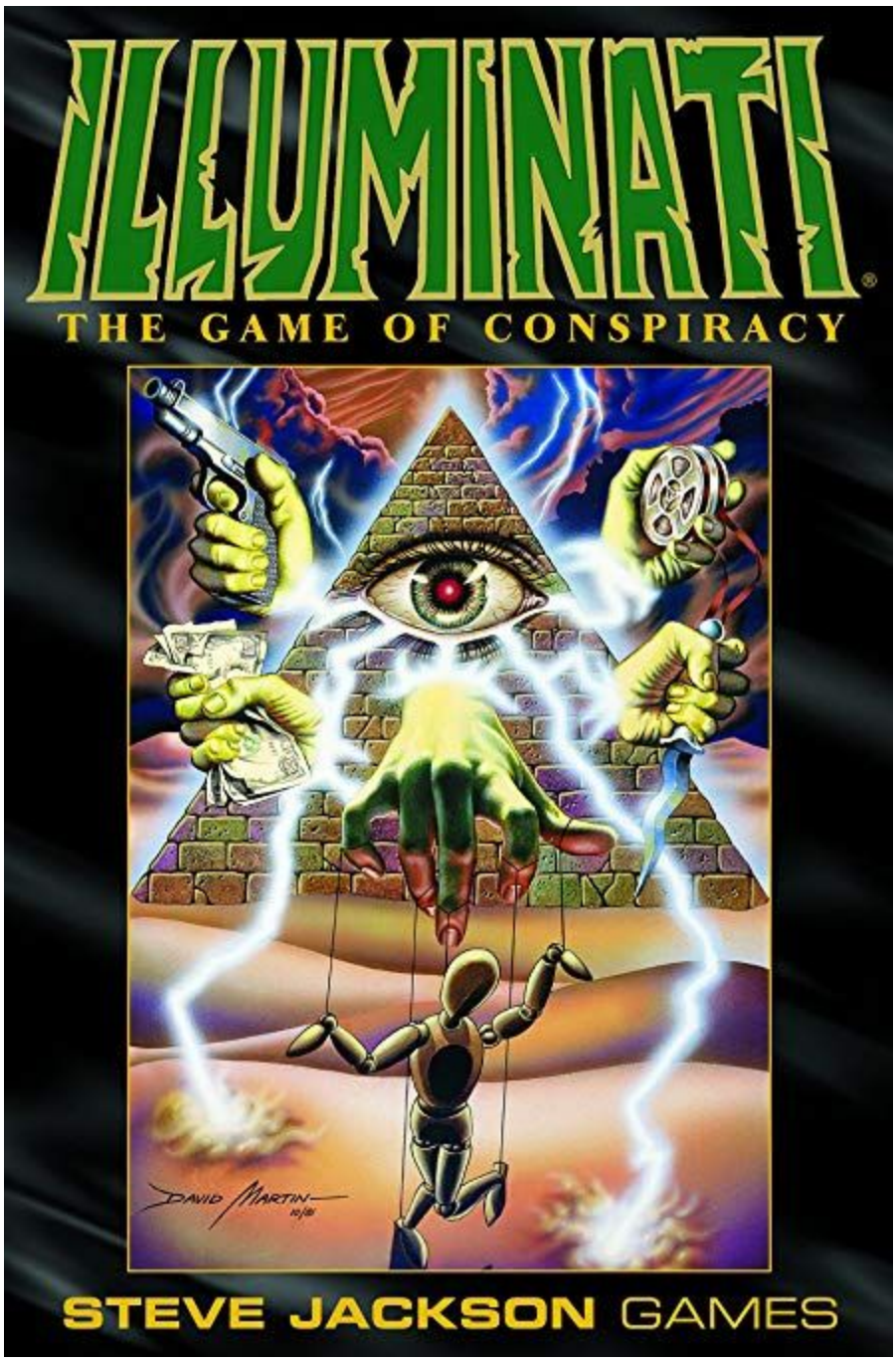
Project Plan

Illuminati



Project Development Team:

Thomas McSwain
Matthew Caponi
Adam Hinkle

1.Overview

"Illuminati" is a card game that was created by Steve Jackson Games in 1982. Our goal is to breathe new life into the game as a revamped PC strategy game. Our new game will be implemented as a cross between a card/board game and a grand strategy game.The original cards, some of the rules, and the spirit of the game will be kept intact. Yet, we will make some changes to the underlying gameplay to make it more suitable for a modern audience. Our market consists of PC owners with Internet access, ages 5+. The project will take approximately 2.5 months to complete.

2. Goals and Scope

2.1 Project Goals

Goal	Priority	Comments/Description
Functional Goals		
Graphical user interface.	high	GUI used by player to navigate through game.
Basic game asset graphics	high	Basic 3D objects for game assets such as cards, the board, the deck, etc.
Card redesign	medium	Redesign card images to be more aligned with new gameplay rules.
Detailed graphics design	low	Non-mandatory beautification of game asset graphics through manual art design and more extensive utilization of assets from asset store.
Music and sound design	medium	Design/selection of music and sound effects for the game.
Gameplay mechanics	high	Mechanics needed to implement game logic.
Multiplayer capabilities	medium	Ability to connect to server to

		play with friends across machines
AI implementation	medium	Design AI agents to play against in either singleplayer or multiplayer mode.
Pass n' play Mode	high	Ability to play among friends with 1 machine
Single Player Mode	medium	Play solely against AI opponents.
Multiplayer Mode	medium	Play against friends on other machines with or without AI opponents to fill in empty seats.
Business Goals		
Low maintenance	medium	Infrequent need for bug-fixes/updates
Easily accessible	high	Easy for user to begin playing
Quick design	high	Short development time
Technological Goals		
Unity research	High	Learn the Unity game engine in preparation for development.
Multiplayer research	high	Research networking implementation in support of multiplayer capabilities.
AI research	medium	Research AI design and algorithms in support of AI implementation.
Card photocopying and cropping	high	Photocopy all cards and use photo editing tools to crop them and isolate them into individual image files.
Organizational Goals	high	
Use Cases	high	Thoroughly detail all use

		cases and possible outcomes.
Program Flow Chart	high	Chart to illustrate and support the use cases.
UML diagrams and implementation specifics	high	UML diagrams for object relationships and specific details about implementation.
Test plan	high	Details on how to test program features
User documentation	high	Documentation to support users in operating the program.
Quality Goals		
Smooth UI	high	Fast and bug-less playing experience.
Entertaining experience	high	Fun and entertaining experience while playing game
Deep, strategic, and challenging gameplay.	medium	Gameplay that makes the user have to think and act strategically.
Constraints		
Release date	high	~2.5 months to complete
AI and networking	high	Difficulty in learning challenging new concepts
Experience	medium	Relatively new programmers
Team organization	medium	Lack of software engineering experience
Low budget	low	Low money investment

2.2 Project Scope

2.2.1 Included

Feature	Work Package[1] [2]	Description
Main Menu	1.3. 5.2., 5.5. 8.1.	Menu for the game with the standard menu options.
Pass n Play Mode	3.1 2.1-2.4 5.1.-5.4.	Gameplay mode where player plays with friends by passing the computer around with each passing turn.
Single Player Mode	3.2. 2.2.-2.4. 5.2.-5.4.	Gameplay mode where player plays against AI agents.
Multiplayer Mode	3.3. 2.3-2.4. 5.3.-5.4.	Gameplay mode where player plays against other real players through a server.
Game UI	1.3, 1.5 5.1.-5.5. 8.1-8.5.	User interface used to navigate various game options, tools, and interfaces.
Informational Tools	1.3., 1.5. 5.3., 5.4. 8.5	Various UI that displays information about traits, attributes, alignments, action history, and other information needed for the player to make strategic decisions.
Game Board	1.1, 1.5. 5.1., 5.5.	3D asset used as a playing surface for the game.
Location	1.2, 1.4. 5.1., 5.5.	3D environment that game board resides in.
Scene Change	2.1. 5.1.	Option to change to a different playing environment.

[1]See 4.1. For Work Breakdown Structure
[2]Features with multiple attributes means it is designed at a basic level in the first listed work package and then either expanded or refactored in the subsequent work package(s).

Cards	1.1, 1.4 5.1., 5.5.	3D card assets.
Turn System	2.1 3.1-3.3 5.1.-5.4.	System used to start and end turns.
Card Manipulation	2.1. 5.1.	Ability to draw a card, discard a card, move a card, magnify a card, and other card manipulation operations.
Money System	2.1-2.4 5.1-5.4.	A system used to regulate, control, and monitor income distribution, money transfers, and any type of money action.
Major Action System	2.2.-2.4. 5.2.-5.4.	A core gameplay feature that is comprised of different major actions used to achieve high-level game goals, such as attack to control, attack to neutralize, and attack to destroy.
Sub-Action System	2.2.-2.4. 5.2.-5.4.	A core gameplay feature that is comprised of specific actions used to help accomplish higher-level goals that can be taken upon by groups and individual leadership members.
Special-Power Action System	2.2.-2.4. 5.2.-5.4.	A core gameplay feature that allows for special-powers to be carried out as defined by the cards.
Diplomacy System	2.2.-2.4. 5.2.-5.4.	A core gameplay system that mediates diplomatic interactions between groups.
Emissary Tool	1.3., 1.5. 5.3., 5.4. 8.3.	A tool that facilitates communication between the player and the AI in diplomatic interactions.

Whisper Tool	1.3., 1.5. 5.3., 5.4. 8.3.	A tool that allows players in multiplayer game mode to chat with each other privately.
Notification Tool	1.3., 1.5. 5.3.-5.4. 8.2.	A tool that provides the player with general game notifications.
Event Notification Tool	1.3., 1.5., 5.3., 5.4. 8.2.	A tool that informs players of any events or actions taken by opponents, both AI and multiplayer.
Leadership System	2.3.-2.4. 5.3.-5.4.	A gameplay system by which groups are comprised of individual leadership members that can take actions and have actions taken upon against them.
Trait, Attribute, and Alignment System	2.3.-2.4. 5.3.-5.4.	Internal system that tracks and updates the complex interaction of traits, attributes, and alignments of groups based on their actions and interactions.
Probability System	2.4. 5.4.	Internal system that determines the likelihood of a specific action working and the extent of its damage, referencing a various interplay of factors, including the traits of involved parties.
Group Dynamics	2.3.-2.4. 5.3.-5.4.	The classification of groups into group types and the defining of traits, attributes, and alignments, as well as sub-actions that can be taken by each group type, sub-group type, and group itself.
Power Hierarchy	2.3.-2.4. 5.3.-5.4.	An internal representation of the different power tiers each group type resides on and their relationships between

		other groups.
Power Structure	2.1.-2.4. 5.1.-5.4.	A core gameplay feature that allows players to take control of groups and arrange them in such a way as to grow the influence, control, and scope of their Illuminati group.
Elimination System	2.1.-2.4. 5.1.-5.4.	A system used to track criteria for player elimination and to act accordingly.
Goals	2.4. 5.4.	An internal defining of basic game goals, as well as classification of special goals per group type.
Win System	2.1.-2.4. 5.1.-5.4.	A system used to track progress toward meeting a win condition.
Music	4.1, 4.3 5.5.	Both background music and music fitted for specific game events.
Sound Effects	4.2, 4.3 5.5.	Sound effects used to make the game more dynamic.

2.2.2 Excluded

Feature	**Reason**
Anti Cheat	It is outside the scope of the project to expect players to cheat in this kind of environment
Modding Capabilities	This game is specifically designed for a certain experience and modding might give players an inferior experience.

3. Organization

3.1 Organizational Boundaries and Interfaces

3.1.1 Resource Owners

Not applicable

3.1.2 Receivers

Receivers are defined in the Delivery Plan in section 8.

3.1.3 Sub-contractors

None

3.1.4 Suppliers

Company	**Deliverable**	**Comment**
Unity	Unity Game Engine	N/A
Steve Jackson Games	Illuminati game	N/A
DigitalOcean	Server	N/A

3.1.5 Cross Functions

Not Applicable

3.1.6 Other Projects

Not Applicable

3.2 Project Organization

3.2.1 Project Manager

Not Applicable

3.2.2 Project-internal Functions

Function	Team Member(s)	Comment
Project Vision	Matthew Caponi Thomas McSwain Adam Hinkle	Outline project trajectory
Creative Direction	Matthew Caponi	Propose goals for project
AI Research	Matthew Caponi	Research methods to implement game AI.
Network Proposals	Thomas McSwain	Pursue online server
Documentation	Matthew Caponi Thomas McSwain Adam Hinkle	Document planning and implementation
Code Generation	Matthew Caponi Thomas McSwain Adam Hinkle	Implement functionality of game
Project Delivery	Matthew Caponi Thomas McSwain Adam Hinkle	Present project

3.2.3 Project Team

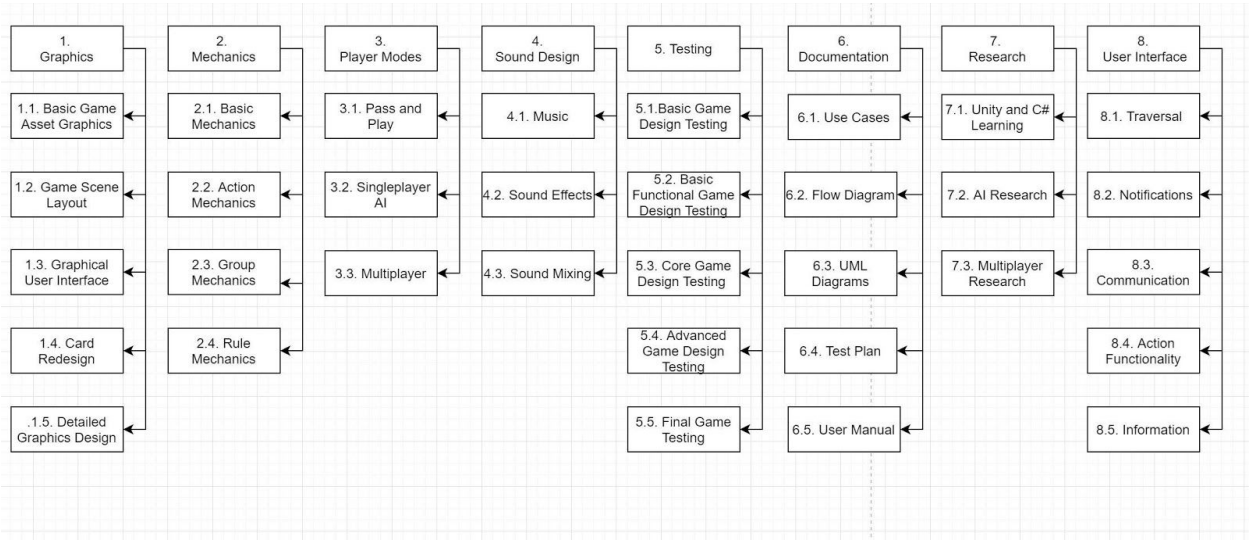
Team Member	Availability	Comment
Matthew Caponi	Full duration	N/A
Thomas McSwain	Full duration	N/A
Adam Hinkle	Full duration	N/A

3.2.4 Steering Committee

Not Applicable

4. Schedule and Budget

4.1 Work Breakdown Structure



4.2 Schedule and Milestones

Milestones	**Description**	**Milestone Criteria (Numbers refer to Work Structure)**	**Planned Delivery Date**
M0	Design Phase 1	Vision Document, Project Plan	2/25/2020
	Created vision document and project plan.	Turned in.	
M1	Design Phase 2	6.1	3/13/2020
	Design use cases.	Use cases successfully completed detailing every possible action the user can take along with every possible consequence.	

M2	Design Phase 3	6.2,6.3, 7.1	3/27/2020
	Design flow diagram and UML diagrams. Learn Unity and C#.	Flow diagram illustrates the flow of gameplay in a succinct and clear manner. UML diagram illustrates the relationships between classes in such a way that code can be implemented with ease. Gain solid understanding of Unity and C# as it relates to game programming,	
M3	Research Phase	6.4, 7.2, 7.3	4/3/2020
	Research Game AI methods and multiplayer networking functionality. Design test plan.	Gain solid understanding of applicable game AI methods and a way to implement them. Gain solid understanding of implementing multiplayer functionality. Test plan completed.	
M4	Basic Game Design	1.1-1.3, 2.1, 3.1, 5.1. 8.1	4/3/2020
	Implement basic game graphics, Pass n Play functionality, user interface functionality, and gameplay mechanics.	Complete Basic Game Design Testing.	
M6	Basic Functional Game Implementation	2.2, 3.1, 3.2., 5.2.	4/10/2020
	Implement basic functionality for core game functions, refactor Pass n Play, and implement basic game AI.	Complete Basic Functional Game Design Testing	

M7	Core Game Implementation	2.3, 3.1.-3.3., 5.3., 8.2.-8.5.	4/17/2020
	Implement group mechanics, refactor Pass n Play and Single Player AI, begin implementation of UI functionality, and start implementation of Multiplayer.	Complete Core Game Design Testing	
M8	Complete Game Functionality Implementation	2.4., 3.1.-3.3., 5.4., 6.5., 8.1-8.4	4/24/2020
	Implement rule mechanics and refactor all gameplay modes and UI tools. Write a user manual.	Complete Advanced Game Design Testing. User manual completed, thoroughly documenting the game for the user.	
M9	Game Polish	1.4., 1.5., 4.1., 4.2., 4.3., 5.5.	5/1/2020
	Redesign cards, improve game aesthetics, and mix in music and sound effects.	Complete Final Game Testing	

4.3 Budget

No budget besides <$50 of fees related to unity and server deployment.

4.4 Development Process

Following the guidance of our Professor Anthony Giacalone our development process is happening in the form of first creating the overall vision document, then creating the project outline, next we create a use cases document, followed by UML diagram, finally we start implementing our design.

4.5 Development Environment

Item	Applied For	Available by

Methods		
See 4.4		
Tools		
Unity	Game Engine	
Languages		
C#	Game Structure/Mechanics	

5. Risk Management

(1 being greatest risk)

Risk	**Probability**	**Impact**
Late Delivery	1%	1
Buggy Code	10%	2
Lackluster AI	20%	3
Dysfunctional Multiplayer	40%	4

We will be constantly reevaluating specifically multiplayer and to a lesser extent AI to make sure we aren't wasting time on something that isn't going to work. There is always the possibility that improper class design could lead to buggy code that is not completely fixed before delivery, but we will do our best to mitigate that by doing due diligence to design.

6. Sub-Contract Management

Not Applicable

7. Communication and Reporting

Type of	Method/Tool	Frequency/	Information	Participants/

Communication		Scheduling		Responsibilities
Internal Communications				
Project Meetings	In Person	Bi-weekly	Update progress, short term goals	Project Team
Sharing Code	Github	Daily	In-game files	Project Team
Design Collaboration	In Person	Weekly	Problems, requirements, implementation	Project Team
Documentation	Google Docs	Pre-Milestones	Project development documentation	Project Team
External Communications				
Required Documentation	Github	Milestones	Project Development	Project Team, Project Advisor

8. Delivery Plan

8.1 Deliverables and Receivables

ID	Deliverable	Planned Date	Receiver
D1	Vision Document	2/14	Project Advisor
D2	Project Plan	2/28	Project Advisor
D3	Use Cases	3/13	Project Advisor
D4	Project Flow Chart/UML	3/27	Project Advisor
D5	Test Plan	4/3	Project Advisor
D6	User Manual	4/24	Project Advisor

D7	Finished Game	5/1	Project Advisor

9. Quality Assurance

One of the main goals of the projects is quality and as such we will be conducting extensive tests at every milestone of the implementation phase of the project in order to ensure our program is of the highest quality.

10. Configuration and Change Management

The main configuration for the workspace of the developers will be through a centrally hosted code repository through git that syncs the different builds of the game in unity projects to each developer.

Our plan for being agile in response to change is to always adhere to strict engineering principles throughout the process of developing and strive for high cohesion low coupling modules that are easy to drag and drop into various aspects of the project without much change to ensure that we can easily make whatever changes necessary with the least amount of headache chasing down side effects.

11. Security Aspects

The only main security concerns are related to game integrity since we have no plans for server side data storage of our users so we will not have to worry about potentially losing player data. The main problem comes in the form of using no validation to ensure whatever the client is saying is correct as well as making sure that no programs are modifying internal registers.

12. Abbreviations and Definitions

None

13. References

1. Giacalone, Anthony. *CECS 343 - Project Documentation*. Cal State University, Long Beach.
2. Giacalone, Anthony. *CECS 343 Spring 2020 Syllabus*. Cal State University, Long Beach.

3. Jackson, Steve. “Illuminati Card Game Rules.” *SJ Games*, Steve Jackson Games, www.sjgames.com/illuminati/img/illuminati_rules.pdf.
4. Meyer, B. “Template Project Plan.” *Successful Software Outsourcing and Offshoring*, ETH.

14. Revision

Date	Version	Description	Author
2/12/20	<1.0>	Created skeleton of document.	Adam Hinkle
2/20/20	<1.1>	Started project goals	Adam Hinkle
2/20/20	<1.2>	Expanded project goals	Matthew Caponi.
2/25/20	<1.3>	Work Breakdown structures, milestones, schedule, Organization, Project Scope, Communication reporting, risk management, Quality Assurance	Adam Hinkle, Thomas McSwain, Matthew Caponi.
2/28/20	<1.4>	Finished document	Adam Hinkle, Thomas McSwain, Matthew Caponi.