Group F

Project Plan

Stanislav Stefinyn, Omar Dominguez, Yann Probst

1. Document description

Title	Project Plan
Author	Stanislav Stefinyn, Omar Dominguez, Yann Probst
E-Mail	stanislav.stefinyn@student.csulb.edu omar.dominguez@student.csulb.edu yannPierre.probst@student.csulb.edu
Key Words	Project Plan
Date of update	02/28/2020

2. Revision History

Date	Version	Author	Section(s)	Description
02/25/2020	0.1	Yann Probst	All	Creating the document and setting up the template
02/27/2020	0.2	Yann Probst	3, 5, 11	Writing the sections
02/27/2020	0.3	Omar Dominguez	4, 9, 11, 12, 13	Writing the sections
02/27/2020	0.4	Stanislav Stefinyn	1, 2, 8, 7	Writing the sections
02/28/2020	1.0	Yann Probst	*	Design of the document

3. Table of Contents

4. Document Summary

This document is our Project Plan for our Illuminati game projects. This document serves as a roadmap that shows the project phases and all managerial aspects of our Illuminati project. The document will cover different aspects such as a brief description of the project goal and scope. The organization of the team and the project. The schedule, risk managements and quality insurance for the project. It is important to follow the steps to realize the project because it is essential for a good realization of the project.

5. Project Plan

1. Overview

E-luminati is an innovative and ambitious electronic version of the famous Illuminati board game. The board game was published in 1982 and became successful since then. However, in the last few years, the game had a hard time competing with other games and slowly lost the general public's interest. E-luminati has the objective to be the true successor of the Illuminati game and bring a new perspective to the game. This project is managed by a team of three developers using commonly available tools to achieve their objective in a 4 months' time span.

2. Goals and Scope

2.1 Project Goals

Project Goal	Priority	Comment/Description/Reference
Functional Goal		
• Compatibility	Low	The game should be compatible with Windows and Mac
• Multiplayer	High	The user should be able to play with other people on internet
• AI	High	The user should be able to play against Artificialyl Intelligent opponents
Business Goal		
• Website	Medium	A website should present the game and propose its download
• Time-to-market	High	The delivery dates should be respected
Interface Goal		
• Design	Medium	The game should recall the old board game Illuminati
• 3D interface	Low	The game should be fluid in a 3D interface
• Rules	Medium	The game should explicitly teach the player how to play

2.2 Project Scope

2.2.1 Included

The deliverables of this project and their receivers are listed in detail in the delivery plan in chapter 10.

2.2.2 Excluded

This project will exclude for the moment:
- a competitive system
- a customized/sandbox mode

Also, certain users will be excluded from playing the game if they don't meet certain hardware specifications. The followings table details the different hardware requirements the user has to meet in order to play the game

Spec	Minimum Spec	Recommended Spec	Ultra Spec
GPU	DirectX 11 graphics card with 1 GB Video RAM (e.g. NVidia GTX 460, AMD Radeon HD 5770)	DirectX 11 graphics card with 2 GB Video RAM (e.g. NVidia GTX 560 ti, AMD Radeon HD 7850)	Latest DirectX 11 graphics card with 2 GB Video RAM or more (e.g. Nvidia GTX 670, AMD Radeon HD 7970)
CPU	Quad core Intel or AMD (e.g. Intel Core2 Quad Q6600, AMD Phenom X4 9750)	Four core Intel or Eight core AMD (e.g. Intel Core i7-3770, AMD FX-8350)	Latest Six core Intel or Eight core AMD (e.g. Intel Core i7-3930K, AMD FX-9370)
RAM	4GB	8GB	8GB or more

2.2.3 Major Software Functions

Game will include the original rules and regulations of the famous Illuminati board game. Moreover, further futures will be developed, such as:

- Multiplayer mode - The game includes a multiplayer mode that allows the player to measure itself to other players. The user will be able to create and/or join a lobby from which, he will be able launch a game of Illuminati

- Solo mode - The user will be able to train its skills against an AI. The user will select the number of AI he wants to play against. The AI will use predetermined

- Statistics - The user will be able to see its game statistics, such as number of wins and some achievements like winning a game with a certain group. Those statistics will be stored on a separate database and retrieved from it through an API for more security.

- 3D Game - The game interface will be a 3D representation of the board game with a plain surface, cards, dices, megabucks. The interface will be created with Unity.

- Installation - The game will be installed through a launcher. This launcher is an easier method of publishing the game and making it available to a wider audience.

- Website - The user will be able to see game future developments on a website. The user will also be able to review the game and give suggestions on this webpage.

- Game Rules - The player will be able to read the game rules directly on the game interface and browse the cards.

3. Organization

From a technical point of view, we will organize ourselves around git, which is a distributed version-control system for tracking changes in source code during software development. The team will use gitKraken to track changes and stay on top of the latest version of the game. Our source code will be available in the github repository at https://github.com/probsty/343-GroupF-Illuminati. The team has already installed a common version of Unity to avoid any conflicts or unpleasant surprises.

From an organizational point of view, we have set up a messenger conversation, we also plan to see each other regularly to talk about the project and to keep a good understanding between us. In addition, we will use the github management tools method for the management of the project, which was set up at the following address: **https://github.com/users/probsty/projects/1**

3.1 Organization Boundaries and Interfaces

The project is a study project, our main "client" is the teacher, who will follow us, support us and advise us on possible questions.

3.1.1 Resources Owners

We don't have a resources owner for the development of our student project unity game.

3.1.2 Receivers

We don't have a receiver for the development of our student project unity game.

3.1.3 Sub-contractors

We don't have a subcontractor for the development of our unity game.

3.1.4 Suppliers

Company	Deliverable	Comment
Unity3D	Game Engine	Student version
MongoDB	Database	Free version
Amazon	Server	No discount

3.1.5 Cross Functions

Function	Responsibility/Comment
Quality/Test	Stanislav Stefinyn
Management	Yann Probst
Documentation	Stanislav Stefinyn
Technical	Omar Dominguez
Product Owner	Omar Dominguez
Design	Stanislav Stefinyn
Marketing	Yann Probst

3.1.6 Other projects

We don't have any other similar projects with which we could evaluate. We have no further experience. This section is therefore not applicable.

3.2 Project Organization

The project is organized around github. To facilitate the use of github, we will use gitKraken so that we can see the progress of the project and be able to visually and more simply use the git commands. As a working method, we have chosen to centralize our tools and thus to use github management. The working method chosen is the Kanban method.

3.2.1 Project Manager

Each member of the group has a similar hierarchical responsibility, we will choose our decisions by vote, the majority prevails (2 against 1). We have nevertheless decided to have party leaders who will follow and be responsible for the different deliverables of the project.

Name of the person in charge	Project part
In charge of the management of the team	Yann Probst
In charge of documents to be returned	Yann Probst
In charge of the web part	Yann Probst
In charge of the API	Yann Probst
In charge of the development of the Game in Unity	Omar Dominguez
In charge of the multiplayer, bot	Omar Dominguez
In charge of the design of the Game	Stanislav Stefinyn
In charge of the documentation	Omar Dominguez
In charge of the testing	Stanislav Stefinyn
In charge of the database and the server	Stanislav Stefinyn

3.2.2 Project-internal Functions

See table above (3.2.1 section)

3.2.3 Project Team

Name	Availability	Comment
Omar Dominguez	Everyday (10am to 9pm)	Student homework so no cost
Stanislav Stefinyn	Everyday (10am to 9pm)	Student homework so no cost
Yann Probst	Everyday (10am to 9pm)	Student homework so no cost

3.2.4 Timeline Chart

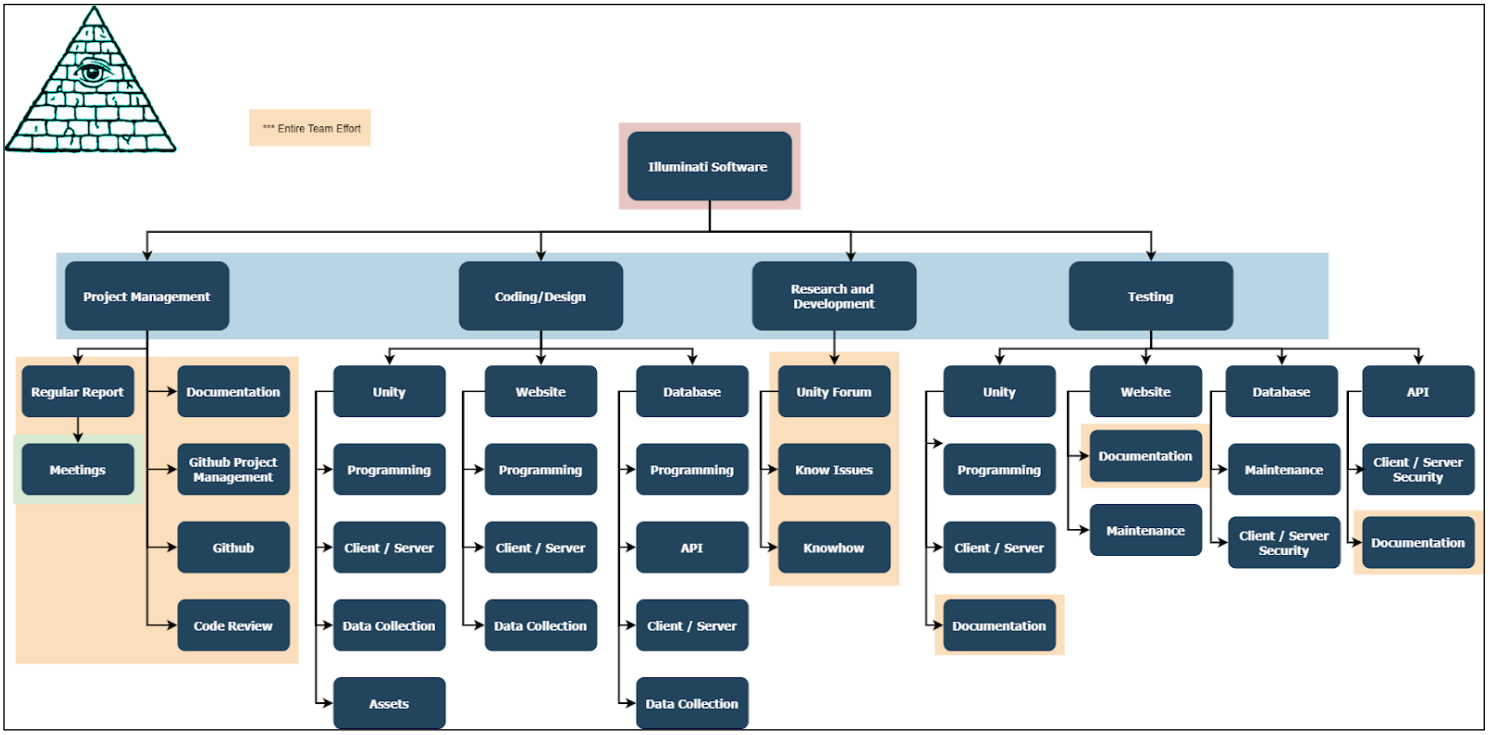
	Design	Game Architecture	Music/Sound	Multiplaying	Testing	Website	Bot	Game
6 march								
13 march								
20 march								
27 march								
3 april								
10 april								
17 april								
24 april								
1 may								
8 may								

3.2.5 Steering Committee

See section 3.2.1

4. Schedule and Budget

4.1 Work Breakdown Structure



Work Breakdown Structure 1

4.2 Schedule and Milestones

MileStone	Description	Requirements	Date
Design and Vision	Basic ideas and setting up our environment	• Setup Unity • Setup Flask webserver • Grasp understanding of game rules • Designs and plans made	03/06/2020
Working Prototype	Have a working prototype for the cards and	• Unity game must be runnable • Game should have simple features and a simple UI	03/27/2020
Playable Game	Have a game that is playable with the standard game with solo mode	• Should be able to play one game of Illuminati • Solo player feature working • Can play with AI's locally • Game should resemble board game	04/17/2020
Online Features	The game is playable and now has the ability to play with other players online. Extra feature may be added or in development at this stage	• Online feature working • Extra features working or in development	05/08/2020

4.3 Budget

Category	Milestone 1	Milestone 2	Milestone 3	Milestone 4
Unity License	$100	$ 0	$ 0	$ 0
Equipement	$50	$ 0	$ 0	$ 0
Other Software	$20	$ 0	$20	$ 0
WebServer	$20	$ 0	$20	$ 0
Design Assets	$50	$ 0	$50	$ 0
Total	$240	$ 0	$90	$ 0

4.4 Development Process

We will use a customized process model, guided by our Professor Giacalone. We are going to apply what Professor Giacalone tells us and follow all these advices to make the development of this project a success.

4.5 Development Environment

Language/Technology	Application	Detail	Version
Unity	Game engine	Unity is a game engine which will run the main game in the background	2018.4
ReactJS	Frontend Framework for the website		16.8.0
C#	Coding and Scripts for games		8.0
MongoDB	Database for the game client and website	Database for the game and users meant for an online experience; will be accessed through an API	4.0
Flask	Creating a website for the game		1.1.1

4.6 Measurements Program

We want to have short measurements for the entire project in order to keep track of our progress and to adjust as quickly as possible. Since we have a short project timeline it is important that we work quickly without having to sacrifice quality in the development cycle.

In order to do so we have proposed a couple of metrics to help us keep track of each other and help one another:

Metric	Details/Reason	Frequency / Metric used
Code Review	The most basic and fundamental way to understand how to know where the project is in terms of code and line numbers	Once every two weeks / Hours or Minutes
Meetings	Meeting where we can discuss that weeks issues and solutions	Once a week / Minutes
Github Project Management	Githubs project management system lets us track individual tasks and meet goals	Update daily / Tasks Completed
Documentation	Documentation can help us determine how complex the game or code has come to a certain point. This gives us a overhead view of the project for anyone to view	Once a week minimum / Should be updated from each member at their discretion

5. Risk Management

5.1 Project Risks

We have many risk issues that can happen during the project lifecycle. The major risks are:

- Basic software installations failure
- Technology will not need expectations
- Lack of knowledge of the group members in the technologies used
- Poor commenting of source code
- Less communication between the group
- Lack of interests from the stakeholders
- Poor project architecture
- Poor time management
- Misunderstanding the rules of the game
- Lack of resources (Design)

5.2 Risk Table

Risks	Category	Probability	Impact	Solution
Basic software installations failure	Technical	15%	1	Create a documentation to install similar versions between each member
Technology will not need expectations	Technical	5%	1	Used a graphical game engine (Unity)
Lack of knowledge of the group members in the technologies used	Internal	40%	2	Make a tutorial unity to take in hand
Poor commenting of source code	Technical	50%	1	Make a documentation of the classes to be used and a UML
Less communication between the group	Internal	15%	4	Creating facebook conversation, Do activities together, Work together

Lack of interests from the stackholder	Internal	15%	2	Buy the board game and play it together to have interest in playing it.
Poor project architecture	Technical	35%	2	Focusing our efforts together to find a good architecture. Creating a UML
Poor time management	Internal	30%	4	Use Tool Time Management to manage the time
Misunderstanding the rules of the game	Internal	30%	2	Read the rules, play the board game
Lack of resources (Design)	Technical	40%	1	Train on tools such as blenders and try to make a classic design.

6. Sub-contract Management

This project is entirely done by the team of developers in charge. There is no outsourced work.

7. Communication and Reporting

Type of communication	Method/Tool	Frequency/Schedule	Information	Participants/Responsabilities
Project Meetings	In person	Every Thursday and Tuesdat	Project status	Project Team and client (our professor)
Written application	Messenger Application	Everyday	To be used everytime	Project Team
Final Project Meeting	In person	Project Deadline	Wrap-up of the project	Project Team

8. Delivery Plan

8.1 Deliverables and Receivers

Indent	Deliverable	Planned Game	Receiver
D1	Webstite	03/22/2020	Users
D2	API	05/08/2020	Project team
D3	Database	05/08/2020	Project team
D4	Illuminati Game	05/08/2020	Users

9. Quality Assurance

The quality insurance code will be insured by the respect of several coding conventions.

Website : Unit test with Jest for ReactJS
API : Use of best practice for Rest API
Database: x
Illuminati Game : C# best practices and Unity Unit Testing
Git commit norm: AngularJS norm

10. Configuration and Change Management

Identification of configurable items	Labelling of configurable items	Protection of configurable items	Keeping of baseline and others informations	Configuration Verification and Audit	Defining Responsibilities
Servers	1	security of the servers managed by an external company (we do not own the servers)	N/A	Verification will take place every two weeks	Responsible in case of change: Yann Probst
API	2	Basic security for api	N/A	Verification will take place every two weeks	Responsible in case of change: Yann Probst
Database Structure	3	N/A	N/A	Verification will take place every two weeks	Responsible in case of change: Omar Dominguez
The Game	4	N/A	N/A	Verification will take place every two weeks	Responsible in case of change: Omar Dominguez

11. Security Aspects

Security Problem	Action Taken	Monitoring/Reporting System
Securing the database	Create an API with python to limit the access to the database	Give access token to the api
Game vulnerabilities and bugs	Use Unity to limit vulnerabilities	Bug reports and unity forums for finding known vulnerabilities

12. Abbreviations and Definitions

AI - "Artificial Intelligence" device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals[1]

Unity - Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Inc.'s Worldwide Developers Conference as a Mac OS X-exclusive game engine [2]

MongoDB - MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema. [3]

API - An application programming interface is an interface or communication protocol between different parts of a computer program intended to simplify the implementation and maintenance of software. An API may be for a web-based system, operating system, database system, computer hardware, or software library.[4]

DataBase - A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.[5]

13. References

Unity

- **Unity Forum**
- **Online and in-person Unity courses & training in 2D, 3D, AR, & VR development!**

MongoDB

- **The most popular database for modern apps**

Work Breakdown Structure :

- What is Work Breakdown Structure