui.org/shader v1.0.6 // indirect
	gioui.org/x v0.2.0 // indirect
	git.sr.ht/~sbinet/gg v0.6.0 // indirect
	github.com/PuerkitoBio/goquery v1.10.0 // indirect
	github.com/ajstarks/svgo v0.0.0-20211024235047-1546f124cd8b // indirect
	github.com/andybalholm/cascadia v1.3.2 // indirect
	github.com/andybalholm/stroke v0.0.0-20221221101821-bd29b49d73f0 // indirect
	github.com/antchfx/htmlquery v1.3.3 // indirect
	github.com/antchfx/xmlquery v1.4.2 // indirect
```

File: go.mod

The go.mod file is the center of dependency management. The modules which are needed are maintained in go.mod file.

(source faun.pub)

```
gioui.org v0.2.0 h1:RbzDn1h/pCVf/q44ImQSa/J3MIFpY3OWphzT/Tyei+w=
gioui.org v0.2.0/go.mod h1:1H72sKEk/fNFV+l0JNeM2Dt3co3Y4uaQcD+I+/GQ0e4=
gioui.org/cpu v0.0.0-20210808092351-bfe733dd3334/go.mod h1:A8M0Cn5o+vY5LTMlnRoK3O5kG+rH0kWfJjeKd9QpBmQ=
gioui.org/cpu v0.0.0-20220412190645-f1e9e8c3b1f7 h1:tNJdnP5CgM39PRc+KWmBRRYX/zJ+rd5XaYxY5d5veqA=
gioui.org/cpu v0.0.0-20220412190645-f1e9e8c3b1f7/go.mod h1:A8M0Cn5o+vY5LTMlnRoK3O5kG+rH0kWfJjeKd9QpBmQ=
gioui.org/shader v1.0.6 h1:cvZmU+eODFR2545X+/8XucgZdTtEjR3QWW6W65b0q5Y=
gioui.org/shader v1.0.6/go.mod h1:mWdiME581d/kV7/iEhLmUgUK5iZ09XR5XpduXzbePVM=
gioui.org/x v0.2.0 h1:/MbdjKH19F16auv19UiQxli2n6BYPw7eyh9XBOTgmEw=
gioui.org/x v0.2.0/go.mod h1:rCGN2nZ8ZHqrtseJoQxCMZpt2xrZUrdZ2WuMRLBJmYs=
git.sr.ht/~sbinet/gg v0.6.0 h1:RIzgkizAk+9r7uPzf/VfbJHBMKUr0F5hRFxTUGMnt38=
git.sr.ht/~sbinet/gg v0.6.0/go.mod h1:uucygbfC9wVPQIfrmwM2et0imr8L7KQWywX0xpFMm94=
github.com/BurntSushi/toml v0.3.1/go.mod h1:xHWCNGjB5oqiDr8zfno3MHue2Ht5sIBksp03qcyfWMU=
github.com/PuerkitoBio/goquery v1.10.0 h1:6fiXdLuUvYs2OJSvNRqlNPoBm6YABE226xrbavY5Wv4=
github.com/PuerkitoBio/goquery v1.10.0/go.mod h1:TjZZl68Q3eGHNBA8CWaxAN7rOU1EbDz3CWuolcO5Yu4=
github.com/ajstarks/deck v0.0.0-20200831202436-30c9fc6549a9/go.mod h1:JynElWSGnm/4RlzPXRlREEwqTHAN3T56Bv2ITsFT3gY=
github.com/ajstarks/deck/generate v0.0.0-20210309230005-c3f852c02e19/go.mod h1:T13YZdzov6OU0A1+RfKZiZN9ca6VeKdBdyDV+BY97Tk=
github.com/ajstarks/svgo v0.0.0-20211024235047-1546f124cd8b h1:slYM766cy2nI3BwyRiyQi/Ud48qiTMtMebDqepE95rw=
```

File: go.sum

After running any package building command, it will install all the packages with specific versions. Then, it will generate a go.sum file to maintain the checksum so, when you run the project again, it will not install all packages again. This is a form of caching.

(source faun.pub)

Key Languages Features [Part 2]

```go
func RunScraper() {
	players := []Player{}

	c := colly.NewCollector()

	// temporary player instance
	var currentPlayer Player

	// extract player team, first name, and last name
	c.OnHTML("div.PlayerSummary_mainInnerBio__JQkoj", func(e *colly.HTMLElement) {
		teamText := strings.TrimSpace(e.ChildText("p.PlayerSummary_mainInnerInfo__jv3LO"))

		teamParts := strings.SplitN(teamText, "|", 2)
		if len(teamParts) > 0 {
			teamName := strings.SplitN(teamParts[0], "#", 2)[0]
			currentPlayer.Team = strings.TrimSpace(teamName)
		} else {
			currentPlayer.Team = teamText
		}

		currentPlayer.FirstName = strings.TrimSpace(e.ChildText("p.PlayerSummary_playerNameText___MhqC:nth-of-type(2)"))
		currentPlayer.LastName = strings.TrimSpace(e.ChildText("p.PlayerSummary_playerNameText___MhqC:nth-of-type(3)"))
	})
```

Uses colly library for performing internet requests for web scraping.

```go
// createBarChart generates and saves a bar chart
func createBarChart(labels []string, values plotter.Values, title, filename string) {
	// Create a new plot
	p := plot.New()
	p.Title.Text = title
	p.Y.Label.Text = "Value"
	p.X.Label.Text = "Players"

	// Create the bar chart
	bars, err := plotter.NewBarChart(values, vg.Points(20))
	if err != nil {
		log.Fatalf("Failed to create bar chart: %v", err)
	}
	bars.Color = plotutil.Color(1) // Assign a color for the bars

	// Set X-axis labels
	p.Add(bars)
	p.NominalX(labels...)

	// Save the plot to a file
	if err := p.Save(12*vg.Inch, 8*vg.Inch, filename); err != nil {
		log.Fatalf("Failed to save plot: %v", err)
	}
	log.Printf("%s chart saved as %s\n", title, filename)
}
```

Uses gonum library for creating graphs.

```go
type Player struct {
	FirstName        string
	LastName         string
	Team             string
	PointsPerGame    string
	ReboundsPerGame  string
	AssistsPerGame   string
}
```

Player struct to contain data for each player

Capstone Project Reflection

Positives

- The application is useful and has practical use cases for us.
    - Fantasy sports, personal interest, etc.
- We learned more about how to structure and initialize (go mod init {module path}) a golang project (general architecture).
- We learned the power of golang and how fast the execution is. (Compared to python for example)

Negatives

- We have been super busy with our classes and external factors that we couldn't spend as much time as we wanted on the project.
    - Add an interactive GUI.
    - Support to customize which fields to order on the graph.
- The projects user-input could be improved upon to make it easier to display the data.

Citation Statement

- This is our own original code
- None of it is borrowed other’s work

Code

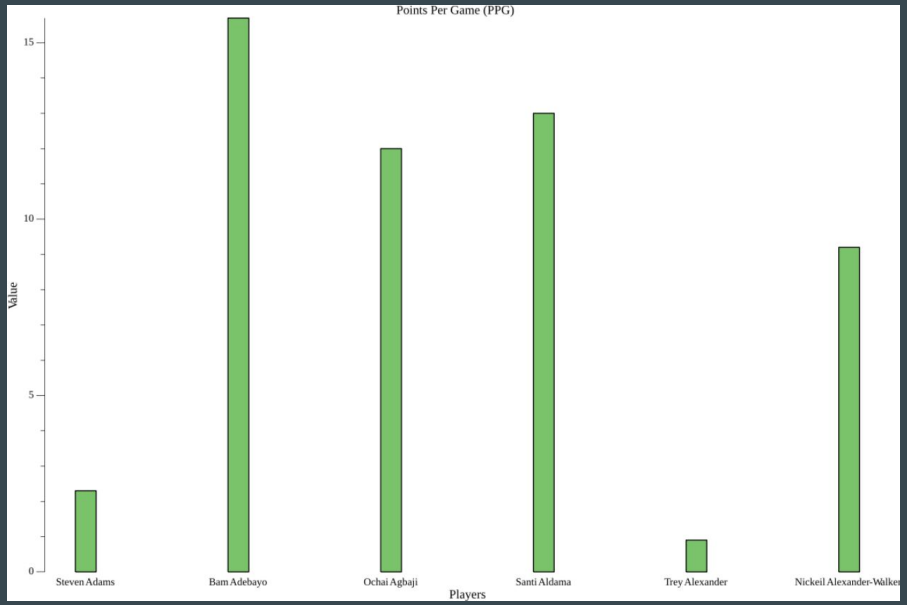
https://github.com/agaroc/GoScrapper/tree/master

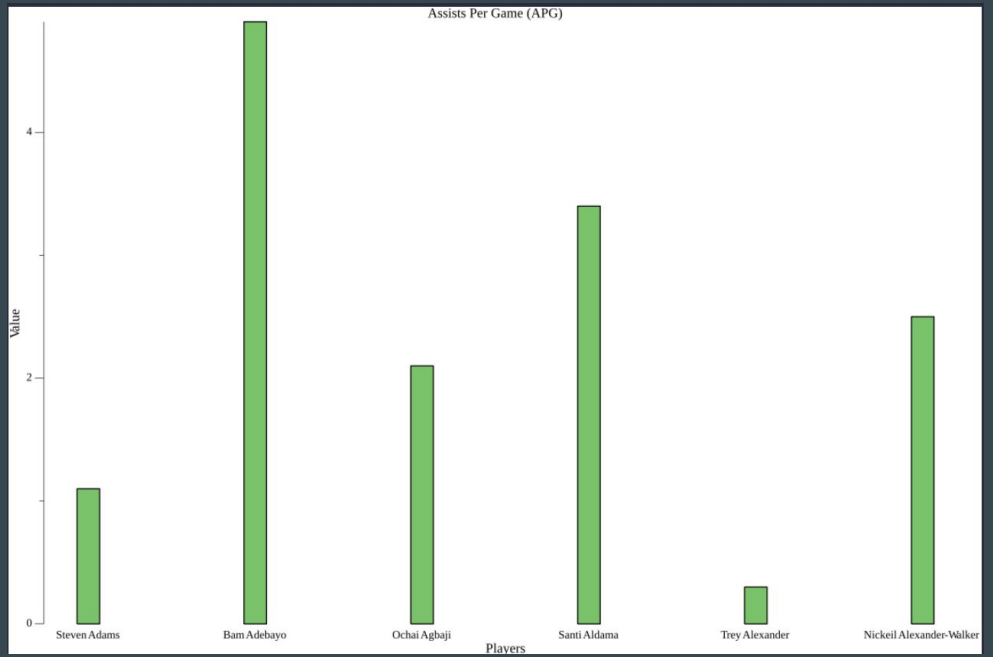
Demo Screenshots

```
Enter NBA player profile URLS from NBA websiter (1 per line). Enter 'done' when ready:
Enter URL: https://www.nba.com/player/203500/steven-adams
Enter URL: https://www.nba.com/player/1628389/bam-adebayo
Enter URL: https://www.nba.com/player/1630534/ochai-agbaji
Enter URL:
Enter URL: https://www.nba.com/player/1630583/santi-aldama
Enter URL: https://www.nba.com/player/1641725/trey-alexander
Enter URL: https://www.nba.com/player/1629638/nickeil-alexander-walker
Enter URL: done
Scraping https://www.nba.com/player/203500/steven-adams...
Scraping https://www.nba.com/player/1628389/bam-adebayo...
Scraping https://www.nba.com/player/1630534/ochai-agbaji...
Scraping https://www.nba.com/player/1630583/santi-aldama...
Scraping https://www.nba.com/player/1641725/trey-alexander...
Scraping https://www.nba.com/player/1629638/nickeil-alexander-walker...
```

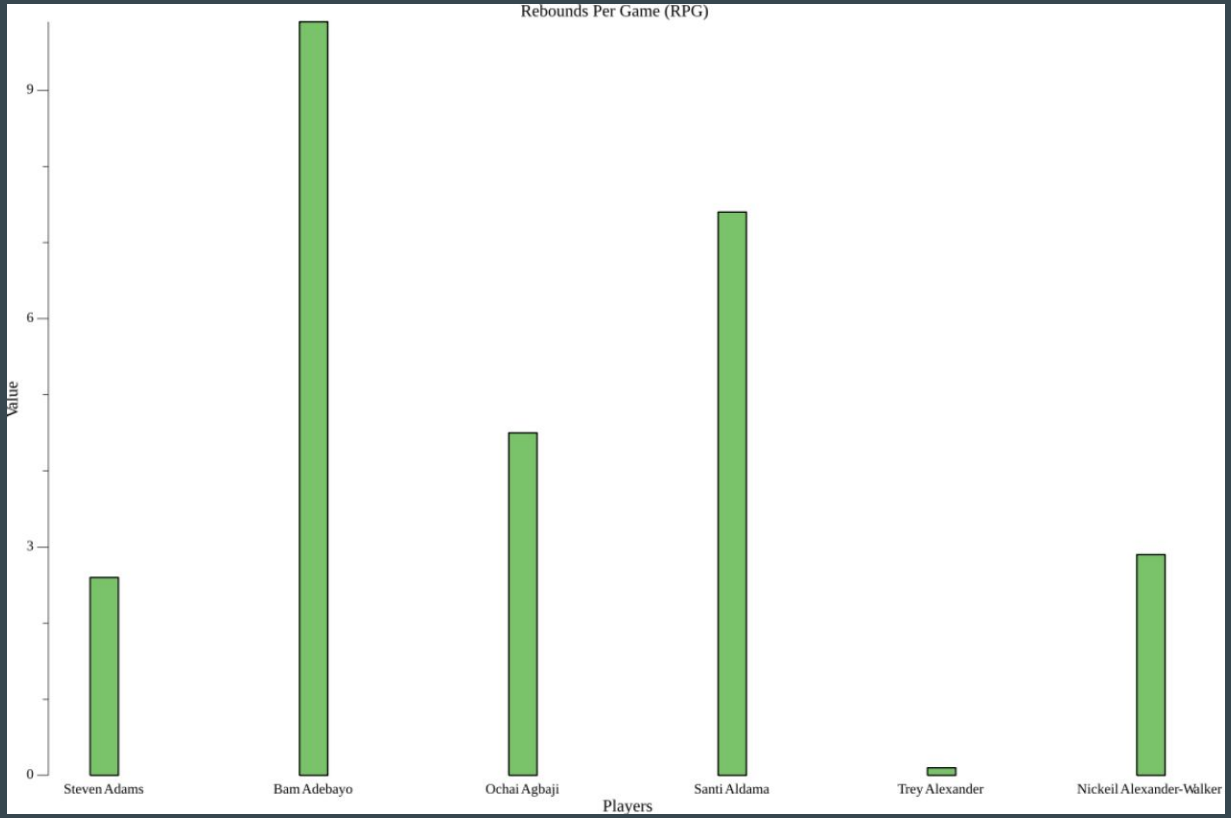
```
Player data successfully written to players_stats.csv
Running data visualization...
2024/12/02 15:04:59 Points Per Game (PPG) chart saved as ppg_chart.png
2024/12/02 15:04:59 Rebounds Per Game (RPG) chart saved as rpg_chart.png
2024/12/02 15:04:59 Assists Per Game (APG) chart saved as apg_chart.png
PS C:\Users\darre\Desktop\CPP\CS 4080 Concepts of Programming Languages\GoScrapper>
```

Demo Screenshots cont.





Demo screenshots cont



Work Cited

Medium: Faun community

https://faun.pub/understanding-go-mod-and-go-sum-5fd7ec9bcc34

Golang Documentation

https://go.dev/doc/

NBA Player Statistic URLS

https://www.nba.com/players