

Project Planning Document
for the
Heavy Duty Machine Auto-Rigger (HDMAR) Project

BCIT

01 09 2021

Contents

[Contents](#)

[Executive Summary](#)

[Overview](#)

[Technical Summary](#)

[Project Scope](#)

[Out-of-Scope](#)

[Equipment](#)

[Hardware](#)

[Software](#)

[Evaluation](#)

[Target Platform \(Maya\)](#)

[Unreal Engine](#)

[Scheduling](#)

[Development Plan](#)

[Milestones](#)

[Updates and Maintenance](#)

[Work Environment](#)

[File Formats & Naming Conventions](#)

[Art](#)

[Setting](#)

[Minimum Viable Product](#)

[UI](#)

[Mockup Design](#)

Development Team: Tyler Millossi, Windy Wen and Rodolfo

Executive Summary

Title: Heavy Duty Machine Auto-Rigger

Platform: Windows (possibly Mac and Linux)

Target: Non-Technical Artist, Students

Release date: April 16, 2021

Publisher: BCIT

Overview

This is a collection of rigging scripts that have a focus on heavy duty machines. With these scripts the user can take a heavy duty vehicle (excavator) and create an auto-rig for the purpose of animation. We will demonstrate the effectiveness of the rigging script by showcasing a final animation of the heavy duty vehicle in a VFX video. Our Video will include excavators playing poker animations.

Technical Summary

Heavy Duty Machine Auto-Rigger will be developed in approximately thirteen weeks with three expert technical artists using Maya. A 3D Model of an Excavator has been provided with textures for the purpose of this project. This project has no revenue plan and is for educational use. There are no monetary resources required.

The tool will be able to be installed by the user with drag and drop functionality in the Maya viewport. After the user has done so, they will have access to a toolbar shelf with tools inside. These will be basic scripts that will simply call the functionality from the scripts folder where the tools have been installed. These buttons will have icons that correspond to a function (main functionalities listed below and a UI interface).

The scripts will contain classes that are modular so as they can be used in many ways. Therefore, a lot of variables will not be hard coded. For example, there will be use of placeholders which can be used for allowing the user to pinpoint where the pivot should exist for joints and

subsequent animation controls. Since there are a lot of similarities between these controls such as setting and getting their location, they will all be derived from a parent class and developed upon.

The main functionality will be broken down into:

- creating a root controller with a COG (compatible with any rig where there is a root control used to position rigs)
- creating arm controls (like a human arm but specific to the form of the excavator)
- creating auxiliary controls (these are nice for getting a door to open, steering wheel to spin and more)
- creating track controls (these controls will be rotating the overall track and the wheel components involved).

Once all these main functionalities are implemented, we will have our minimum viable product for our code.

I will now describe how this will be achieved:

Template

With the UI selected from the tool shelf, the generated main UI will contain a list of possible models that have all the functionality prepared for the user (Select Template (enum)). There will be the excavator(enum), and this is in the UI for the purpose of streamlining the process as well as adding functionality beyond the scope of this project for continued development. With the excavator enum selected, a description and an option to create a proxy rig will be visible. The Proxy Rig for the excavator includes the three main functions (root controller with COG, arm controls, track controls). The process of creating the proxy will have the steps that are required from the main functions but eliminate the need for the user to find them.

Root controller with COG

The root controller with COG options will be in the subsequent tab and labeled "Master". This will also be accessed with the proxy rig

mentioned above. The Root will require the user to specify what geo will be included in the rigging process (text field with a button(add selected)). A joint placeholder will be provided and the user can move it to the pivot of the excavator cab (COG) and assign the geo. These will be FK controls.

Auxiliary controllers for slewer, door and steering

The controls for the single pivot components on a given rig can be accounted for with our "Body" tab. This will take into account some models may not have or require such controls but have them available should the need arise. The user will need to add the geo (text field with a button (add selected)) and set the pivot for each auxiliary control. These controls will be FK.

Hook control

The arm that is used to move debris will require the user to set up the pivots (using placeholders) and geo. Once the user has completed this for the Bucket, arm and boom the script will provide functionality to use both IK and FK control options.

Track controls

The Track will be controlled either in part or in whole based on user setting. They can add the geo for each rolling component in the legs with a pivot. This will give them controls to animate them all together or separately. There will be an option to use a track that is provided to create the tread or to use the tread that the model has. There will be an issue with our model as it does not have the tread that can be animated easily. So the option to use a provided tread will be the only option we could test on this model. We may decide to use a texture UV position later for this situation on the tread. The animation control for the tread will contain various attributes that give the user options to rotate the treads based on the movement of the main control or the individual leg movement or the individual components of the leg.

The rig will need to be animated by a person using the controls generated. The movement of the excavator will be done by a person of our team for the purpose of the demonstration video.

The video demonstration will be a poker game between heavy Duty Machines. They will be scaled to the size of a bottle and pick up and throw chips. We will take our animated model of an excavator into Unreal to create a video. There will be texturing done in Substance Designer for the poker table. There will be destruction effects in Houdini and simulation for the chips.

- Toy excavator playing poker with other excavators
- There's a big mountain of poker chips
- Each toy excavator has the cards learning on their scoop
- The excavators tap the table -They all push their scoop forward revealing their cards
- Camera above rotates from the top and then it zooms in to the winning excavator, which spins in excitement and raises it arm
- It then starts moving into the pile of chips, climbing it up -It stretches its arm and claws into the pile of chips, pulling some and going back
- The other tractors can be seen shaking their upper bodies in disappointment

Project Scope

The scripts will be developed for Windows (possibly Mac and Linux). The scripts will be Python script files specific to Maya. There will be a demonstration component in the form of a VFX video that will use a textured excavator model destroying blocks. When information on weekly requirements are provided by our instruction we will update our scope to reflect our desire to meet all new expectations given.

The minimum user requirements include:

- Computer
- Windows(possibly Mac and Linux)
- Autodesk Maya 2017 or greater

Out-of-Scope

These scripts are not intended for the purpose of creating rigs in programs other than Maya. Although these scripts are intended for use in Maya, they are not concerned with creating the models they are

intended for or any associated asset to the model other than the control rig. Additionally, as the scripts focus on rigging an excavator, they have limited application, if any at all, to other forms. If used to rig an excavator there are requirements to how the geo must be separated and organized.

Further, there is limited information on what will be expected week to week by the courses instructors. Limitations therefore need to be added regarding all the additional content that becomes required by our instructors.

Equipment

Hardware

Each team member will be responsible for their own hardware. It is expected that each team member will use a personal computer capable of any task within the scope of this project. The following table details each team member's equipment.

Product	Description	Cost	Quantity	Total (CDN)
PC (64bit intel i9-10980XE, 2 x NVidia RTX 2080 SUPER, 64Gig DDR4 Ram)	Tylers previously owned working computer	\$0	1	\$0
PC (64bit Intel i5 - 8600k, Nvidia GTX 1080, 16gb ddr4)	Rodolfo's power station™	\$0	1	\$0

Laptop (i5-8300h, GTX 1060, 8Gig DDR4)	Windy's laptop	\$0	1	\$0
			TOTAL	\$0

Software

This project is committed to readability of the code. So each member of the team is required to have the same software. The following table is a breakdown of the software utilized for this project.

Product	Task	Cost	Quantity	Total
Autodesk Maya 2020(Student License)	Content Creation	\$0	3	\$0
Charcoal Editor(Free Version)	(IDE)Interactive design environment	\$0	3	\$0
Unreal Engine	Demo	\$0	3	\$0
Houdini	Destruction Effects	\$0	3	\$0
Google Docs	Project Management	\$0	3	\$0

Discord	Project Management	\$0	3	\$0
gitHub	Source Control	\$0	3	\$0
Substance	Shaders	\$0	3	\$0
			TOTAL	\$0

Evaluation

Target Platform (Maya)

Maya is the industry standard and more than capable of handling anything we throw at it. Maya is capable of python scripting which is what all the generated code will be. Maya is a common digital content creation software available to students and artists in 3D, who may benefit from these scripts.

Unreal Engine

Unreal is becoming increasingly popular among visual effects and film studios alike. The software provides real-time rendering, which facilitates efficiency. Unreal will be used for showcasing the fully functional rig. Unreal has many included assets that some of which will be used with the intention of making the demo aesthetically pleasing.

Scheduling

Development Plan

The Development plan is partly guided by the instructors at BCIT. We will complete the scripts together all at the beginning and work together for the video presentation. As time progresses we identify our strengths

and weaknesses to bring together the best deliverables. That said, Windy is amazing at Design for the UI, Icons, controls and more. We have decided to make Windy in charge of all artistic direction. Rodolfo and Tyler Have great ability in the software packages and will simply do what is required by the artistic direction. We Have made a gantt Chart that reflects the tasks we all feel are our strengths or are within a specific discipline we can overcome.

Milestones

(Date)	(Due dates/timeline)
January 9, 2021	Created the Project Planning Document
January 12, 2021	UI / Rigging / Scripting References Collected
January 14, 2021	UI Concept Designs
	Prepared 3D Model
January 15, 2021	5500 - Submission 1
January 19, 2021	Source Control
	Skeleton Setup
	Controls Setup
January 22, 2021	UI script

January 23, 2021	Template Code
January 23, 2021 (8am, group)	6700 - Presentation
January 22, 2021	Gantt Chart
January 22, 2021	Project Charter
January 24, 2021 (midnight)	6700 - Project Documentation
January 29, 2021	Door Script Complete
January 31, 2021 (midnight)	6700 - Project Update
February 5, 2021	Scripting Functions Complete
February 7, 2021 (midnight)	6700 - Project Update
February 12, 2021	UI Integration
February 14, 2021 (midnight)	6700 - Project Update
February 21, 2021 (midnight)	6700 - Project Update
February 26, 2021	Model/Material Integration into Unreal, Repository setup, Update planning document and gantt chart
February 27, 2021 (8am, group)	6700 - Presentation
March 5, 2021	Environment start, rig in unreal, video structure/cameras

March 7, 2021 (midnight)	6700 - Project Update
March 12, 2021	Environment, functional rig in unreal, video structure/cameras
March 14, 2021 (midnight)	6700 - Project Update
March 19, 2021	Finnish unfinished business
March 21, 2021 (midnight)	6700 - Project Update
March 26, 2021	Relaxation and play
March 28, 2021 (midnight)	6700 - Project Update
April 2, 2021	Fix Bugs, Simulation and rendering, final camera work
April 4, 2021 (midnight)	6700 - Project Update
April 9, 2021	Final Project Submitted
April 10, 2021 (8am, group)	6700 - Presentation

Updates and Maintenance

(Month/Year)	(Name the specific object in the game)	(Explain if your adding new specific object or updating the specific object)
---------------------	---	---

TBD	TBD	TBD

Work Environment

Our team is international and no two team members are in the same country. The team will work in Pacific Standard Time (PST) (GMT-8). Tyler is in Burnaby, BC, Canada (PST), Windy is in China (GMT+8), and Rodolfo is in Mexico (GMT-6).

We utilize several platforms to remain connected and operate efficiently. Primarily, we communicate through the Discord platform but meet over Zoom on a weekly basis. We also rely on Google Docs to update and share documentation as needed. For Source Control, we will use bitBucket.

File Formats & Naming Conventions

Asset Type	Subtype	Naming Convention	File Format	Annotations
3D asset	Filename	fileName	.ma	suffixes material = _mat materialSG = _matSG groups = _grp geometry = _geo shapes = _shp joints = _jnt controls = _ctr textures = _tex orientConstraint = _ortCt parentConstraint = _prtCt

				Effector = _efr ikHandle = _ikH
Animations /exports	Filename	fileName	.fbx	
Textures	N/A	fileName	.tga	
Scripts	Filename	fileName	.py	
Materials	N/A	fileName	.txt	
UI	Filename	fileName	.py	
UI icons	N/A	fileName	.png	
File Folders	N/A	Filename		

Art

Setting

The theme of the logos and UI will be Maya. This means that we intend to keep the UI looking as if it could have been shipped with Maya. We do not intend on altering the aesthetic from the Maya default.

Minimum Viable Product

The bare minimum requirements are that the user can install the project into their Maya. The User can then use the core functionality (root control, COG control, arm controls, auxiliary controls, and track controls). We as developers are able to utilise the functionality to create a rig for our excavator. The excavator can be exported to unreal with animations. A video demonstration of the excavator demolishing blocks is produced.

UI

<https://www.youtube.com/channel/UCHi1YBVSnDFGsXrvVVhArOQ>