Join us in #EquusOnline on irc.canternet.org!

Beta website: http://blue-chaos.net

We are still <u>very early</u> in development, we are currently working on the basic storyline, gameplay mechanics, and getting a working prototype. Game development takes time, don't expect a playable alpha in two weeks.

(Notice: We need a new logo! If anybody is interested, please contact PM notpinkipie)

WHAT IS EQUUS ONLINE?

Equus Online is a My Little Pony: Friendship is Magic inspired MMORPG. Initially planned to be created using HTML5 and WebGL, further research into the limitations of the platform revealed that this system is horribly inefficient for a project as ambitious as an MMO. Instead, the project has been moved to Unity 3D for the client, and the server is being coded 100% from scratch in C++. Do not assume that Unity3D is a bad platform due to some games that have been made with it - Unity is actually an extremely nice and full-featured engine/IDE.

WHAT WILL IT BE LIKE?

Our gameplay style is a blend a fairly unique graphics style of the traditional JRPG and the modern MMORPG. Using and extremely lightweight client/server system, this game will run on almost any computer on very low-speed Internet connections - even a phone signal!

HOW DOES IT COMPARE TO [insert other pony MMO here]?

We support all projects and do not wish to compete against anyone, but many other projects seem to be more focused on being kid-friendly or a social experience. While we

still aim to be somewhat kid-friendly, we are putting much more focus on the gameplay and storyline than we are on social interaction. The game will still be an extremely social experience, but more heavily driven by the story and gameplay itself. The story is designed to have depth, as well as intense emotion and heavily atmospheric cutscenes. Despite our unconventional graphics style, we also put a large amount of effort in the graphics and immersion. This game will be challenging, fun, immersive, emotional, and long enough to keep you sustained for months.

WHAT HAPPENS WHEN WE FINISH THE GAME?

We'll do our best to include as many side activities as possible, but the game will continue to be maintained and updated after release. Not only will this be in small patches, but also in the form of large expansion pack-type updates. Even if you manage to finish all the quests in the game (which there will be a LOT of), there will still be more in the future.

CAN I HELP? WHEN WILL THE GAME BE READY?

The game will be ready whenever it's ready. This could be months, or even years. We do plan on having a public alpha, however keep in mind that during alpha, the game is NOT complete and may have many bugs. The purpose of a public alpha is to test the game against REAL players to better iron out any possible bugs. This also means you'll need to report any bugs you find! And about helping - there's a table of open positions down below, and my contact info. Go take a look!

WILL YOU GUYS MAKE ANY MORE GAMES?

Our plan is to focus entirely on Equus Online until release, and while we will still update EO for as long as it has players, focus will be shifted to other games. This will allow us to make a profit from our games, since they will be our own intellectual property. EO will remain 100% free and active as long as there are players.

UPDATE: A smaller sub-project has been founded by a team member. It will be a story-driven singleplayer RPG written in Java. Progress and release date is unknown at this point.

UPDATE 2: A game is currently in development for the My Little Game Dev Mareathon, which will end on December 30th, 2013. We will be releasing it as an HTML5 browser game and a free app in the Google Play Store for Android phones. We will continue adding features even after the mareathon, and if the game receives enough attention, we may continuously update it. *(progress has slowed, Christmas and other various life activities have hindered our progress. Even if we can't finish in time for the MLGD*

Mareathon, we will continue working on it.)

TEAM MEMBERS
<u>very out of date</u> - if you feel you should be on this list (or taken off), contact a project manager

Project Managers
Cameron Seid - notpinkiepie
Game designer and scripter, project manager
wincam97@gmail.com
twitter.com/WinCamXP
Texas, USA
Skype name: wincam98
IRC name: WinCamXP

Jacob Pedersen - frash23
Game designer and scripter, website designer, project manager
frash23@cotse.net
Denmark
Skype name: frash23

Organization Manager
Ruben Calvo - ruben.sandwich
Team and project organizer, efficiency manager
ru13encalvo@gmail.com
Skype name: ruben.sandwich

Music Designers
Sonorous Gravity
Lancaster, CA
Music designer
sonorous.gravity@gmail.com
Skype name: sonorous.gravity

Lars Dobbertin-King - LYNKD
Music designer
larztheloser@gmail.com
Skype name: lynkd_ltd

Sound Designers
Jengah
Sound Designer
jengahmo@gmail.com
Skype name: jengahmo

Programmers
P. C.
C++ & server developer

Graphics Designers
Scott Borg
Concept artist

scottborg@outlook.com
Skype name: live:scottborg

Nag Prince(ss)

Graham - mrgraham626
Canada
Skype name: mrgraham626

Drew Bogan - Captain Miracles
Texas, USA
Skype name: captainireland7

ShowStopper
New Jersey, USA
Skype name: showstopper.ladytothian

POSITIONS TO BE FILLED (May extend onto next page)

Contact notpinkiepie to apply for a position. If you have any questions or would like to propose another role, please contact a project manager for info. You may apply for more than one position. It would help tremendously if you already have some of your work to show us. The "amount needed" column will tell you if there are any available positions - this may change with time. **Positions in bold and italics take a higher priority**

NAME	DESCRIPTION	SPECIFIC TASKS	AMOUNT NEEDED
Server developer	Codes the server software in C++	Client/server interaction, script interpreters, file I/O, misc. operations	1-2
Client developer	Codes the client software (C# and/or Javascript via Unity)	Client/server interaction, script interpreter, game mechanics	1-3
Concept art designer	Creates game concept artwork to illustrate landscapes or new features	Follow criteria given by the game designers to construct concept artwork. Most likely will not be used in actual game - this is *concept* artwork	1 for characters 1 for landscapes 1 for GUI/gameplay (you may apply for several)
GUI designer	Creates sprites and textures for ingame GUI	Follow criteria and concept art to create the artwork to be used ingame for GUI elements	1
Tile/terrain designer	Creates textures to be used in the game's landscapes and terrain	Follow criteria and concept art to create ingame artwork for terrain, tiles, and other landscape-related textures	1-2
Character designer	Creates the ingame artwork for clothing and armor, NPC's, players, and other characters	Creates sprites for characters, mane/tail styles, clothing, armor, etc using a layering system.	1
Sound designer	Creates sound effects	SFX such as button presses, mob sounds, attack sounds, etc	1-3
Music designer	Creates game music	BGM such as title screen	(Currently full)

		music, area BGM, cutscene or battle music, etc	
Prince(ss) of Nagging and Productivity	Makes sure nobody quits the project without good reason, helps playtest the game if needed	Check up on ALL team members via skype as often as possible to ensure maximum productivity **This is the most important role.**	2
Representative	Helps advertise and distribute the game	Helps find new team members, brings publicity to the game	1-2
Storyline Writer	Helps create storyline	Collaborates with all team members to create storyline elements such as questlines Remember - this is not the same as a fanfiction!	1-4

Notice to GUI Designers, Tile/terrain Designers and Character Designers: These positions have been split from "Graphic Designer". If there is any mention of a "Graphic Designer" below, keep in mind it was written before the change and alert a project manager to update it.

Notice to all potential team members: Due to the fact we are very early in development, many positions will not be needed until later on. You may sign up now, but depending on your task, you may have little to no work until later in development.

Everything below this point is planning and brainstorming. Some material may be missing or incomplete as most brainstorming is done via Skype chats. Also, not all brainstorming may reflect changes made to other ideas and concepts if it was written before the change. And finally, anything may be changed or removed at any time.

Do not take anything below this point as guaranteed, nor should you expect to see everything down here in its current state. This is merely a peek into various ideas and concepts thrown around by the team in pre-development

phases.

Current tasks, generally in order of importance:

1. Script interpreter and basic gameplay mechanisms (create working prototype)
2. Server/client interaction
3. Recruit others to help with project
4. Plan out storyline and maps in gameplay format
5. Start implementing actual gameplay features
6. Begin work on game world, ingame graphics

Features to be implemented later, mostly in order

1. In-game server connection
2. Multiplayer base (sharing X,Y,rot etc between all clients)
3. Ranks (user, moderator, admin etc.)
4. Map generator

STORY LAYOUT

(the following two paragraphs are merely a concept - it may be changed or scrapped)

The main idea that makes the game unique, is the way the story and time is going to flow.
The plan is to run the game in "Seasons". Each season contains a main goal of the game, i.e. "Save Princess Celestia", or "Save the world from villain X", which is an actually reachable goal. We are going to do our best laying the "final quest" out, so it takes about a year from going from newly registered player, and able to beat the quest.

Once the main goal of the season is beaten, the time where "everything is mainly peaceful (no evil villain or anything)", the developers write the storyline and quests, new NPC's, changed maps etc., so time actually flows, instead of most RPG's that stay still. There will still be activities to do during the period between seasons, or you can always just make a new character and start over.

(end concept)

You will be able to go back to storyline quests either to help out other players, or just to play them again - playing your OWN quests again is only unlocked after you have completed the

main storyline, and you can only help players who have not yet completed the story IF you are within the correct level range.

Another note, the idea is that the final quests should be unable to be beaten by one person - cooperation is required. This helps enforce the idea and concept of friendship and cooperation, conveying the lessons taught by the show. Also, maybe make the whole "impossible to beat alone" concept subtly spoken throughout the game.

GRAPHICS STYLE

1-Point Perspective, 2D sprites/tiles - think "Don't Starve" or "Paper Mario"-ish style, no 3D models
3D XYZ plane, for elevated pathways and similar things (elevated pathways may possibly be removed, however depth/hills is still a priority) - will probably NOT be pixel art, however will require animation.

COMBAT SYSTEM

Combat in Equus Online is much more challenging and fast-paced than most other MMO's. In EO, the combat takes place realtime in the open world. Multiple players can easily participate in a fight, or watch from afar. Instead of having a random chance for an attack to miss, an attack hits *if the attack hits*. What this means is the combat, especially ranged combat, will miss if the target moves out of the line of attack. Combat in Equus Online is very fast-paced and challenging, requiring constant attention. Not only is the player attacking, they have to determine which attacks are best to use, fire them without missing, try to dodge the enemy's attacks, and consume healing items to perform well in a battle. This also allows for very engaging team combat, putting focus on cooperation and coordination.

Stats, skills, and similar calculations have not yet been properly determined or planned.

POSSIBLE MAP DEPTH SYSTEM

In the tile data (see below), there is an attribute on each tile for "depth". Standard ground tiles (grass, pathway, etc) are seen by the client as "hill-able" (such english). This means that when two "hill-able" tiles are placed next to each other and set to different heights/depths, the client will render it at an angle, creating a hill. (Smoothing optional, but possible - maybe graphics setting for level of terrain smoothing?)

Obviously the clientside movement would need to adapt to the change in height, however the server may not actually need to process depth (unless we do have elevated pathways)

<u>SOFTWARE, LIBRARIES, RESOURCES, ENGINES, ETC CONSIDERED</u>

Map of Equestria: http://tiny.cc/bta9yw

UNITY:
We've decided to switch from HTML5/WebGL to Unity. Unity has many more development features, is much better optimized, has better network protocols, et cetera... previous plans remain only for documentation. Yes, it will still be cross-platform. It actually supports the usage of three languages (Javascript, C# and Boo), even in the same game - they all work in harmony. http://unity3d.com/

(below text remains only because of original plans - please disregard)

GAME MAKER STUDIO (Client only): *(See next paragraph)*
Javascript/WebGL based 3D engine: http://threejs.org/
It's definetly possible: http://tiny.cc/86oezw (please don't use this, this is just proof of concept)
Making a javascript based extension: http://tiny.cc/szgdzw
In-game Text Editor (admin tool): http://tiny.cc/kygdzw

WHY WE ARE USING GAME MAKER:
Game maker gets a bad rap for being "cheap", "too easy", "crap", "incapable of ___", and various other insults. What people fail to realize is that many of these insults began due to a portion of its community, and not the program itself. In addition, the program has improved dramatically in recent years compared to its primitive ancestors. The current release of Game Maker Studio supports exporting to many platforms including HTML5 and Android, among others. It also allows the usage of Javascript libraries as a form of plugin, enabling us to fully utilize HTML5/WebGL's capabilities. A large portion of the game will be coded from scratch, merely using GMS as a platform to help organize development and simplify certain tasks.

And, to clarify, **Game Maker is only being used for the client - the server will be coded 100% from scratch.** The previous paragraph fully explains why.

MISC:
derp

ANDROID/iOS PORT:
Supported by Unity, just requires alternate GUI. Android support will be guaranteed from the start, iOS will not because iOS sucks

<u>SERVER OPERATION</u>

Maps and scripts:

Server sends client initial map data upon player joining. Client requests other maps when the player travels into or near them. Map data stored in text files; each tile has multiple variables that specify things such as floor type, special objects, etc - interpreted by client

Client also requests scripts when needed - the scripts contain instructions for both server AND client, each side knows what to ignore and what to execute. (This means script interpreter on the server)

Script file structure and operation

(NOTICE: Has been changed somewhat, will be updated soon):

Scripts are separated into two types: root scripts and reference scripts.

ROOT SCRIPTS
Root scripts serve as the "base" or "spawner" for characters, NPC's, signposts or other "special" objects with more importance than a basic floor tile. They are also used as an identifier. In the map's tile data, each tile may specify a root script. The root scripts are organized like this:

A\
      001.txt
      002.txt
      003.txt
B\
      001.txt
      002.txt
C\
      001.txt

...and so on. In the map tile data, it references the script as "A001", referring to the script "001" in the folder "A". The scripts are categorized so that A means NPC, B could mean doorways, C could mean mobs, and so on. It's a form of manual organization by the developers, meaning the actual categorization is handled manually by the devteam, but it's easy to implement and use.

The root scripts are essentially what "creates" a character, NPC, or object. It defines things such as sprite, armor, health, and various similar traits. For functions like shop vendors or other special operations, the root script will call upon a reference script like a function in programming.

REFERENCE SCRIPTS
A reference script is like a class in many programming languages. A separate script points towards a reference script, saying "use this". For example, in a shop vendor NPC's root script, the script will make the NPC greet the character. After the NPC greets the character, the root script calls upon the "vendor" reference script. The root script contains data for items being sold and their prices, but the vendor script takes over. The vendor script contains all the instructions for the game to open a window with the available items, all the buttons and mechanics, etc. When the vendor window is closed, therefore ending the vendor script, the root script continues where it left off.

IDENTIFIER SYSTEM
There are two important uses for this identifier system. The main use is for scripted events like cutscenes. Cutscenes often use more than one character and object, constructing a scene. Instead of using some kind of complicated system where the scripts communicate with each other, instead one script can "override" another. Every NPC is generated from a specific root

script, exclusive to that NPC. This means the NPC can be identified by the root script from which it was created. A completely different script can tell the game "Make the NPC from script A023 move to the left". Obviously this will not work if said NPC does not exist, but it works like this: When one script tries to "override" another, it is actually using a reference script to perform this. It tells the reference script what ID and action to perform. The reference script then makes sure that is actually possible, and then proceeds if it is.

Now, for things like monsters, we don't want them all to be identical. We want to have some variance in their health and stats. Instead of creating separate scripts for each and every one, why not just have the root script do a little randomizing? 23% chance of this sprite, 33% of this other sprite, etc. This is helpful for variance in mobs, but what about a situation where we try and override another mob, of which there are several? Instead of doing something crazy like moving them all at once, or simply crashing, each of the mobs has a sub-ID. When the root script is executed, it is executed again and again for each instance of the same script. When each of them executes, it checks to see if another of itself already exists. If so, it checks how many. Each of these mobs' sub-ID's work like this: A001:01, A001:02, A001:03, and so on. To make this system predictable and easy to manage, these ID's are assigned in a left-right top-bottom format. Even if there is only one of a given script on the map, it still has the sub-ID :01. If a script references another script, and supplies no sub-ID, it will default to sub-ID :01. This way, in cutscenes that have several of the same mob, it is easy to distinguish between them.

Chat:

Server receives message from client with chat message. Send message out to all other clients (super basic and simple)

Anti-cheat, lag kicking:

Pings that are too high are automatically kicked, to prevent movement speed exploits or lagswitching

For movement checking, the server ALSO interprets the map file, but only enough to make sure the player is not inside walls/flying

CONTROLS, GUI, AND GENERAL OPERATION

Left-click to move character, interact, or standard melee attack.
Right-click to use currently selected spell/ability.

CONCEPT ART - Dear Devin Aldrich...TAKE MY UNBORN FUTURE CHILDREN
(This image is merely a mockup of the potential GUI system. It will most likely be very different from the actual game in various aspects, the point was to establish a layout)

Screen elements visible in mockup: "12: ...sting, Testing, 1...2...3... This is probably a drill. I repeat, this is probably a dri"

Chat log (14): "Dasheh: okay I'll move first / TwiSprkl: Wrong way, Dash... / Dasheh: wait how do pawns move / i don't play chess / TwiSprkl: Rainbow this is what I / have to put up with. EVERY DAY."

Location info (15): "Checkerboard / Levels 0-10 / Neutral Territory / A debug area. Totally fun to play / pony chess with friends here."

Targeted mob info (17): "Albino Changeling / HP: 97/110 / MP: 200/200 / Friendly / Abnormal changelings are often / exiled from their hives, due to t..."

1-5: Spell/ability hotbar. Click to select active spell.

6-9: Options and stuff. Possibly Options, Spellbook (name change), Friends List, Character (contains stats, armor/equip screen)

10: Picture of character, just because.

11: Character's current overall level

12: Scrolling/marquee "news" panel - displays info regarding game updates, news, etc (can be disabled, option between blank panel or transparent)

13/18: Health and Mana/Stamina/Energy (for U/EP/P)

14: Chat log (box to right is scroll bar)

15: Location info - has things like name, recommended level range, faction/group that owns area, etc

16: Inventory (box to right is scroll bar) - click item for options

17: Targeted mob info. Hovering your mouse over a mob for 0.75 (?) seconds sets it as currently active mob (can be friendly or unfriendly). Has portrait, health, M/S/E, and name. This does not affect combat.

• Graphics Designer: You will be in charge of creating the images to be used for the different GUI elements. You will be asked to follow specific criteria to ensure compatibility with the game.