# **PROJECT DETAILS**

## What is the overall summary of the project?

## Motivation for creating this project?

**I WILL display commitment, faith, willpower, discipline and power.**

This project will give you your freedom, how long are you going to stay stuck? With this project all your hard work will be paid off, all the sacrifices and the countless hours of work you have put into this line. You will hit steam and then things will take off from there, Unity conferences for revolutionary technology, Steam and GDC conferences and more. Then fame, money and everything will come and you will have learnt so much throughout this experience, you would have learnt so much that you will be able to create a game like this again, but next time you will know what to do!

## PRE-DEVELOPMENT

Currently the entire project folders are wrecked and filled with trash, so I am going to go through delete scripts I do not need and delete scenes which have no use and then I am going to go through and begin building the editors to help me during the development process of this game…

I am restarting the entire game project, so I will be resetting the entire file –

* Create new project
* Create folder hierarchy
* Create folder system in Unity
* Import some needed

## Editor Tools and Assets Required (Buy Required Assets)

* Take advice from the experts
* What tools can you get from the asset store to speed up production?
* <http://www.procedural-worlds.com/blog/best-free-unity-assets-categorised-mega-list/>

**All experts** say that you should use other people’s tools if they are **within your budget and you don’t have the skills or time required** to make the tool yourself. Most of the stuff I am creating myself anyway so it is completely fine if you use some assets here and there in your development.

Your budget - $99 (Use for steam dev license)

Therefore you can use free assets and work on it yourself…

## Simplifying the development process (PRE-DEVELOPEMTN)

* Debugging can be made 1000x easier with some debugging tools
* The animations in the game can be simplified through some Mocap data.
* Use an editor script to follow through on the design pipeline (Automatically setup LOD’s and setup prefabs, collision (based on object)… Setup tags, layers, scripts, shaders, rotating the object to fix. Everything…. Editor gets description of the object and applies what is needed.
* Time management and tidiness is key, because then you will get lost. I feel like the best way to manage time for the game is in the unity editor so I think I will be creating some simple editors such as loggers to keep a journal of what I am actually doing each day, an auto backup saver.
* Main objects such as regional points (Make a menu item)
* Scene switching in editor easy
* Unity editor tool window which will have all the tools I need to repeatedly do… In editing or others.
* You have to make a tool for auto backups, all you do is connect your hard drive in and you click a button in the editor and it copies everything into a pre-defined folder and labels the version name.
* You have to make a tool for the GAANT chart and Changelog, you can add changes and then it adds it to a log file which will be included in every backup…
* The GAANT tool can be used relatively inside the project, for example you are working on a script and you need like 20 different functions inside that script you can just use that inside your GAANT… Local gaants…
* Use the new presets system in order to save camera/world/player presets then save that data and can be re-used in similar objects!

NEFARIOUS

# VISION

* Release it onto steam and be able to make this my first stepping-stone into the big leagues…
* When people play the game, they need to be able to say that they just played an AAA game…

**Okay so how can these goals be accomplished? (EVERYTHING IN THIS DOCUMENT IS ESSENTIAL SO DO NOT BE LAZY!)**

Optimize your time… Layout your systems and the things that need to be done in order to accomplish them and then make your own ‘pipeline’ which will allow you to prototype and setup all the aspects – design, dialogue, coding, materials really quickly…

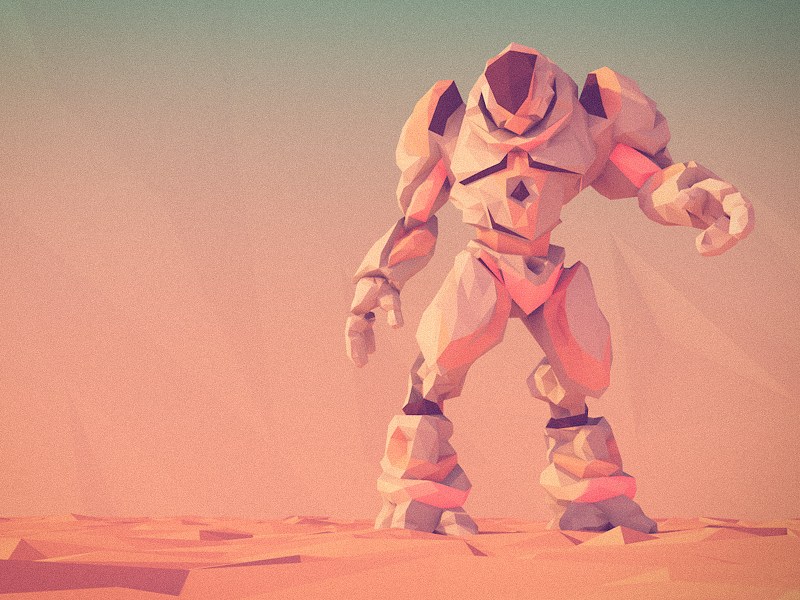
Focus on presentation… AAA is always about the developers going over the top on every aspect of the game, which creates a very polished look. So what if you could take the audio (voice acting) out of it?

# DESIGN ELEMENTS

## Presentation of characters is key

* Believable character animations (No Mocap)
* Player interaction be through another worldly voice, which can be translated through an effective manner on screen.
* Believable character lighting + UV Texturing
* You cannot accomplish AAA graphics, so how do we down scale? (LOW POLY)

Example - >



* This lighting is a very nice color
* The character is detailed yet extremely simple in textures and design but accomplishes a very nice effect
* The armour is extremely complex and detailed.

The creation of this character would have simply been to create the base geometry from a simple pre-defined drawing and then using the decimate modifier + triangulation to create the nice triangulated effect, and then using rigging tools to rig up the character. Alternatively, the character may have been rigged before decimated, decimation should be done last even after UV mapping and the non-decimated player should be saved separately. Most of the components on the body should be modelled and attached onto the player rather than modelling with the player. Some other needed features include shape keys of injury in the player, wearing down of the players outfit, wires and other mechanical particle effects shooting out to create a sense of realism, change in animation with blend trees as he grows more injured will also help with realism…

This witch can be used to represent the witches that roam the forests; it shows a very dark and powerful character which animations can be applied to, quite easily… You can show emotion through color and the sharp edges of the character compared to the light and soft edges of the previous character so keep all of that in mind while your design your enemies and yourself?



Some key issues with finishing this player would be –

* The process of designing such characters as the forest is an imaginative place where ‘ANYTHING’ can exist
* The topology has to be perfect so we can move the bones and get some solid movement.
* Animation of cloth…?

Therefore, for the player model we will just have to rough sketch and hope that we can model it in the game, but for the modelling part we can focus on the body first then apply the clothing ☺

## SOLID Animation

Therefore, we have some nice blueprints for our characters and we know we have to make good animations but what makes an animation feel fluid and powerful ->

* Inverse Kinematics (Watch GDC on predicting movement) (Full Body IK)
* Inverse Kinematics with personality, try and modify and copy Ubisoft’s Inverse Kinematics solution
* Every character needs personality and blending between animations needs to be seamless and no visible snapping between animations.
* Solid anticipation of movement (not just moving the player)
* Creating a sense of mass while moving the player or for example, carrying a heavy rock, he should be bent at the knees and struggling which can be easily achieved through Blend Trees and noises.
* Follow the principles of animation.
* The player is not perfect, fill the animations with stuff like the player falling, tripping, failing a jump and stuff like that because the audience get bored of watching perfect animations all the time ☹
* Weight simulation in the inverse kinematics would be achieved through adding extra force in the direction of travel from the object that the character is holding.

## Color Palette

What color scheme are we using, because it is always good to have a followed color scheme because it creates a sense of belonging inside the game and realism.

<http://colormind.io/>

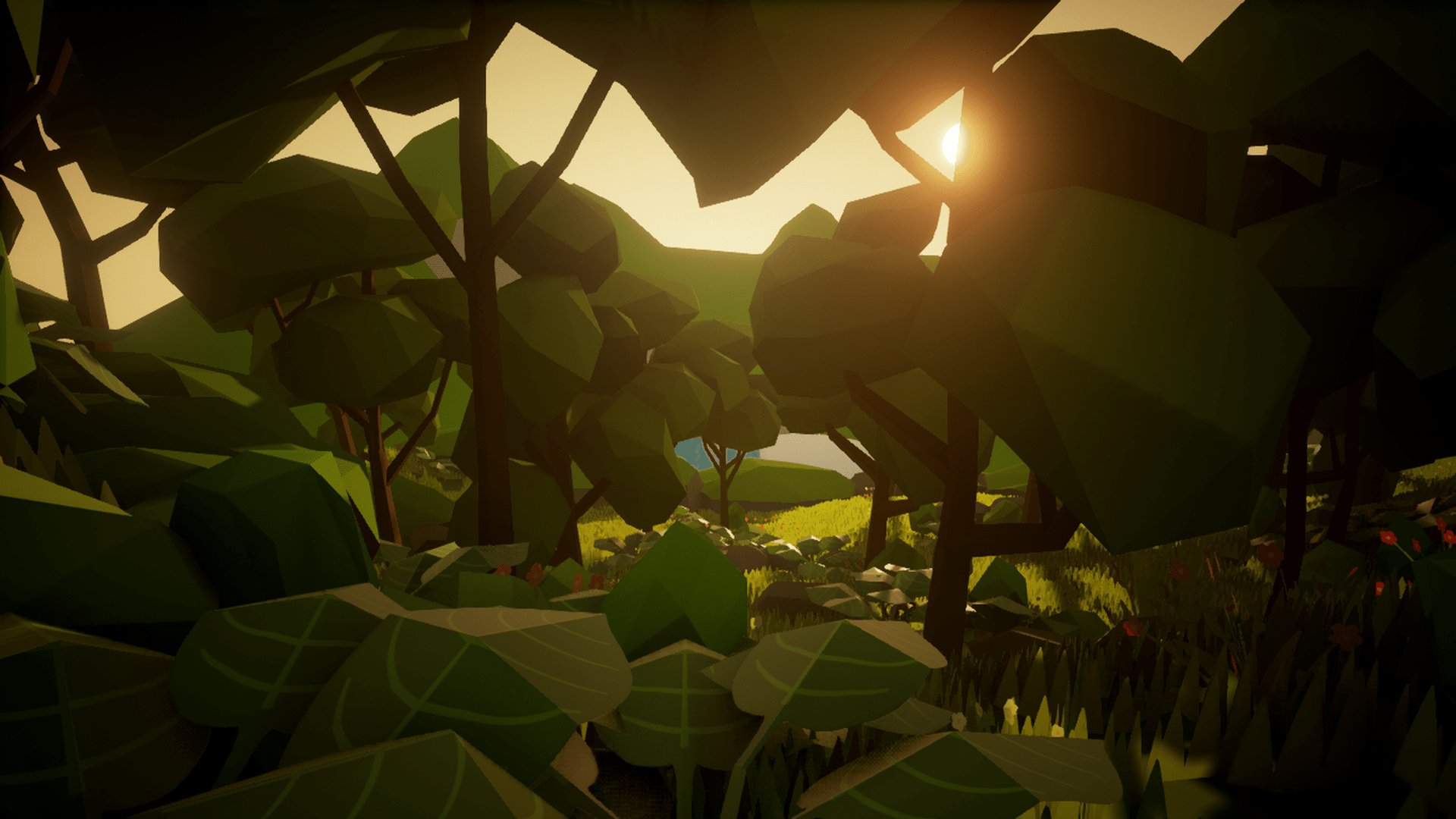
# WORLD DESIGN

## Graphical Features and World Design

Therefore, what design approach are we taking on to the environment and the world? The art style is going to be low poly and we can build a mysterious foggy forest through low poly trees that can be variated through code and or shaders… we can do that through vertex shaders. Hand placing grass, logs, brushes, twigs, flowers, mushrooms and so much more will just be assets that will modified through scale, color, particle effects, shape etc…



Rocks and such can be modelled through sculpting in blender, and decimated to create a triangular effect and will just have simple oriented box colliders on them, trees can easily be split into the different sections and in the pipeline can be dropped into a procedural generator which will just modify any element of an object that you pass in and will create a new prefab that you can save into your assets.



Placement of all objects in the scene can be done procedurally by setting levels such as tree density and ability to avoid pre-set paths and allow free pathway.

Another major problem to tackle is the creation of spline based geometry because there will need to be clear pathways for a while which will be a pain as they need to be textures into the ground + rocks and other normal features.

*The sun in the picture has lens flare which you will need to add and the skybox is plain with low poly clouds which you can add to revolve around the earth quite easily with procedurally generated clouds ☺ The lighting is not dull and is soft and always present with its own color and everything is fairly illuminated. Therefore, with reflective objects use reflection probes and with dynamic objects make sure there are light probes and make big non-dynamic objects into static! Bake your light maps and keep UV geometry simple so light calculations do not take too long, you still need to study more about the lighting system in Unity to create beautiful results.*

## Subtractive Design:

Subtractive Design is the design element that the famous game Journey used, the term implies to the design process where you revise on your work every now and then and remove everything in the game that does not add to the core gameplay and story. If the object or element is just there for the sake of being there then that needs to be removed, this just clutters a game with useless junk.

For example, a simplistic UI would be better than un-needed features on the UI, which would just break the user experience.

The full explanation of the Subtractive Design by a professional:

*‘Subtractive design is the process of removing imperfections and extraneous parts in order to strengthen the core elements. You can think of a design as something you build up, construct and let grow, but it’s pruning away the excess that gives a design a sense of simplicity, elegance, and power. ‘*

*‘Make everything as simple as possible, but not any simpler.’*

This system needs to be in your mind the entire time when you develop this game so you do not need to waste time going back and adding stuff. I know you will be tempted but just understand a good game is a simple game. ☺

## Post Processing and Shaders:

* Global Fog (with color maps)
* <https://github.com/wudixiaop/ShaderlabVS> (HELPS WRITING SHADERS!)
* <https://github.com/OCASM/SSMS> (FOG)
* Physically based shaders (interacting w/ environment)

**UNITY 2018 NOW SUPPORTS SHADER GRAPH SO NOW VISUALLY SCRIPTING SHADERS IS POSSIBLE SO YOU CAN LEARN INTUITIVELY!**

These links will help you in deciding what to do with your skills in shader programming, so how will you learn how to code shaders? When you get to this stage you need to use your 24 hours to learn about shaders 24/7, every second of free time use to learn about shaders so you can create beautiful games, until you have a knowledge of shaders so high that you can envision a result and your brain will accommodate the steps to get there. You have the udemy course and all the online assets so go and learn!

One of these assets will help with code completion in shader lab, this will help you experiment and try out different outcomes. This is what you have been looking for. Time to go!

Some effects that you will have to create include, a fog that will change color in terms of a color palate passed in! This will look amazing. You also need screen space reflections and stuff like that to make sure materials are interacting with each other.

One other shader you will need is the snow shader, a physics shader which will calculate shapes when you step in snow, or the mud it will look amazing and all you have to do is learn how to get shapes and stamp them into the ground (w/ depth)…? You will learn ☺

## Cinematic System:

The cinematic system can be an amazing editor script which I will go into detail right now ->

1. You will be able to move the camera around and record the movement of the camera and smooth it out using steering controls.
2. Everything can be done in scene view so looking and such will all be done there if you want or in game view and you can save the data.
3. Each recording will be done inside the editor and saved as a SEQUENCE file which has all the data of movement and transformations that occur during the scene (which you will have to predefine which objects are going to be moved and that data will be saved into a binary format)
4. In the timeline there is also the ability to drag in pictures, videos and audio and reposition and re-scale these elements to fit how you want it and moving it around and stuff like that, adding fade effects all can be done in Unity it’s self…
5. Then you can go into the Unity Cinematic Clip Editor where you can drag and drop sequences onto a timeline while adding fading effects and such and export that into it’s own file and then you can play that stuff during runtime by calling the file and receiving data from it.

This system will be revolutionary in the cinematic industry of Unity and if you just publish this and add features like, adding keyframes and event keyframes (after production because we don’t need it) and stuff like script control and post production control then we will be able to sell it for a lot of money on the asset store because a fully fledges clip editor doesn’t exist in unity at the moment and we can even have the ability to load in actual video and audio clips into the timeline if you want…

## Vertical Slice:

So what is the vertical slice, the vertical slice will be all the gameplay elements in the game and basically testing out what works, what doesn’t work, what to remove and add to the game and using subtractive designing to make the game simpler to appeal to the genre, you currently have a lot of features in terms of AI and such but most will not be burdening the player in fact stuff like the epic combat system will help with the game play aspects as the genre is an Adventure Action RPG style. Make sure you still play around with the graphics and become better at modelling and morphing shapes etc… But the vertical slice is a prototype of the game with all the base GAMEPLAY features, not really big on design.

All features including combat, optimisation, survival, inventory done, then you will work on the world design and entirely design and just adding the gameplay elements to the world. This will simplify the process because you can worry about the entire coding process at one half then you can work on the design phase which will be writing the shaders, modelling the NPC’s, animations, world design, environment design and all of the complex modelling and graphics (skyboxes, lighting) then you will go into post-production which is writing the menus, end screens and stuff like that.

Cut scene development will also be POST-PRODUCTION but the cut scene (editor) will be created in unity.

## Design Pipeline:

There has to be a set rule for setting up any new model or prefab or character in Unity from blender. So what is that process?

Phase 1 -> Sketch out and plan the object/character through references and images, sketch it out in 3D and plan how you will make it look good and keep poly count minimal. Think procedurally on how you will achieve the end result in a systematic way of thinking.

Phase 2 -> Collect resources (Reference images) and make the color scheme for the object first and create a color palette for the object (this is important)

Phase 3 -> Begin modelling and then make sure everything is perfect, topology for characters, the design of leafs and bushes -> Make variations and make it accessible in Unity (make sure you get the Rotational Fix Plugin for Blender)

Phase 3.5 -> Very important part of variating the object is to make sure you create at-least 3 LOD’s of the object so in Unity we can use the LOD system efficiently, even more is fine.

Phase 4 -> UV Unwrap the object and character and start to attach it to the color palate that you prepared.

Phase 5 -> Begin creating the animations and to do this properly you will have to add all the actions and animations onto the one dope sheet and I will manually cut it up in Unity as that seems to be the easiest and best way. Make sure the object is properly weight painted.

Phase 6 -> Add any morphing that you will need into shape keys and add that into Blender and import into Unity as well.

Phase 7 -> Go into Unity and setup the animations and animation controller if needed, as soon as you get the model add it to the respective (Prefab) folder by converting to a prefab and saving the textures and materials in their respective folders (Organisation is key)…

Phase 8 -> Set-up the LOD system onto the prefab, you may be able to automate this later using some type of script where you pop it in and it generates the LOD’s until then do it manually.

Phase 9 -> Begin coding the object and place as needed….

# NUERAL NETWORKS

## Use of Neural Networks in Combat Situations

* Honestly writing behaviour for combat can be fun, using your own custom behaviour tree and randomisation trees to mix it up using variables, or one other situation is giving it a bunch of outputs and inputs about predicted position and such and allowing the neural network to decide how to kill the player and it can be a real time network as well in order to predict and understand the player’s combo’s and what he regularly uses to counteract those things to fight back.

## Use of Neural Networks in Inverse Kinematics

* Inverse kinematics is a solution where it has many solutions and we most likely won’t get the most realistic simulation through simple algorithms (this could tie into the Inverse Kinematics Editor)… The network will get the end effector positions and the joint body from the IK editor and be able to modify the bones using FABRIK in a smart manner and move the end effectors to a more comfortable position… Or in a realistic way that a normal human would move, rotations and movement would take time and feel real.
* How training data would be provided? This is easy, place where the end effector should be and manually rotate the joints and reposition the body to how it should look like when the end effector is there and then save that data into one training sample. Repeat for thousands of situations and put into the system ☺

## Neural Network Editor

* This would be my absolute dream to create, an editor where you can drag nodes and connect up layers like, add an activation function here, a training layer here, a hidden network there and get some input data, and add a training method and some code execution. Then it creates the network from scratch. This tool can allow any developer to create AI in their code.
* Some things to remember is : You have to have deep understanding of the subject, the code needs to be optimised, the tools such as graphing has to be given, many types of networks have to be supported, easy implementation of data (training) and the ability to access output and input data easily. Easily able to write code for the network as well.
* Data such as news paper articles even can be added and the power is with the user, so they can define training samples and their expected outputs, see the graph improving see their activation functions and so much awesome stuff like that…. They could create a stock prediction network right inside unity! Give that power to people and you will be on your way to GDC or Unity Conference…

# CORE MECHANICS

## The Inverse Kinematics System

Therefore, in order for us to develop a character who is constantly interacting with the environment in subtle put powerful ways, we need inverse kinematics, which needs to adapt to the environmental situation.

* <https://www.bing.com/videos/search?q=ubibosft+inverse+kinematics&&view=detail&mid=9E31837AD2C18890818E9E31837AD2C18890818E&&FORM=VDRVRV>

This is a video of Ubisoft’s revolutionary technology, which uses IK that ray casts forward predicting motion using motion fields, this technology can be used with fabrik…. In order to learn the in and out of this technology you will have to watch the GDC talk but one major thing I will have to work on is implementing joint constraints which can effectively just be a cone. By understanding the process than we can implement into a simpler cone for our character. Then using that technology we can implement the IK Joints to give a sense of mass and give character and life to characters for example just sliding hand across wall when near and adjusting pose in tight spaces, crowd manoeuvring.

The inverse kinematics system uses neural networks so if you decide to go down that path of making stuff that complex then you can of course, or you can **make an editor in order to run IK only. Behaviour editor for IK for example functions like Ray cast Right from Right Hand, get hit point and move right hand to ray point at a speed of 3… This editor can work directly with fabrik and can be used to emulate weight, comfortability and stuff. Editor fully!**

Regarding the issue of constraints in inverse kinematics, I am getting extremely close to solving this problem and as soon as I do I will upload on YouTube but until then, I am not uploading as if I upload I want to only upload latest and greatest content…

Inverse Kinematics Features:

* Works with animations
* **Un-Comfortability feature (So basically IK calculated where the feet should be placed or something right because it predicts motion, when the player stops then it can revert to a more comfortable position)**
* Balance (Over balance will send a message to fix or un-balance animation) Calculate through finding centre of gravity.
* Realistic outputs (Move only how much it needs to move)
* Forwards casting
* Search function for best placement, not just directly down
* Multiple effectors throughout body (Leg affects body vice versa)
* Work in range of situations
* Intractable with environment
* Constraints (Cones)

## Player movement and World Interaction

The key to success in the field of input to responsive movement is proving that the player is in control at all times, so cut scenes need to be as minimal at all time, ALL combat needs to be done through player interaction and no scripted combat… The combat needs to feel realistic and responsive as well alongside the movement; the animations needs to directly translate what the player is aiming for.

The core movement of the player will be very smooth and powerful, but needs to be extremely realistic, first we will talk about general movement and then we will talk about combat and those systems…

The key elements to realism in player movement is:

* Personality
* Anticipation
* Mass Consideration
* Interaction with the world
* Current state (injury and accommodate for that)
* IK to manage proper handling of object
* Smooth
* Reacts to events around him (no cut scenes)
* Responsive
* SFX at every turn (responsive)
* Speed of player w/ current state

## Combat!

Awesome Combat – Overgrowth is an example that is explained as: *Overgrowth’s combat is unique because of a procedural animation system that calculates the impact and consequence of every blow. Even the rabbits’ ears are coded to flex and bend accurately when rolling under a sabre swipe or delivering a spinning roundhouse to the body of a wolf.*

Combat that would feel good consists of elements such as:

* Procedural and responsive combat (All blows accounted for, including force, point of impact and all)
* AI is not stupid, they are unpredictable and can fight you as effectively as you can fight it, all AI have their own fighting style and pre-set combos so you cannot replicate your moves on the same species all the time…
* Anticipation, in order for you to truly master combat, you must learn about the enemy you are fighting and learn it’s unique timing and what is follows as regular patterns and this can change, but every second matters, the enemy may have the guard down for less than a second where you may need to strike.
* The bow and arrow combat is really the type of game I am going for here, but the bow and arrow cannot be a simple system to just master, you have to account for the wind and the dropping of the arrow… The longer you hold it the more force you will get but do not overdo it or your bow will break and then you will have to fight with your fists, which is more dangerous.
* All enemies have their own assortment of weapons, skills and magic. ☺
* No snapping between animations during combat, everything is procedural; the AI is procedural and can anticipate your moves. Therefore, it does not exactly know what you are going to do but it can pick up patterns in your behaviour and implement that into its own system to predict what you are going to do.
* AI has personality, it detects danger and can run when it is weak and injury affects its performance as well as your combat performance.
* The AI I repeat has to be believable and smart, so not only does it have its own niche such as wizards with magic, but it is also able to use its environment such as climbing up trees or harnessing a fallen sword that you dropped. It is aware of its surroundings and in the end it may decide to run if it is going to be beaten or anything is possible… It may be able to use its fists even, skill level may vary based on how experienced it is with a weapon, and that will affect the animations of how accurate it is.
* NO PRE-SCRIPTED COMBAT! NO HACK AND SLASH. ALL COMBAT IS TACTICAL AND MAKES UP A BIG PART OF THIS GAME. You are not a hero you are just a man, so you must be careful…. In addition, you have to protect your friend as well.
* Injury will affect your aiming, speed, stamina. Stamina is affected by the amount of combat you do so you have to fight smart or long distance. Stealth is a big part of this game as well.
* You can run away from combat without any consequences unless the AI decides that you are weak and decides to pursue you, unless you were crossing its territory.
* AI can call upon friends and defence and backup, also it can run back to its horde.
* Secondary animations are key in order to give a sense of beautiful combat, swinging cloth, clashing of swords, epic sound effects.
* Smooth and animated beautifully… Procedural, curves blend trees everything together… Foot placement, personality, simple key framing… ☺ Curves are never linear but has some type of effect placed upon them, which you can experiment by making your own animation system if you like or learning how to implement these curves into blender, then importing that.
* Physically based combat, you can fall trip and are injured; you have no guarantees because you are the player and you have no targets on your back because you are the player.
* Ragdoll physics when the player dies and a nice death animation, death needs emotion… and personality. People do not just die! They are angry. ☺
* OKAY SO COMBAT NEEDS TO BE EFFECTIVE BUT NOT UNPREDICTABLE AND WONKY….

Animations need to be interacting with the player state at all times and needs to anticipate where and what the player is going to be doing next so you can play the next animations.

If you pull off this epic combat system then you will forever be known as the guru of combat creation in games! ☺

## Realism and a Co-Existing World (Independent Time)

The key to a realistic world like in assassin’s creed has to have an independent world that does not rely on the player in order for events to occur!

In order to achieve this effect I will have it so that the forest is it’s own independent life form, with it’s own backstory (without the player) and clans, fights, class rankings (masters, servants), magic and more…. [YOU WILL HAVE TO DEFINE LATER]

But the spread of these creatures and events has to be nicely created, so I want to create a perlin noise map and actually move the perlin noise map in order to create horde movement and fights. Everything can be displayed through perlin noise, but it doesn’t have to be used, but it can be used to make sure every time a player opens the game, the enemies are different and maybe one enemy is more powerful, maybe the enemy isn’t there at all… The only reason we can achieve this is because the forest has to be massive!

Obviously optimisation will be a big problem so we will have an on-going state machine will define what state and area is in, and all the states of the NPC’s in a low cost solution and then when the player enters that area the world will update to accommodate that solution.

## Realistic NPC

So what does a realistic NPC really need in order for it to ‘feel’ realistic (I say neural network) – but this website says

<http://d20despot.blogspot.com.au/2014/02/making-realistic-npcs.html>

We can implement this type of behaviour through behaviour trees (but not just any behaviour tree, a dynamic system with loops and predefined rules that checks if something has already ran and more stuff like that)…

You need to make sure all the NPC’s can be killed even if they are vital for the player’s to know some information… Maybe indicate they are important but give the player the choice if they want to kill them or not. Because people aren’t indestructible so who cares…

Realistic NPC:

* Procedural NPC’s (don’t want look all NPC’s the same, so why not procedurals their entire body?) like different capes, symbols to represent clans,
* Customizing the shape
* Customizing the color (cloth, skin tone) everything modifiable but dynamically not through shared materials.
* Adding props overlays and meshes
* Lightning fast reaction to changes!
* NPC’s need motivations beyond the Game’s motivations, what drives them to do what they are doing.
* What is going on in their life right now, why are they doing it, are they rebelling against something, why do they hate humans? Why are they even attacking you???
* NPC’s need limits, they have to understand what is worth it and what is not, like explained in the combat section, if they sense defeat they try to run if they are the scared type.
* NPC’s need power and learning… Memory also is important, NPC’s can remember you from before and other features.

Procedural creation would be done through sliders in blender, even morphing shapes can help with this! Clothes also have to be different even amongst species, every small thing count.

All of this needs a extended editor which needs to be built that supports proceduralised character building, extremely complex behaviour editing with dynamic script controlling and variable control. This will be very high quality and will take around **4 weeks to build.**

## Realistic sense of Time

Time is not unlimited and time cannot wait around until the player feels like moving, so making time independent can really help with increasing the realistic ability of time, things affected by time will be the injury of your friend, starvation, sleep and all the survival meters are mainly affected through time, and eventually if you just sit there you WILL die. That is a fact, and time also means so many more things, your enemies can heal as well, not only do you have the ability to heal. Due to the independence of this world from the player the enemies may move locations and do new things, rebel who knows…

Day night cycles need to be realistic so you can use volumetric cloud shaders (one that I saw for free) to create a nice sense of day and night. The colors are not the only things that change, weather, the fog, the animals that come out, the sounds change, everything goes more still and eerie.

The NPC’s are all enemies but they have their own agenda with their own leaders and beliefs, each section of the forest has a backstory of how the humans used to tamper with them and therefore humans have been outcast from the forest but they used to all be one.

While you enter the forest an old man (realistic Ai, you can talk to him if you want but he is not waiting for you to come up and talk to him, he is just wandering the forest while you are healing your friend and he will tell you about the war that is going on currently.

# GAMEPLAY CORES

## Regional Movement System:

The regional movement system is used for optimisation and will be used through an editor script in Unity, the purpose of this system is to notify the game whenever the player is in a region so appropriate steps can be taken in order to optimise the game. This will be done through the means of disabling NPC scripts, using automatic LOD’s to re-detail the game in order to improve graphical performance.

The way this will work is through custom doors as such which the player which will pass through and then he will be in a new region and the doors will be able to cast out and find out which surrounding regions are still in range and keep those LOD’s at a high level in order to keep graphical performance + quality.

## Inventory System:

The inventory system will be very powerful and dynamic and will not require the user to open another window just to be able to get some items. I want an inventory system like assassins creed where a circular UI element opens up and you can select a region out of there for example (medical, swords) and regions like that and you have some limited an allocated space for each item in your backpack. You can just press a button and it will open up (translucent) so you will be able to see the background and then you will be able to select a region and another circle will open up, select your item and go.

The menu will display some states about the item, examples include damage, capacity, amount left, value and stuff like that.

## Regional Event System:

## Survival System:

* The survival system is at the very core of this game

### Crafting

* Crafting in this game has to be very minimal and realistic, I do not want any fancy UI and or construction patterns or complex things in this game. I want it to be very straight forward and obvious but without much UI at all involved.
* Crafting will be achieved through destruction of the environment, not everything is destructible but most objects such as rocks, wood, branches, water, steel that you see here and there, statues can be broken down, houses that you see can be destroyed, marble…
* You can pick up objects such as steel, arrows, swords and build furnaces and melt the steel and build other things out of it.
* ***If you’re going to build a crafting system make it in-depth and detailed. Crafting items and combining it with other items.***
* In order to make this system you will have to create a table of all the items that are base level that can be acquired and go from there to see what can be combined to generate an output.
* There needs to be an abundance of things or it will just feel like a system thrown together for the sake of throwing it together, if you make a feature make it good and polished.
* ***Remember MINIMAL UI! The less amount of UI + more power the better ☺***

### World Destruction

* World destruction needs to ‘FEEL’ real because I don’t want a Minecraft style mining where I’m just hitting a rock and nothing is happening…
* For example hitting a rock, as you begin hitting it you just get steel on rock sounds, but then soon you start to hear cracking sounds and a crack starts to spread throughout the rock slowly but surely and then the rock breaks. This can be done procedurally I’m pretty sure.
* The rocks hitting has to split into multiple pieces and just fall down.

If this cannot be done procedurally you can break up the objects that you need inside the Blender and then you can just use some rigid body explosions to push them away from each other, at least we can use Rigidbodies for that hahahah. If you can try and learn how mesh vertices work and break them up and try fill the edges or something. I just want to tell you, that people will understand if everything is not AAA standard okay, because you are just 1 person and a (15 yr old) in his room making this game, okay so don’t stress the minority details like this one.

### Life Meters

The different life meters in the game will be what affects the game the most in terms of playability and style, I need a lot of these meters and the fact of the matter is that the player has to try and manage all of them and I will get to the different meters and all their effects/causes ->

This meter includes to both you and your friend and both of you will experience similar losses when it comes to each meter.

Main meters include:

* Hunger, Thirst
* Sleep
* Health
* Stamina

Side meters that appear when they need to be there include:

* Breath under water
* Ammo (Not exactly a bar just a number w/ a icon at the bottom right of the screen to tell how many arrows left)
* Bandages
* Bleeding rate
* Sickness
* Torch (Light battery)
* Campfire life
* Weapon damage
* Mission (Description + Completion)

Most of these won’t show up on the main ui and if they do I want it to be faded with slashed lines like a near future look but clean no techno stuff.

### Health system / Damage system

* Damage can be realistic if you try!

For example if you shoot an enemy in the shoulder then the IK in their shoulder needs to be knocked back and they should have an arrow sticking out and blood coming out from there (hole there). They would be weak in that area so do some calculation to cut of circulation from that part of the arm. For example if you shoot an enemy in the leg the game should know that they are supposed to collapse to the ground, you cannot run perfectly with an arrow sticking out of you.

Different regions of the body do different damage, one straight to the heart or head is instant kill and the neck should also be instant kill. The shoulder areas can be a little damage where the enemy can pull out the arrow and keep charging or something.

Death animations need to be a mix of ragdoll and animation. I don’t know how exactly this will work but for example if you get shot in the heart then your arms will move to the arrow and you will drop to your knees and die… IK with the ground then I guess.

## Map System:

* The map system is an essential part of the game as it will provide the essential navigation throughout the game, you need to have a physical map with a predefined region of the world (which comes from the camera)

So just put a camera at the top that encompasses the entire map and then take those pixels and then you convert the colors to pixel and you push it into a render texture and then put that render texture onto an animated plane which can be held by the player…

The things that need to be on this map, is the signification of moving hordes of enemies, evil lairs, healers, food places, lakes, gardens, houses…. Everything is there and you can set paths and it will show up with a marker so you can get there easier.

**THE MAP NEEDS TO USE IK SO IT CAN MOVE ACCORDING TO THE HOLDING MAP ANIMATION...**

## Ledge Climbing/Traversing Terrain

* Ledge climbing will play a major role in this game because of the rough terrain, you will have to traverse many rough terrains throughout the game. For example there will be a part where you have to climb a vertical mountain and the way that this will work is through pre-defined points.

Set up points along the ledges and this will represent the hand placement and all you have to do is use IK in order to suit that position and just move those arms along those points and move the body. Then when you are jumping to another ledge will be so complicated but you can do it…

Jumping to another ledge will be done through calculating which ledge the player needs to jump to through the user’s input, then you have to calculate the distance and velocity required to get over to that ledge, then play that jumping animation from the ledge and make sure the body warps towards that direction.

* Ledge climbing has to be smooth
* Responsive
* Good animation quality in the jumping

# AUDIO

## Audio and Voice in the Game

Voice Acting : Due to the limited resources in my game, voice acting cannot be existent and we can resort to a silent game of two technological beings who understand each other because the communicate through hand signals and create an emotional bond through communication.

SFX: Sound effects can be unlimited throughout the game, but it needs to be high quality and powerful throughout the game. Some of the sound effects such as gunshots, bow and arrow and nature sounds and bird noises I can record in my own backyard and most of the noises I can record and setup. Some audio effects I can buy if 100% needed to the development to the game.

Audio Engine: …? For mixing and fading the SFX. Because most SFX is very rigid, so you need to fade in and out which you can do procedurally through code and then you can learn how to mix up sounds and everything in Unity.

## Music:

Music is crucial to game development because of the lack of the voice as well, now that I have a Windows Computer we can easily get FL Studio, get some plugins, and develop music… Music has to have proper blending between moods and different tunes that can repeat seamlessly, different tunes for each region.

The music has to be atmospheric and beautiful using orchestral music and imperfections, please take a course on this and learn some basics on music theory… This music uses the same tone in many different varieties, which will help in your game ->

<https://www.youtube.com/watch?v=3jWB_YidG-c>

* Music like this while having a boss fight… or calm while traversing dangerous terrain (all will help)

Atmospherically Calm + Power

<https://www.youtube.com/watch?v=hcjFMcX6oEk>

Subtractive design can be used in music as well, learn from Hans Zimmer, study the music. You can be famous for creating the most beautiful music in the world as well you just need the right plugins/software/courses. May take more than a year to perfect but if you can create music like this then I am down!

# COMMUNICATION

## How is communication achieved?

The friendship of these two guys is the centre of attention when it comes to the story part of the game which is EXTREMELY intertwined so how do you communicate friendship and love for each other when you don’t have a voice?

Gestures and hand signals + mumbles + voices. Chatting through dialogue seems extremely unnatural and unrealistic so I want them to be able to effectively communicate through actions purely, you can understand when someone cares for you without them having to talk, actions speak louder than words.

For basic words such as let’s go this way all you have to do is lift up your hand and point in the direction and the IK of your friend will look at your face and will nod in response. And for asking things like are you okay just he will put his arm out and thumbs up and then down and he will reply with thumbs up or down.

One important thing we have to keep in mind is that the friend also needs to be able to communicate to the player, so even randomly the friend can ask if you’re okay, or hurry up, or let’s go this way. If he’s in the lead he will do these actions, make sure they make sense when they appear…

Dialogue plays a big part in the actions carried out inside the game, for example if he is leading and you tell him to ‘STOP’ and then ‘GO THIS WAY’ and you both die, game over…. So since there is no dialogue or predefined terms the dialogue is very dynamic and can directly affect the game’s direction because technically this game is free to explore except the first 10 mins or so to make sure that the friend does get mortally wounded.

* STOP
* LETS GO THIS WAY
* SLOW DOWN
* HURRY UP
* ARE YOU OKAY?
* DID YOU HEAR SOMETHING?
* KILL IT
* HIDE
* DUCK
* RUN
* QUIET…

Not only the main characters have communication, even the NPC’s and enemies can connect and interact with each other for example saying like KILL HIM or GO GET HIM or SILENT when creeping up to us. (ALL LINKED TO BEHAVIOUR TREE) . Even during horde movement they will all be chattering in their weird languages and connection with each other, even some of them will be fighting and being NON-STATIC!

## Dialogue System?

* The dialogue system has already been created but can be used anyway. For sequences where stories need to be told you need to use this dialogue system which won’t be often. I would like this whole gaming experience to be a silent one, with music mainly taking up course in most of the parts in-stead of voice.

## Emotions and Conveying messages?

* The main way to convey emotion and messages is not by pushing it into the audience’s face, do it like journey (the journey of life was the story) this was engrained in the environment, music and design.
* The music is a main part of conveying emotion for example when your friend is about to die you want slow and extremely sad orchestral music but where there is a sudden burst of emotion of anger there needs to be upbeat beautiful music.
* The environment also plays a big part in the emotion for example you wouldn’t expect a sunny day when someone important is dying, you would expect to see rain, storms and lightning. You can also use the environment to tell a story, for example rocks with engraving (you can destroy it) but the user can find hidden secrets about the world and the message that you are trying to convey is that friendship is powerful.
* You drag your friend through all the storms, sacrifice your life to save him and all of that pain and in the end he sacrifices his life in order to save you (he doesn’t die but he takes a risk to help you)… When you are backed up he takes a shot for you or something.

# GAMEPLAY OPTIMIZATION

## Debugging

### Organised Debugging

Debugging needs to be organised because you need to know from where are what time this message is coming through this will help you in so many aspects because believe me things will get complicated so you will get that editor for organised debugging and you will always structure your code in order to tell me what script and object the message is being sent from in this way:

‘SCRIPT NAME in OBJECT NAME : MESSAGE NO. ----‘

## Code Optimisation

### C# JOB SYSTEM IS HERE

In order to harness the full power of the C# Job system you need to learn the architecture of the code: <https://www.youtube.com/watch?v=AXUvnk7Jws4> The C# System will be released very soon with Unity’s 2018 update which is going to include so many new features such as the visual shader scripting tool, finally!

* Writing high performance scripts : <https://www.youtube.com/watch?v=tGmnZdY5Y-E&t=1270s>

### MULTI THREADING

* You need to handle your data properly and watch this talk and implement these things inside your actually game
* Learn how to control GC
* Multi-threading so it can run on multiple processors at the same time so it will speed up so much!
* Generating efficient code
* Link up with C# Job System

### REGIONS

Region out your code in a clear manner which will allow you to enter a script and know exactly where you have to go inside the script to do something. For example region out your movement and then inside that region, region out the velocity creator, jumping and inside that region it out even more. The more regions the better.

* Whenever you begin to write a new massive block of code, plan it, flow chart it then you can create it in code. Then you can understand in flowchart form how this all works.
* Always write down and document what the code actually is and what it does, and inside the code comment out any regions and new blocks of code to clearly explain what you are doing there.

NEATNESS

The neatness of the code and organisation of your scripts will maximise your productivity and will help you in future. It will also motivate you to write more code!

## Graphical Optimisation

### New Features

With the release of Unity 2018 there is new Scripting Rendering Pipeline which is a new architecture of Rendering so it will allow for high definition power in the graphics! You need an option to switch but it is super amazing!!!! Will help so much with this game!

### GPU Instancing

There are many thousands of way to graphically optimise something but the way we are going to do it is basically amazing, for example in the game I would like grass to be 3D therefore you will need millions of copies in order to do this you can use a technique called – GPU Instancing (<https://docs.unity3d.com/Manual/GPUInstancing.html>)

GPU Instancing can be done through the particle system now which is grown even more powerful throughout the update. Now unity is getting super powerful and I cannot wait for the updates!!!

### Texture and UV Mapping Optimisation

What about optimisation of textures and such, do not individually have a UV map for each and every object, use 1 texture map for nature and then use the UV to set the colors so all nature in the game will follow a color pattern and then can be modified through shaders and image effects, so this will have a nice optimisation.

### Lighting Optimisation (Lighting Information)

What about lighting and shadow optimisation, make sure EVERY NON MOVING OBJECT, trees/terrains/ etc… are always static. Then you can set the light map so 99% of all objects are static and the player and the animals which will only be initiated when needed will be dynamic so how will you get that lighting on to the player and such…? ***You will use the lighting probes and reflection probes, these are all 100% needed and can be disabled and enabled with the terrain optimisation system. This will allow the dynamic objects such as your player to get the sufficient amount of lighting….***

Lighting links/tutorials: (LIGHTING SETTING DEFFERED OR FORWARD) LEARN THE INS AND OUTS OF LIGHTING SO YOU CAN MAKE IT AS BAEAUTIFUL AS POSSIBLE! + SHADERS [*https://www.youtube.com/watch?v=eGu9\_8HS2uI*](https://www.youtube.com/watch?v=eGu9_8HS2uI)

<https://www.youtube.com/watch?v=VnG2gOKV9dw>

<https://www.youtube.com/watch?v=fRHMB4MWSFA>

Billboard trees, when you are extremely far away from trees but still want the visual of trees or anything else, you can just simply use a 2D plane with a tree texture and you rotate it all in order to face the player (Make sure all the trees are combined into one mesh!) Or something similar…

## World and terrain optimisation

### World Optimisation

World optimisation will be done through the regional movement system -> which will change and optimise the world based on the player’s position relative to the map.

### Map Door System (Optimisation)

* This will just be a custom AABB bounding box which will check the player’s position when dist is lower than x.
* The doors will be activated whenever the player moves into a new region.
* These doors need to make sure that they realise that surrounding regions can be seen so quality has to be preserved.

# PROJECT MANAGEMENT

## Creation and Development (Time Management)

* Once all the sections have been filled out you will begin to set a time limit for all of the aspects of the game, making sure that you keep in mind all the various holidays, weekends, weekdays and the work hours you will have to put into school then you will have to add more hours just to be sure.
* Once you calculate hours per task then you can create a view where you will list all the activities from your GAANT chart which you will map out based on the previous data.
* In this view you will write down the expected hours for completion and every time you work on that task you add hours and you can track each task in an elegant manner.
* You need to have an organised project folder, with project files, ideas, sheets, journals, logs, assets, art, concepts, papers, articles (All regioned out)
* Before you start the project I would like to create a whole new PROJECT from scratch because the other one feels old and I will set up a new git repo which will be well managed. The player controller will be done again as well.
* The assets folder has to feel nice as well with a 100% polished look and no just dropping stuff whenever and wherever you feel like it, the more organised you are the better.
* You need organisational editors to start.
* Time management and tidiness is key, because then you will get lost. I feel like the best way to manage time for the game is in the unity editor so I think I will be creating some simple editors such as loggers to keep a journal of what I am actually doing each day, an auto backup saver.

## User Interface Plan

* The UI In the game will be minimalistic at all times, with a black tone and white text but slightly transparent and will only show what is necessary and will change according to the user’s needs at any time, but they will always be able to access any UI at any time.
* Before you start make sure you design the different UI screens and what they will represent and how you will animate / move them. **This is 100% needed for a successful UI.**
* I would also like a natural movement UI, so if the user tilts to the right the UI will slightly damp to the right and snap back to give a sense that the player is wearing a helmet and that the UI isn’t just one static object.

## Development Blog and Recording

* During the creation of this game, I will personally be hosting a blogger website which I will keep updating the current logs, at start I will write about the story and the idea. Then I will tell people all the features I need to implement and sector of the website into chunks of different sectors where I will take people through my development process and the difficulties I had and the way I will be overcoming those difficulties.
* I will not have to worry about the theme and all of those small things, It’s going to be a simple website until I finish the game, then I will transport all the content over to WordPress.
* In order to create my changelogs, I will have an editor tool which I will be inputting my changes and it will automatically record the time, details and other things into a log file which I can read at the end of the day and then be able to copy paste into the website.
* At the end of every day I have to use my backup tool and save it to my website, or at least save it to GIT.

## Publishing and Deploying

* This game will be deployed to the Steam store using an accurate price point (slightly higher when you buy on steam rather than the original website). My website will be a parallax website which will allow the user to buy the game through credit card and an activation link will be emailed to the user (so no one can copy) then you will input the activation link into the game. The algorithm to get this will be extremely complicated.
* [**https://github.com/Facepunch/Facepunch.Steamworks**](https://github.com/Facepunch/Facepunch.Steamworks)

Steamworks

* The steam integration has to be very powerful ( will not be including multiplayer in the game ), overlays, achievements, friend comparing, trading, **STEAM WORKSHOP**….
* Through these features the games may take up a life of its own, when you finish the game you can extend the life of the game by having a f2p mode where you survive in the forest alone. **Probably not going to happen…**

## How will you deploy to steam?

* ­­By going through the rigorous steam selection process and it will consume a lot of time, at the same time I will begin an independent website which will sell copies on there.

## Pricing?

* <https://www.inc.com/guides/price-your-products.html>
* <https://www.helpscout.net/blog/pricing-strategies/>

## Audience?

The audience of this product will be:

* Consumer Type : Gamers
* Consumer Age Group : 12-25
* Targeted Platforms : Steam, PC, Mac
* Consumer Location : Worldwide

## Things I need to learn

* **Shader programming**
* **Keep pushing (Learn what doesn’t work and apply)**
* **Animation design**
* **Modelling (Low poly)**
* **Rigging to a higher level**
* **Better editor scripting**
* **Full body IK’s**
* **Writing meaningful and powerful stories**
* **Level designing in a powerful and simple way**
* **Creating high performance editors**
* **C# Job System**
* **Writing efficient code**
* **Organising the project and my time**
* **Optimisation**
* **High level procedural animation**
* **Quality movement**
* **Vertex and mesh building (in unity)**
* **Lighting and Image Effects**
* **Camera Effects**
* **Data management**
* **Project Planning (Higher Level)**
* **Time Management**
* ***STICKING WITH IT!***

## Estimated End

* **Will be addressed in project scheduler**

## Project Management Techniques

* <https://business.tutsplus.com/tutorials/project-management-kickstart-how-to-tackle-large-projects--cms-24612>

# ARTICLES AND LEARNING PAPERS

* Building Immersive Worlds
* Free Quality Assets
* Building a game intertwined with the story
* Conveying a story effectively
* Good game designing
* Getting an emotional response