



GDG Golang Korea, DEC 19 2019

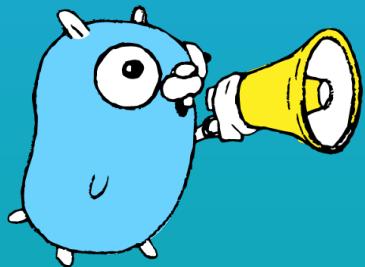
GAME PROGRAMMING WITH Golang



Yujin Park

Freelancer

Game Developer



Today's contents

-
- Golang과 게임 개발의 궁합 01
 - Ebiten 라이브러리 02
 - Game programming 코드 분석 03
 - 학습 Tip 및 레퍼런스 소개 04
 - 게임 개발 분야에서 Golang의 성장 가능성 05
 - Q&A 06
-

Self Introduction

강연 경력

-JetBrains Day Seoul 2018 <Rider와 Unity 사용자 경험>

-GDG Seoul Devfest 2019 <안드로이드 기반 AR세계로의 초대>

-GDG Campus Devfest 2019 <증강현실 개발의 핵심, ARCore!>





SECTION ONE

Golang과 게임 개발의 궁합



1. High performance

2. High portability



Go vs C++

reverse-complement

source	secs	mem	gz	busy	cpu load
<u>Go</u>	3.83	1,782,940	1338	6.67	21% 74% 16% 62%
<u>C++ g++</u>	3.80	979,596	1087	4.95	11% 10% 29% 80%

pidigits

source	secs	mem	gz	busy	cpu load
<u>Go</u>	2.10	8,448	603	2.17	1% 48% 55% 0%
<u>C++ g++</u>	1.89	4,552	513	1.92	1% 100% 1% 0%

fasta

source	secs	mem	gz	busy	cpu load
<u>Go</u>	2.08	3,560	1358	5.61	80% 37% 76% 78%
<u>C++ g++</u>	1.46	2,216	2711	4.40	75% 75% 76% 75%

fannkuch-redux

source	secs	mem	gz	busy	cpu load
<u>Go</u>	17.56	1,524	900	70.00	100% 100% 100% 100%
<u>C++ g++</u>	10.69	1,896	980	42.28	100% 100% 96% 100%

출처: The Computer Language Benchmarks Game





Go vs Python 3

mandelbrot

source	secs	mem	gz	busy	cpu load
Go	5.47	31,088	905	21.77	100% 99% 99% 99%
Python 3	259.50	48,192	688	1,036.70	100% 100% 100% 100%

spectral-norm

source	secs	mem	gz	busy	cpu load
Go	3.94	2,740	548	15.74	100% 100% 100% 100%
Python 3	169.87	49,188	417	675.02	100% 99% 99% 99%

n-body

source	secs	mem	gz	busy	cpu load
Go	21.25	1,588	1310	21.48	0% 1% 100% 0%
Python 3	865.18	8,176	1196	874.96	2% 20% 79% 0%

fasta

source	secs	mem	gz	busy	cpu load
Go	2.08	3,560	1358	5.61	80% 37% 76% 78%
Python 3	63.55	844,180	1947	129.71	40% 71% 33% 61%

출처: The Computer Language Benchmarks Game





Go vs Node js

pidigits						
source	secs	mem	gz	busy	cpu load	
Go	2.10	8,448	603	2.17	1%	48% 55% 0%
Node js	12.42	64,436	530	12.54	100%	1% 0% 0%
k-nucleotide						
source	secs	mem	gz	busy	cpu load	
Go	11.77	160,184	1607	44.52	94%	98% 94% 92%
Node js	25.74	393,728	1812	76.83	74%	86% 68% 70%
fasta						
source	secs	mem	gz	busy	cpu load	
Go	2.08	3,560	1358	5.61	80%	37% 76% 78%
Node js	3.35	61,832	2047	9.40	59%	55% 75% 91%
regex-redux						
source	secs	mem	gz	busy	cpu load	
Go	6.84	340,376	810	9.43	30%	43% 13% 52%
Node js	10.44	787,312	668	12.48	27%	17% 73% 3%

출처: The Computer Language Benchmarks Game



Gomobile is a tool for building and running mobile apps written in Go

Usage:

```
gomobile command [arguments]
```

Commands:

bind	build a library for Android and iOS
build	compile android APK and iOS app
clean	remove object files and cached gomobile files
init	build OpenAL for Android
install	compile android APK and install on device
version	print version



SECTION TWO

Ebiten 라이브러리

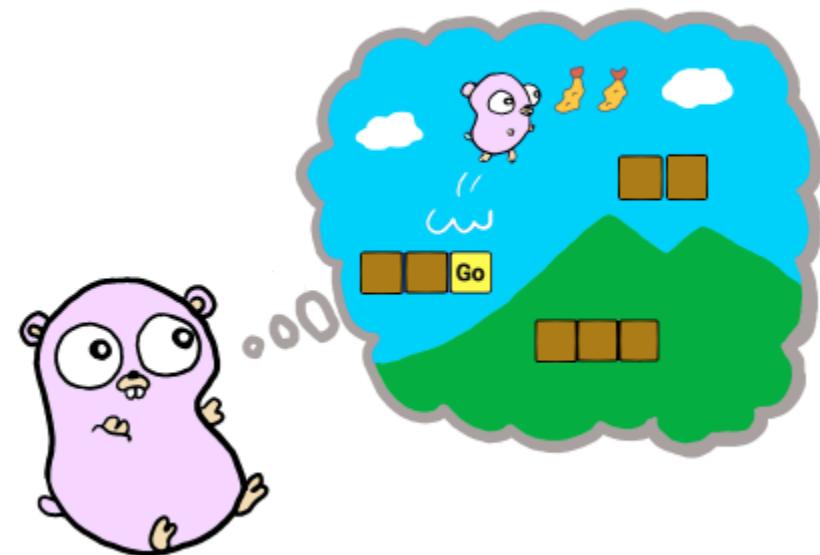
What is Ebiten?

Ebiten

A dead simple 2D game library in Go

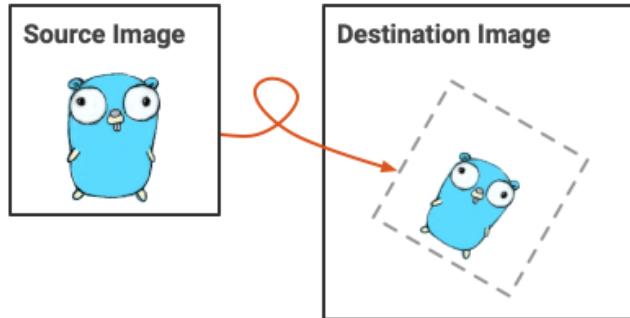
Ebiten (/ebíten/) is an open-source game library, with which you can develop 2D games with simple API for multi platforms in [the Go programming language](#).

 [hajimehoshi/ebiten](#)



Ebiten의 주요 특징 (1)

Dead Simple



Most rendering operations are represented as drawing an image to an image. Everything is an image: the screen, data from an image file, an offscreen are represented as image objects.

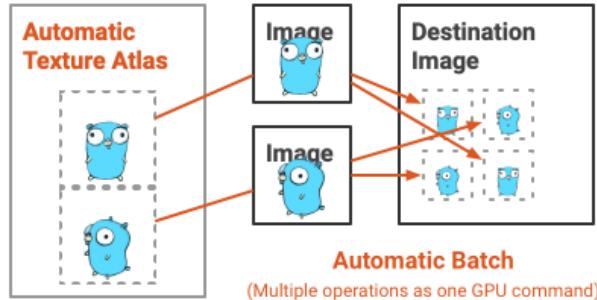
Multi Platforms

Windows, macOS, Linux, FreeBSD
Chrome, Firefox, Safari, Edge
Android, iOS



Ebiten games work on various platforms like desktops (Windows, macOS, Linux, and FreeBSD), web browsers ([GopherJS](#) and WebAssembly), and mobiles (Android and iOS).

High Performance



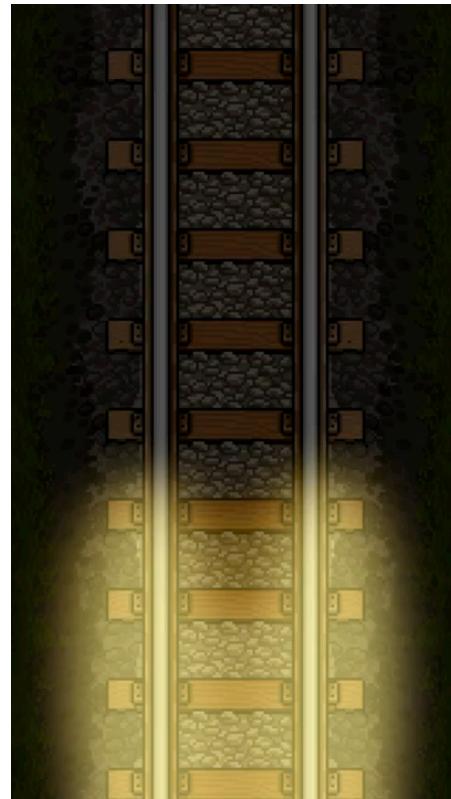
While Ebiten's drawing API is very simple, Ebiten games run very fast with GPU power. Multiple images are integrated into a texture atlas internally. Multiple drawing operations are integrated into a batch automatically when possible.

Production Ready



Ebiten has capability to develop product-level games. [Bear's Restaurant](#), that reached 400K downloads, is a mobile application in Ebiten.

Ebiten's Examples





SECTION THREE

Game programming 코드 분석

Flappy



[Run locally](#)

package, import

```
package main

import (
    "bytes"
    "fmt"
    "image"
    "log"
    "math"
    "time"
    ...
)
```

다른 패키지를 프로그램에서 사용하기 위해서는 import 를 사용하여 패키지를 포함시킨다

const, var

```
const (
    screenWidth      = 640
    screenHeight     = 480
    tileSize        = 32
    fontSize         = 32
)

var (
    gopherImage    *ebiten.Image
    tilesImage     *ebiten.Image
    arcadeFont     font.Face
)
```

변수 키워드 `var`

상수 키워드 `const`



func init()

```
func init() {  
    img, _, err := image.Decode(bytes.NewReader(resources.Gopher_png))  
    if err != nil {  
        log.Fatal(err)  
    }  
    gopherImage, _ = ebiten.NewImageFromImage(img, ebiten.FilterDefault)  
}
```

이미지를 컴퓨터로부터 불러오는 초기화 함수

```
type Game struct {  
    // The gopher's position  
    x16 int  
    y16 int  
    vy16 int  
  
    // Camera  
    cameraX int  
    cameraY int  
  
    gameoverCount int  
}
```

struct는 필드들의 집합체이며 필드들의 컨테이너

필드 데이터만을 가지며 메서드는 가지지 않는다

func jump()

```
func jump() bool {
    if inpututil.IsKeyPressed(ebiten.KeySpace) {
        return true
    }
    if inpututil.IsMouseButtonJustPressed(ebiten.MouseButtonLeft) {
        return true
    }
    if len(inpututil.JustPressedTouchIDs()) > 0 {
        return true
    }
    return false
}
```

사용자의 입력을 받아 캐릭터가 점프하도록 해주는 함수



func Update()

```
func (g *Game) Update(screen *ebiten.Image) error {
```

```
    switch g.mode {
```

```
        case ModeTitle:
```

```
        ...
```

```
        case ModeGame:
```

```
        ...
```

```
        case ModeGameOver:
```

```
        ...
```

```
}
```

```
}
```

각 게임 모드에 따라 해당하는 화면을 출력해주는 함수



func score()

```
func (g *Game) score() int {  
    x := floorDiv(g.x16, 16) / tileSize  
    if (x - pipeStartOffsetX) <= 0 {  
        return 0  
    }  
    return floorDiv(x-pipeStartOffsetX, pipeIntervalX)  
}
```

점수를 계산해주는 함수



func hit()

```
func (g *Game) hit() bool {  
    const (  
        gopherWidth = 30  
        gopherHeight = 60  
    )  
    ...  
    if y0 < -tileSize*4 {  
        return true  
    }  
    if y1 >= screenHeight-tileSize {  
        return true  
    }  
    return false  
}
```

캐릭터가 장애물에 부딪혔는지 확인하는 함수



func drawTiles()

```
func (g *Game) drawTiles(screen *ebiten.Image) {  
    op := &ebiten.DrawImageOptions{}  
    for i := -2; i < nx+1; i++ {  
        // ground  
        op.GeoM.Reset()  
        ...  
  
        // pipe  
        op.GeoM.Reset()  
        ...  
    }  
}
```

게임 배경을 화면에 그리는 함수

func drawGopher()

```
func (g *Game) drawGopher(screen *ebiten.Image) {  
    op := &ebiten.DrawImageOptions{}  
    w, h := gopherImage.Size()  
    op.GeoM.Translate(-float64(w)/2.0, -float64(h)/2.0)  
    op.GeoM.Rotate(float64(g.vy16) / 96.0 * math.Pi / 6)  
    op.Filter = ebiten.FilterLinear  
    screen.DrawImage(gopherImage, op)  
}
```

게임 캐릭터를 화면에 그리는 함수



func main()

```
func main() {
    g := NewGame()
    if err := ebiten.Run(q.Update, screenWidth, screenHeight, 1, "Flappy
Gopher (Ebiten Demo)"); err != nil {
        panic(err)
    }
}
```

게임을 실행시키는 메인 함수



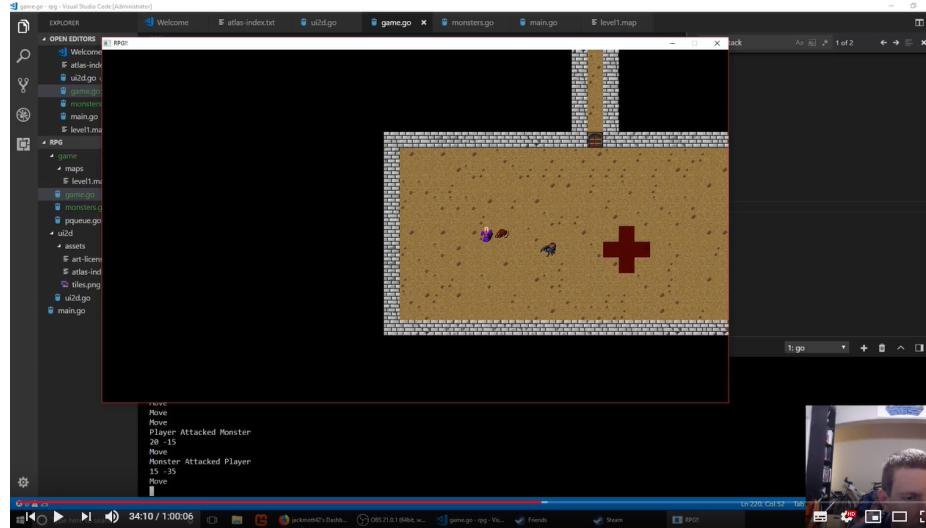
SECTION FOUR

학습 Tip 및 레퍼런스 소개

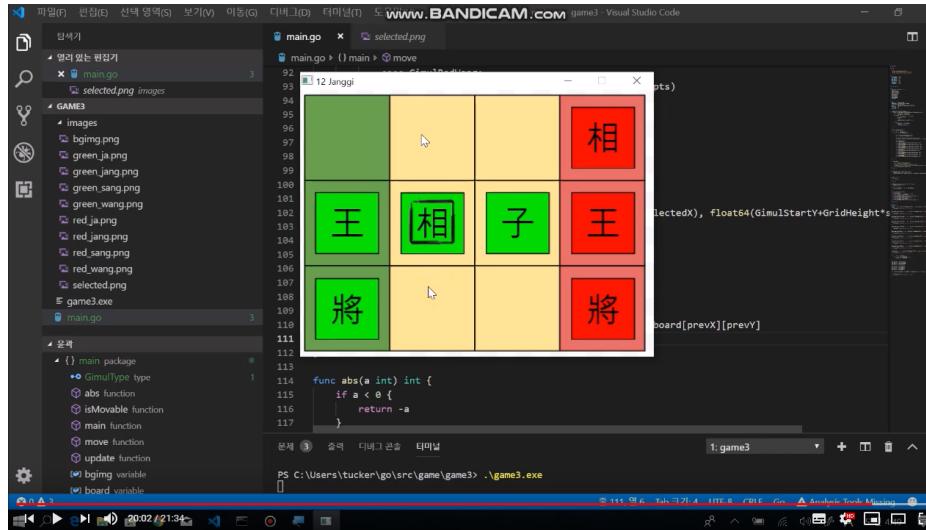
- + 총 7개 episode 보유: Introduction, Graphics, User Input etc...
- + 전반적인 Go언어의 게임 프로그래밍 기능 소개



- + 총 46개 episode 보유: UI, Sound, OpenGL etc...
- + RPG 게임 기반으로 다양한 게임 요소에 대해 Go언어로 개발하는 법 소개



- + 총 10개 episode 보유: 12장기, 그림 퍼즐, Run Game etc...
- + 다양한 게임을 만들면서 배울 수 있고 한글로 된 설명이 장점!





SECTION FIVE

게임 개발 분야에서 Golang의 성 장가능성



1. 게임 관련 Library 부족

2. Golang 게임 개발 know-how 부족





SECTION SIX

Q&A



“



Thank you!

”

Yujin Park
eugene5802@gmail.com

