

개발 결과 요약

이 프로젝트가 가져다 준 새로운 지식

- Github 사용법
- Shell 사용법
- Javascript(Node.js,Express.js) 코딩
- HTML 코딩
- MongoDB, DB 설계와 사용
- 수업시간에 배웠던(HTTP) 내용들이 실제로는 이렇게 쓰이는구나!

중간 개발 현황

- Express.js를 이용해 개발한 웹 서버 & 웹 애플리케이션 띄우기
- CSS를 활용해 Nav-bar 만들기.
- HTML 태그 배워서 적용 - 유튜브 링크 띄우기
- 게시글 작성 및 저장 : Client - Server 간 HTTP를 통한 통신!!
- Domain 등록 안되어 있어서 localhost:3000으로 접속해야하는 단점.

추후 과제

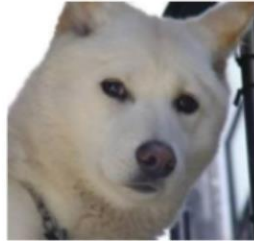
- 새로운 오픈 소스 : Bootstrap 배우고 디자인 적용
 - 인터페이스 배치 제안서 디자인대로 변경
- 라우터 및 서브 도메인 공부 → Admin 전용 페이지(글쓰기) 구현
- <게시글 작성 / 보기 기능> 더 디테일하게
- 도메인 등록 & 호스팅
 - 돈이 필요해서 방법을 찾는 중 → 경희대 아마존?



YouTube <https://youtu.be/CxSN7q4Kf8s>

중간 데모 발표

“내가 직접 활용할 홈페이지 ”



백구군

(gjwodnr3454)

20년 후, 나는 E3에서 있다.

EDIT

프로필 ▸

포스트쓰기 관리·통계

category

..

검색

전체보기 (370) EDIT

공지

다른사람의 게임

공부/개발 ▾

가난한 기숙사생의 식단

음악

나

글쓰기

시

기타 ▾

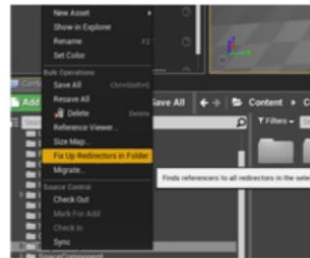
Unreal (73)

목록열기 ♡



[ProjectZWorld] 10.10 - 애니메이션 적용

2017. 10. 11. 9:12



[ProjectZWorld] 9.30 - 리팩토링

2017. 10. 2. 21:24



넥슨 갔는데. [2]



[ProjectZWorld] 9.14 - Creating First Demo

2017. 9. 15. 0:49

[ANIMATION STORE](#)[GAMES](#)[BLOG](#)[CONTACT](#)[ABOUT](#)[TUTORIALS](#)[KUBOLD YOUTUBE](#)[UNITY ASSET STORE – FAQ](#)[UNREAL MARKETPLACE – FAQ](#)

LONGSWORD ANIMSET PRO FOR UNITY

Buy on Asset Store - all net income is going directly to charity



Kubold is a game mocap animations producer and an indie games developer. Please visit the animation store (link above) to buy royalty-free, game ready mocaps for Unreal Engine, Unity, or other engines. You are also welcome to try my free games.

LONGSWORD ANIMSET PRO FOR UNITY MADE \$2,485 FOR CHARITY TO THIS DATE

BY KUBOLD.COM

17/09/2017

BEZ KATEGORII

WhiteDog

Games & Development

You
Tube

Games

Project Z
World

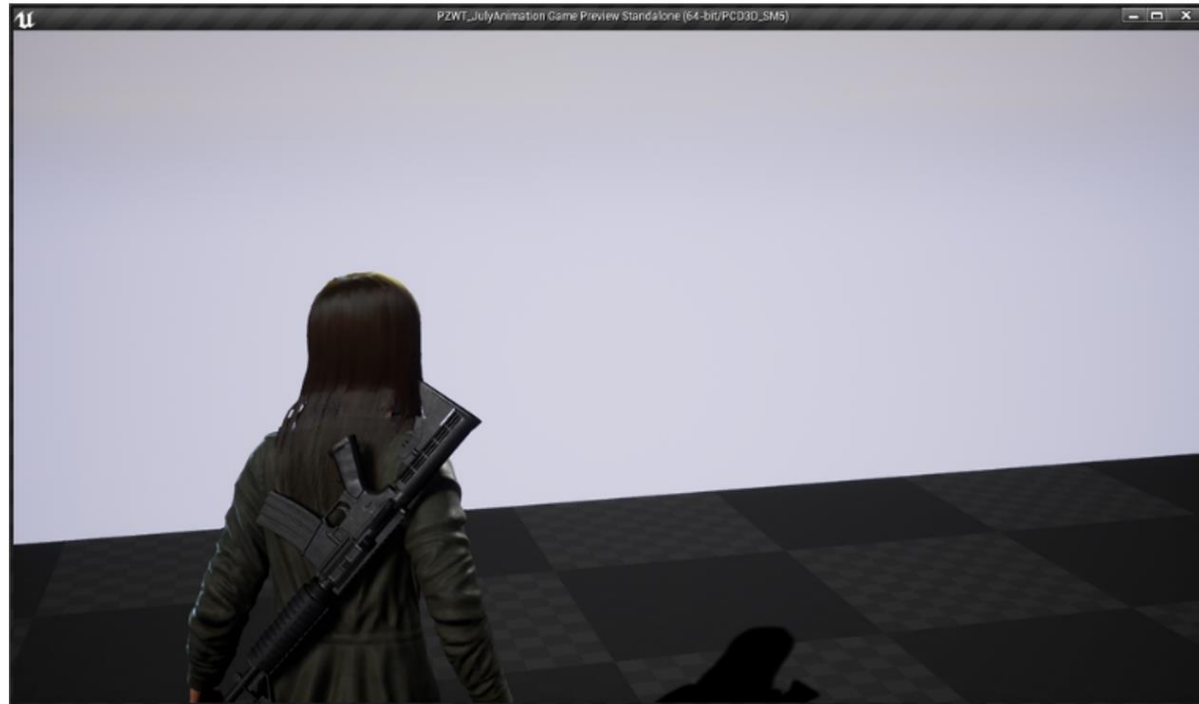
Contact

About

WhiteDog
Youtube

Character Animation and Statemachine

15/10/2017



Inside the project, we are currently working on Smooth character animation and make it as a playable character.

아는 것 하나도 없음.
그래서 모든 것이 새롭게 배우는 것이었다.

Shell (Command Line)


```
MINGW64:/d/W/Project

--single-branch      clone only one branch, HEAD or --branch
--no-tags            don't clone any tags, and make later fetches not to fo
allow them
--shallow-submodules any cloned submodules will be shallow
--separate-git-dir <gitdir>
                    separate git dir from working tree
-c, --config <key=value>
                    set config inside the new repository
-4, --ipv4           use IPv4 addresses only
-6, --ipv6           use IPv6 addresses only

WhiteDogDESKTOP@DESKTOP-56HOLPB MINGW64 /d/W/Project
$ git clone https://github.com/WhiteDog123/ComputerNetworkProject.git /d/W/Project
fatal: destination path 'D:/W/Project' already exists and is not an empty directory.

WhiteDogDESKTOP@DESKTOP-56HOLPB MINGW64 /d/W/Project
$ git clone https://github.com/WhiteDog123/ComputerNetworkProject.git
Cloning into 'ComputerNetworkProject'...
remote: Counting objects: 8069, done.
remote: Compressing objects: 100% (5716/5716), done.
Receiving objects: 91% (7343/8069), 8.07 MiB | 1.18 MiB/s
```

GitHub

 This repository Search Pull requests Issues Marketplace Explore

WhiteDog123 / ComNetPrj

Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

컴넷 과제를 위해 홈페이지를 만들면서 이것 저것 공부중 [Edit](#)

[Add topics](#)

18 commits 1 branch 0 releases 1 contributor

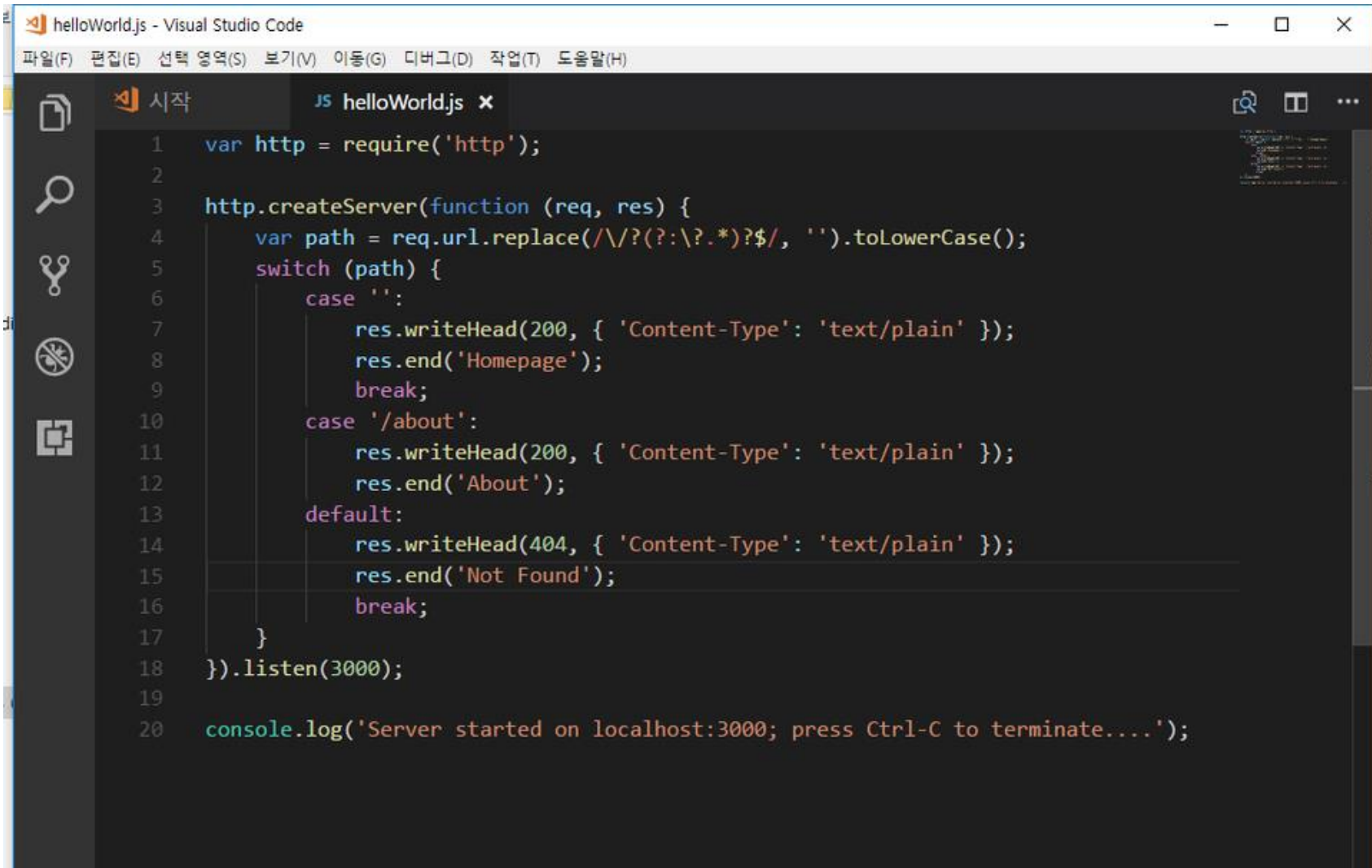
Branch: master New pull request Create new file Upload files Find file Clone or download

WhiteDog123 Add routes, and mongoose! Latest commit 16b97e0 3 hours ago

lib	I Added logo on the topside	a day ago
models	Add routes, and mongoose!	3 hours ago
public	I Added logo on the topside	a day ago
qa	Added test codes and grunt	4 days ago
routes	Add routes, and mongoose!	3 hours ago
views	Add routes, and mongoose!	3 hours ago
.gitignore	I deleted the files again! and fix .gitignore	a day ago
Gruntfile.js	Added test codes and grunt	4 days ago
meadowlark.js	Add routes, and mongoose!	3 hours ago
package.json	Add routes, and mongoose!	3 hours ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

Javascript (Node.js, Express.js...)



```
helloWorld.js - Visual Studio Code
파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 디버그(D) 작업(T) 도움말(H)

시작 JS helloWorld.js x

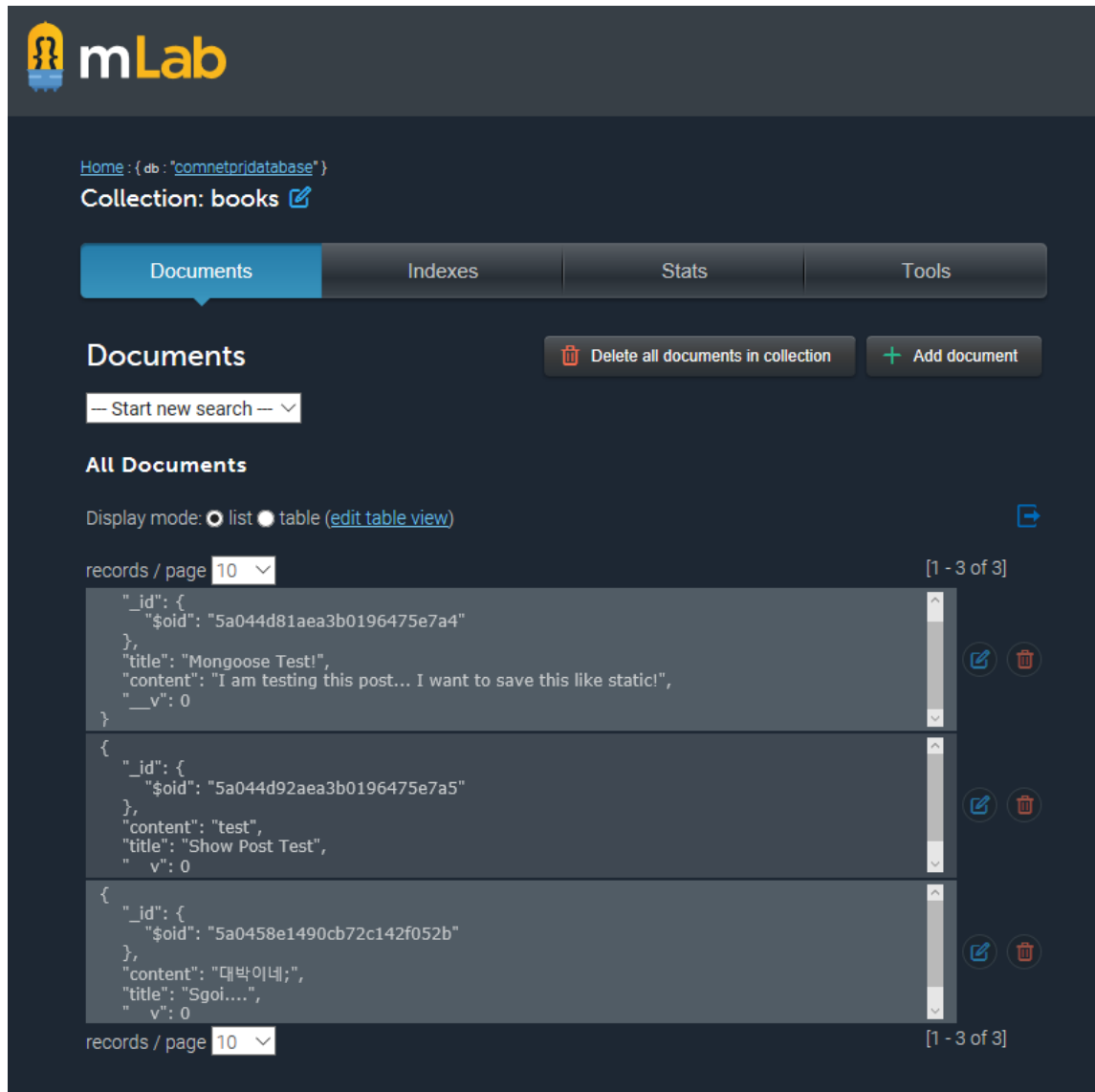
1  var http = require('http');
2
3  http.createServer(function (req, res) {
4      var path = req.url.replace(/\/?(?:\?\.*)?$/, '').toLowerCase();
5      switch (path) {
6          case '':
7              res.writeHead(200, { 'Content-Type': 'text/plain' });
8              res.end('Homepage');
9              break;
10         case '/about':
11             res.writeHead(200, { 'Content-Type': 'text/plain' });
12             res.end('About');
13         default:
14             res.writeHead(404, { 'Content-Type': 'text/plain' });
15             res.end('Not Found');
16         break;
17     }
18 }).listen(3000);
19
20 console.log('Server started on localhost:3000; press Ctrl-C to terminate....');
```

HTML & CSS

```
</style>
{{! HTML 폼은 브라우저와 서버의 통신하는 거}}
<form action = '/upload' method = "post">
<div>
|  {{!<label for="title">Title:</label>}}
{{! 서버와 데이터를 주고 받을때, 데이터 이름 지정은 name으로 한다.}}
|  <input type="text" placeholder = "Title" id="title" name="post_title"/>
</div>
<div>
|  {{!<label for="post">Post:</label>}}
|  <textarea id="content" placeholder="Write Post Here" name = "post_content">
</div>
<div class = "button">
|  <button type="submit">Upload Post</button>
</div>
</form>
```

```
<style>
  #datas
  {
  }
  div{
    background-color: #333;
    padding-top: 5px;
    padding-bottom: 5px;
  }
  /*신박하네.. 이렇게 하나하나 아이디로도 할수있구나.*/
  #txtTitle{
    color : white;
    text-align: center;
  }
  #txtContent{
```

DB (Mongo DB)



The screenshot shows the mLab MongoDB web interface. At the top, the mLab logo is visible. Below it, the current database is 'comnetoridatabase' and the selected collection is 'books'. There are tabs for 'Documents', 'Indexes', 'Stats', and 'Tools'. The 'Documents' tab is active, showing a list of documents. A search bar is present with the text 'Start new search'. Below the search bar, the 'Display mode' is set to 'list'. The 'records / page' is set to 10. The interface shows three documents in the collection:

Document
<pre>{ "_id": { "\$oid": "5a044d81aea3b0196475e7a4" }, "title": "Mongoose Test!", "content": "I am testing this post... I want to save this like static!", "v": 0 }</pre>
<pre>{ "_id": { "\$oid": "5a044d92aea3b0196475e7a5" }, "content": "test", "title": "Show Post Test", "v": 0 }</pre>
<pre>{ "_id": { "\$oid": "5a0458e1490cb72c142f052b" }, "content": "대박이네;", "title": "Sgoi....", "v": 0 }</pre>

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

//스키마는 document구조가 어떻게 생겼는지 알려주는 역할
var bookSchema = new Schema({
  title:String,
  content:String,
});

//모델 정의
//모델은 데이터베이스 데이터를 읽고, 생성하고, 수정하는 프로그래밍 인터페이스.
//아하. 그래서 모델형 변수를 생성하면 데이터를 내가 직접 쓸수있는거구나.
//데이터 가지고 놀 수 있는 프로그래밍 인터페이스니까.
//첫번째 인자는 document가 사용할 collection의 단수적 표현?
//collection이 뭐냐. 내 생각에 한 덩어리 한덩어리 인듯, 테이블.
var Book = mongoose.model('book', bookSchema);
var book = new Book({
  title : "Mongoose Test!",
  content: "I am testing this post... I want to save this like static!",
});
//데이터 베이스에 저장
/*
book.save(function(err,book){
  if(err) return console.error(err);
  console.dir(book);
})*

//모델을 모듈화
module.exports = mongoose.model('book',bookSchema);
```

수업시간에 배운 내용들(HTTP)

```
</style>
{{! HTML 폼은 브라우저와 서버의 통신하는 거}}
<form action = '/upload' method = "post">
<div>
```

```
app.use(function (req, res, next)
  res.status(404);
  res.render('404');
});
```

```
//전역변수 추가. 모듈 밖에서도 사용할 수 있게.
exports.getFortune = function() {
  var idx = Math.floor(Math.random() * fortunes.length);
  return fortunes[idx];
};
```

배운게 많다.
그래서 배운거 가지고 뭐 만들었니?

“클라이언트가 게시물 작성 & 서버로 전송, 저장”

동적 콘텐츠 생성...

유튜브 링크...

Navigation Bar...

데모영상

 **YouTube** <https://youtu.be/CxSN7q4Kf8s>

최종 때는...

“Admin 전용 페이지랑 유저 페이지 구분!”

“디자인, 제안서대로 꾸미기 ”

“게시글 작성 & 보기 디테일”

“도메인 등록 & 호스팅”

질문