Project Proposal Document

Mach G

Disc Ex Machina

Alvin Cheung, Cyrus Duong, Grant Guillen, Henry Nguyen, Michael Wang
Team 2
Information

**Team Name:** Team Mach G

**Team Members:**
- Alvin Cheung
- Cyrus Duong
- Grant Guillen
- Henry Nguyen
- Michael Wang

**Email(s):** [TeamMachG@gmail.com](mailto:TeamMachG@gmail.com) (All emails fall under this)

**Github link:** [https://github.com/cnhenry/machg](https://github.com/cnhenry/machg)

**Meeting Time:** 10/7, 11/18 @ 11:15

**Team Meeting Times:** Tue, Thurs - after class; Varying weekends

**Lab Meeting Time:** Mondays @ 4:00 PM

**Project Sponsor:** None

**Project Game Name:** Disc Ex Machina
Project Description

Virtual Reality is current technology that has yet to be pushed to its full potential, and is a currently growing market. Disc Ex Machina would allow us to design an game that works in tandem with the latest VR hardware and controllers. The primary goal is to create a game in an immersive virtual world to learn how to setup and maintain a workflow to develop video game software with industry-level quality standards in mind.

Disc Ex Machina is developed in Unity, as it has many libraries that can be used to save time from creating certain aspects of games. These include physics, game logic and components, and a fluid user interface with user experience as a high priority.

The goal is to design a game in which the player is able to throw a disc like object which would bounce around the arena to hit a target or opponent. The ideal gameplay involves player versus player combat, as the two players attempt to eliminate each other with the discs. Being in VR provides an environment with quality visuals and an immersive gameplay experience with accurate physics and motion.

The finished product would be a game that could be downloaded from a distribution platform such as Steam and would ideally feature a multiplayer mode in which players may compete in real time with one another online. We are aiming for an industry level of quality with Disc Ex Machina to produce a fully fleshed out product with all the features and functionalities that one would expect from a polished game.
Project Milestones

Implementation Milestones¹

1. Virtual world created with basic dimensions and player camera movement.
   a. 1 Person - 1 Day - September
2. Plan a design for all aspects of game
   a. 5 People - 3 Days - September
3. Player movement and objects capable of being thrown.
   a. 2 People - 3 Days - September
4. Player object rebounding with physics along with addition of targets/AI enemy added into the game. Refined collisions and hit detection.
   a. 3 People - 7 Days - October
5. Training target practice mode
   a. 2 People - 2 Days - October
6. Basic UI Elements with menu navigation
   a. 2 People - 2 Days - October
7. Base models with some thematic appeal
   a. 3 People - 15 Days - November
8. Fully develop game logic (scoring, win conditions, options modifications)
   a. 3 People - 7 days - November
9. Collision Noises and Sound Design
   a. 3 People - 15 Days - November
10. Introduction of 2 player mode with addition of second machine.
    a. 3 People - 7 days - December
11. Main Theme Song, Background Music, Ambient Music
    a. 3 People - 5 Days - March
12. Completed Models and Textures
    a. 2 People - 4 Days - March
13. Finalize aesthetic and theme of the game
    a. 5 People - 2 Days - April

¹ Days do not directly translate to hours. One day could vary between 4-6 hours accordingly to school hours and school workload.
Documentation Milestones

1. Team Formation - September
2. Initial Project Description - September
3. Gantt Chart - October
4. Quad Chart - October
5. Project Proposal Report - October
6. Project Proposal Video - October
7. Design Documentation - November
8. Requirements Documentation - December

Mach G will be collaborating as a full team, and each document will require all five individuals to actively participate and contribute to the creation of the documentation. Each member is expected to work on and develop clear, concise documentation that accurately reflects our product and its many aspects. Each member is also required to attend weekly meetings and is expected to work on development as well as documentation in their free time. As a team, we are accountable for each member to achieve a certain standard of quality as well as making sure each document is submitted punctually.
Project Budget

Hardware
- Hardware is all owned by us
- 2 HTC Vives
- Desktops: i7 Processors; Nvidia GTX 1070/980 Ti’s; 16GB RAM
- Play Areas: Living room of team apartment and additional room.

Software
- Game Engine
  - Unity
    ■ https://unity3d.com/
- Modeling
  - Blender
    ■ https://www.blender.org/
- Texture/Material Editing
  - Substance Painter
    ■ https://www.allegorithmic.com/products/substance-painter
  - Photoshop
- Video Editing
  - Sony Vegas
    ■ http://www.vegascreativesoftware.com/us/
- Version Control
  - Github
    ■ https://github.com/
- Task Management
  - Trello
    ■ https://trello.com/
- Audio and Audio Editing
  - FL Studio Producer
    ■ https://support.image-line.com/jshop/shop.php
  - NoCopyrightSounds
    ■ https://www.youtube.com/user/NoCopyrightSounds
Work Plan

Team Member Proficiencies

While we have dedicated roles that each member is proficient at, we will be following more of a fluid design in terms of team roles. With this, each member can switch roles at any time and fill in for development.

Listed skills in order of proficiency:
**Alvin Cheung** - Design, Scripting, Documentation
**Cyrus Duong** - Video, Design, Scripting
**Grant Guillen** - Design, Scripting, Documentation
**Henry Nguyen** - Design, Scripting, Documentation
**Michael Wang** - Scripting, Documentation, Design

Member Fulfillment of Proficiencies

**Scripting**
- Alvin, Michael, Henry, Grant, Cyrus
- These scripts determine the inner workings and fine tuning elements of the game.

**Physics**
- Cyrus, Henry, Michael
- Sets the regulations of how objects move in correspondence to certain variables.

**Game Logic**
- Cyrus, Michael
- Determines the overall correctness and fairness of game decisions in the engine.

**Design**
- Alvin, Cyrus, Grant, Henry, Michael
- Thematic decisions in terms of UI, UX, Modeling, and Sound Design.

**UI Design**
- Alvin, Grant, Henry
- Design an efficient UI that is simple to use.

**UX Design**
- Alvin, Grant, Henry
- Design a user experience that is comfortable, efficient, and natural for the player.
Graphics
- Alvin, Cyrus, Grant, Henry
- Creation of images, models, and special effects that enhances the user's experience.

Sound/Audio
- Alvin, Cyrus
- Create sounds/music that adhere to our thematic decisions.

Video
- Alvin, Cyrus
- Trailer to promote the launch of Disc Ex Machina.
<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start Github Repo</td>
<td>1d</td>
<td>09/19/2016</td>
<td>09/19/2016</td>
</tr>
<tr>
<td>2</td>
<td>Set Up Views</td>
<td>2d</td>
<td>09/19/2016</td>
<td>09/20/2016</td>
</tr>
<tr>
<td>3</td>
<td>Debug World with Basic Dimensions</td>
<td>2d</td>
<td>09/20/2016</td>
<td>09/21/2016</td>
</tr>
<tr>
<td>4</td>
<td>Player Camera Movement</td>
<td>2d</td>
<td>09/20/2016</td>
<td>09/21/2016</td>
</tr>
<tr>
<td>6</td>
<td>Plan a Design for All Aspects of Game (Theme)</td>
<td>5d</td>
<td>09/20/2016</td>
<td>09/26/2016</td>
</tr>
<tr>
<td>8</td>
<td>Initial Disc Model</td>
<td>2d</td>
<td>09/27/2016</td>
<td>09/28/2016</td>
</tr>
<tr>
<td>9</td>
<td>Initial Disc Texturing</td>
<td>2d</td>
<td>09/27/2016</td>
<td>09/28/2016</td>
</tr>
<tr>
<td>10</td>
<td>Initial Disc Physics</td>
<td>7d</td>
<td>09/28/2016</td>
<td>10/06/2016</td>
</tr>
<tr>
<td>11</td>
<td>Basic UI Elements</td>
<td>7d</td>
<td>10/03/2016</td>
<td>10/11/2016</td>
</tr>
<tr>
<td>12</td>
<td>Menu Navigation</td>
<td>7d</td>
<td>10/04/2016</td>
<td>10/12/2016</td>
</tr>
<tr>
<td>13</td>
<td>Basic Targets</td>
<td>3d</td>
<td>10/07/2016</td>
<td>10/11/2016</td>
</tr>
<tr>
<td>14</td>
<td>Target Destruction Shatter</td>
<td>3d</td>
<td>10/10/2016</td>
<td>10/12/2016</td>
</tr>
<tr>
<td>15</td>
<td>Training Target Practice Mode</td>
<td>7d</td>
<td>10/12/2016</td>
<td>10/20/2016</td>
</tr>
<tr>
<td>16</td>
<td>Refined Collisions and Hit Detection</td>
<td>5d</td>
<td>10/17/2016</td>
<td>10/21/2016</td>
</tr>
<tr>
<td>17</td>
<td>Initial Base Models and Textures (Players)</td>
<td>10d</td>
<td>10/17/2016</td>
<td>10/28/2016</td>
</tr>
<tr>
<td>18</td>
<td>Design Player Movement Mechanics</td>
<td>4d</td>
<td>10/21/2016</td>
<td>10/26/2016</td>
</tr>
<tr>
<td>19</td>
<td>Initial Environment Models</td>
<td>20d</td>
<td>10/21/2016</td>
<td>11/17/2016</td>
</tr>
<tr>
<td>20</td>
<td>Implement Player Movement</td>
<td>7d</td>
<td>10/25/2016</td>
<td>11/02/2016</td>
</tr>
<tr>
<td>21</td>
<td>Advanced Disc Physics (Rotation/Curve)</td>
<td>14d</td>
<td>11/02/2016</td>
<td>11/21/2016</td>
</tr>
<tr>
<td>22</td>
<td>Collision Noises</td>
<td>5d</td>
<td>11/04/2016</td>
<td>11/10/2016</td>
</tr>
<tr>
<td>23</td>
<td>Initial Environment Textures</td>
<td>12d</td>
<td>11/07/2016</td>
<td>11/22/2016</td>
</tr>
<tr>
<td>24</td>
<td>Refined Base Models and Textures (Players)</td>
<td>23d</td>
<td>11/21/2016</td>
<td>12/21/2016</td>
</tr>
<tr>
<td>26</td>
<td>Refined Environment Textures</td>
<td>23d</td>
<td>11/21/2016</td>
<td>12/21/2016</td>
</tr>
<tr>
<td>28</td>
<td>Multiplayer</td>
<td>24d</td>
<td>12/01/2016</td>
<td>01/03/2017</td>
</tr>
<tr>
<td>29</td>
<td>Finalize Base Models and Textures (Players)</td>
<td>12d</td>
<td>12/15/2016</td>
<td>12/30/2016</td>
</tr>
<tr>
<td>30</td>
<td>Main Theme Song</td>
<td>12d</td>
<td>12/15/2016</td>
<td>12/30/2016</td>
</tr>
<tr>
<td>31</td>
<td>Background Music</td>
<td>24d</td>
<td>12/26/2016</td>
<td>01/26/2017</td>
</tr>
<tr>
<td>32</td>
<td>Augmented Reality Visor</td>
<td>24d</td>
<td>01/10/2017</td>
<td>02/10/2017</td>
</tr>
<tr>
<td>33</td>
<td>Initial Environment Models #2</td>
<td>15d</td>
<td>01/16/2017</td>
<td>02/03/2017</td>
</tr>
<tr>
<td>34</td>
<td>Refined Environment Models #2</td>
<td>15d</td>
<td>01/16/2017</td>
<td>02/03/2017</td>
</tr>
<tr>
<td>35</td>
<td>Scoring System/Won Conditions</td>
<td>10d</td>
<td>01/26/2017</td>
<td>02/02/2017</td>
</tr>
<tr>
<td>36</td>
<td>Initial Options Menu</td>
<td>10d</td>
<td>01/23/2017</td>
<td>02/03/2017</td>
</tr>
<tr>
<td>37</td>
<td>Ambient Music</td>
<td>12d</td>
<td>01/30/2017</td>
<td>02/14/2017</td>
</tr>
<tr>
<td>38</td>
<td>Finalize Environment Models</td>
<td>30d</td>
<td>02/01/2017</td>
<td>03/14/2017</td>
</tr>
<tr>
<td>39</td>
<td>Finalize Environment Textures</td>
<td>30d</td>
<td>02/01/2017</td>
<td>03/14/2017</td>
</tr>
<tr>
<td>40</td>
<td>Other Training Modes</td>
<td>24d</td>
<td>02/06/2017</td>
<td>03/08/2017</td>
</tr>
<tr>
<td>41</td>
<td>Other Maps</td>
<td>24d</td>
<td>02/06/2017</td>
<td>03/08/2017</td>
</tr>
<tr>
<td>42</td>
<td>Advanced UI</td>
<td>30d</td>
<td>02/16/2017</td>
<td>03/23/2017</td>
</tr>
<tr>
<td>43</td>
<td>Finalize aesthetic and theme of the game</td>
<td>30d</td>
<td>02/16/2017</td>
<td>03/23/2017</td>
</tr>
<tr>
<td>44</td>
<td>More Audio Cues</td>
<td>100</td>
<td>02/15/2017</td>
<td>02/28/2017</td>
</tr>
<tr>
<td>45</td>
<td>More Background Music</td>
<td>30d</td>
<td>02/26/2017</td>
<td>03/31/2017</td>
</tr>
<tr>
<td>46</td>
<td>More Thematic Music</td>
<td>30d</td>
<td>02/26/2017</td>
<td>03/31/2017</td>
</tr>
</tbody>
</table>
Preliminary Project Design

This project is split up into the broad topics of scripting, design, sound, graphics that comprise a game. These topics are separately developed and are combined in the Unity engine in order to produce our finalized product.

Scripting

Unity engine provides the ability to manipulate the scene of game assets using C# scripts. This project will take advantage of this scripting engine in order to specify the physics, game entity component behaviors, and game logic.

Physics

1. Disc Mechanics (bouncing/trajectory/etc) - October 2016
   a. We will describe the disc behavior using scripts to manipulate Unity physics engine. These components will describe attributes such as bouncing, trajectory, and gameplay behaviors.

2. Player Movement - October or November 2016
   a. We script Unity’s physics engine to implement player movement within our virtual world. This will include movement within the player arena and teleportation around the arena. The components will include the player hitbox and model as well as the disc the player is holding.

3. Overall Game Environment Physics - October or November 2016
   a. We will implement scripts to manipulate Unity physics engine to implement environmental physics so that the player arena responds accurately to player and disc interaction. This includes interactions like the disc bouncing off the wall and the player pressing a button on a wall to trigger an action.

4. Object Interaction - October - April
   a. We must create a basis for each of the objects, and different scripts are written for each type of object. For example, some objects spawn other objects, some objects are destroyed when they are contacted, some objects bounce when they are interacted with, some objects trigger events, etc. Refer to Class Diagrams I and III for examples of object classes.
Game Entity Components

1. Hit Detection - November 2016
   a. We will use mesh colliders to detect collisions between two objects. These objects have scripts attached to them and we will implement certain scripts that trigger off of collision events. The possible collision outcomes are quite numerous, but the main ones that will be utilize will be deflection/rebound off of a collision along with impact and destruction after a collision. Every object such as the player, controller(s), disc, arena, and targets will have mesh colliders along with a series of scripts. With unity there are libraries which have built in functions that detect collision. We will run scripts that trigger once a collision is detected and utilize a series of conditional statements to determine the proper outcome. For example if a disc collides with another disc the two objects should reflect and bounce off of each other similar to what happens in reality. The scripting method for hit detection is adaptable and can be modified to suit certain situations.

   a. We will use the Unity VR libraries to write scripts to implement disc manipulation with the HTC Vive controllers. This will allow for accurate grabbing, holding, and releasing of the disc from the player. The orientation and position of the disc relative to the Vive controller is the most important aspect of making this manipulation and interaction accurate and natural for the player.
Game Logic

   a. For the game logic regarding the scoring engine, we have made some preliminary decisions determining how the scoring will be handled in DXM. We have proposed a health based system for each player which is visible in an augmented HUD that will appear on the corners of the player’s vision. We’ll signal to the UI to flash a certain color when getting hit, and will produce particle effects whenever the opponent gets hit. We’ll likely proceed with a round based system in which players would play a “Best of 5” or “Best of 3”, attempting to eliminate one another.

2. Manipulating the play area - October 2016
   a. For the initial play area, we would start with a cube/rectangular prism area to act as the basis for our arena. From here, we are considering implementing a dynamic mapping in which we could press buttons within the arena by throwing a disc or physically pressing a button close to the play area and change the map by adding barriers and moving platforms - altering the arena to give an playable advantage.

   a. Multiplayer would initially be created in a LAN format - if it ends up being easier. There will be a “Network Manager” that handles all the networked states of the game. From here, we can operate client hosted games where the host is also a player; with this, we can sync up two copies of our game to play with one another. By serializing the data and sending the data to another machine, we can give 1:1 feedback with seamless integration. Multiplayer would be a huge step for the game as the core functionality surrounds having a combative experience with other players. If possible, we could implement a matchmaking service based on skill ratings.
Design

UI Design

   a. We will create a user-friendly interface that is intuitive and easy to navigate. It will be fluid and simple, and provide all necessary options. These include gameplay, settings options, and exiting the game itself. They will be organized in a simple menu format of buttons which are selected with a laser emitting from the Vive controller. Clicking the button will take the player to the selected menu. A fluid menu system is a very important aspect of DXM, as it must feel natural and easy to navigate to the player to ensure a streamlined experience.

2. Diegetic Context UI - November 2016
   a. One of the goals in DXM is to implement Diegetic UI as it provides a more natural and immersive environment for a player. By attaching UI elements to objects within the game, we can create this virtualized environment that we were envisioning as we were creating the theme. One of our goals is to create a sort of Augmented Visor that we can pull in front of the User’s face that shows a futuristic user interface that one would expect out of a Sci-Fi movie. The major benefits of going with Diegetic UI is that it provides UI elements wherever we want, as they are attached to game objects.
Graphics

Most 3D models will be developed using an open-source modelling tool known as Blender. Models developed in blender will be assigned different priorities to describe the level of details. The reasoning behind this decision is related to the amount of relativity each model has to the player. For example a player will be observing an asset representing the player more often than a distant environment game asset, thus the logical action would be to consider the player representation as a higher priority.

1. Theme
   a. The graphics will be defined in a common cyberpunk and/or science fiction theme similar to movies such as Blade Runner, Tron, or Ghost in the Shell. Using these references in mind we will observe other similarly styled media through the internet in order to inspire our art asset designs.

2. Player Models
   a. The player's controllers will be represented by a set of hand models which will be animated and placed on a high level of priority for detail. The player will also be represented by a player model.

3. Disc Models
   a. The disc models are a main asset related to the play of the game. These disc models will also be placed on a high priority level of detail and shaped to be consistent to our scripted behaviors of a disc.

4. Game Environment
   a. We will define several game environments based on our references. Using these environments in mind, we will develop a list of assets to model using Blender. It is important to develop these assets with the intention of consistency as they will be modularly placed in a game environment, thus all game environment assets will be analyzed, and improved throughout development to support our designated art direction and theme. Game environment assets will have varying levels of detail based on the possible time of exposure to the player. For example a distant game model which the player is barely exposed to will be less detailed than an up-close and centric part of the game environment.
Sound/Audio

1. Background Music - January 2017
   a. We will use various soundtracks that fit the theme of the game and use a script that will allow various sound/song files to play in the background. For the menu background music there will be a timer/counter that will allow music to fade in and fade out as a transition between tracks. For the in-game background music one track will be played for the majority of this scene and will loop to create a seamless continuation as long as the game is still going.

2. Item Interaction Noises - December 2016
   a. For the in-game/battle mode we will add to the hit detection scripts so that when certain interactions happen different sounds will be played. As an example grabbing a disc would trigger a script that would play a sound that would indicate that the disc is being powered up and give the queue that the disc is ready to be thrown. For the menu we will utilize the same method to add sounds to the interactions in the menu.

3. Environmental Sounds - January 2017
   a. We will design and implement environmental sounds for the Main Menu scene as well as the player arenas. This will complement the game’s theme and provide further immersion for the player experience. Having appropriate environmental audio is important to supporting the game’s theme and making the environment make sense and feel natural to the player.

   a. A major component of sound that leaves a memorable experience resolves to collision noises. When a disc strikes a wall, it should leave an impression of the power capabilities. It should sound powerful and dangerous to the victim. But not just the disc, when we collide objects of different materials, we should be able to hear noises that we expect, hitting a hammer against a wall and hitting a hammer against a block of jello produce very different sounds. By providing the user with proper sound recognition, such as a blade slicing through an object, we give a stronger impression in our gameplay.
### Overview

**Virtual Reality System:** Disc Ex Machina is specifically developed for the HTC Vive virtual reality headset due to its technical specifications and handset integration.

**Game Engine:** Development is conducted in the Unity engine, which acts as the core of our gameplay architecture.

**Premise:** Disc Ex Machina is a competitive disc combat game in which the player(s) utilize disc throwing within a virtual environment.

**Theme:** The setting exists in a cyberpunk world of bright neon colors and futuristic sci-fi.

**Graphics:** Hand-developed using modern open-source 3D modelling software, and Photoshop.

**Sound:** Crafted in house by Mach G using FL Studio.

### Alpha Gameplay

![Alpha Gameplay Image](image_url)

### Mach G

**Who we are:** We are a team of aspiring indie game developers focused on industry-quality games that give an immersive, fun experience for the player.

### Virtual Reality

Virtual reality is the trending use of computer technologies to produce a realistic simulated experience to the user with strong emphasis on immersion.
Class Diagrams

I. Object Class Diagram

- **InteractableObject**
  - + bool beingInteracted
  - + void Start()

- **InteractablePhysicalObject**
  - + Vector3 grabOffsetPosition
  - + Vector3 grabOffsetRotation
  - + float throwMultiplier
  - + float angularThrowMultiplier
  - + void Start()

- **InteractableTriggerObject**
  - + void Start()
  - + void Update()

- **IPODisc**
  - + float spinVelocity
  - + float curveAmount
  - + void Update()
  - + void FixedUpdate()

- **Spawner**
  - + GameObject prefab
  - + Rigidbody spawnLocation
  - + float delayInSeconds
  - + float lastTime
  - + FixedJoint joint
  - + void Start()

II. Miscellaneous Class Diagram

- **SceneSwitcher**
  - + string nameOfLevel
  - + void loadLevel()

- **SpawnManager**
  - + GameObject objectToSpawn
  - + float spawnTime
  - + Transform[] spawnPoints
  - + int numberOfSpawns
  - + int numberOfSpawned

  - + void Start()
  - + void Spawn()
III. Destructible Object Class Diagram
Ethical and Intellectual Property Issues

Ethics

From the Utilitarian perspective this project maintains the set of ordinances determined by common morality. The software that is being developed aligns with public interest. This project offers an immersive experience that will first and foremost provide entertainment; but also to help relieve stress and give the opportunity for individuals to exercise and create an active lifestyle.

From the Respect-for-Persons standard this project is in compliance with the fundamental guidelines. The project offers an equal opportunity for individuals to excel in a new and upcoming realm of entertainment. The project work environment allows students to explore new ideas and concepts with upcoming technology to improve individual knowledge pertaining to certain fields.

The working environment and development standards follows current industry standards. For example, by providing a form of version control, creating proper documentation, and obtaining licenses for software we approach an industry level of quality. To achieve a complete product the scope of the project is dynamic and it adjusts according to features as they are implemented.

Intellectual Property

The intellectual property of this project is completely owned by MACH G. This is to a first approximation as per the University of Kansas's policy. The category for this project falls under the Student Academic Creations section. The gist of this section states that the ownership of the works submitted By the students for fulfillment of academic requirements will stay with the creators.

The current license on the game development software is a non-profit license offered by Unity. There are other licenses from Unity that allow for commercialization of products that have been developed utilizing the Unity game platform. With the different licenses the intellectual property remains a complete ownership under MACH G, and gains the ability to be commercialized. The current monetary value of Disc Ex Machina is currently free to play, but later down the road may develop into a polished game on the AAA game standard.
Change Log

The implementation milestones along with the gantt chart spread the project over the course of two semesters while the initial project proposal has a timeline that is spread over one semester. The reason for the changes were because the team developed a feel for the work progress and adjusted the deadlines according to the full work time available. By spreading out our workload, we aim to achieve more *momentum* as we progress throughout the year.