

Project Plan

An Easy 3D Modeling Tool

Tai, Tsen-jung 3035146669

Supervisor: Professor Francis C.M. Lau

October 1st, 2017

Contents

1	Introduction	2
1.1	Background	2
1.2	Problem	2
2	Related Works	2
2.1	TinkerCAD	2
2.2	Blender	3
3	Objective	3
4	Methodology	5
4.1	Front end	5
4.2	Back end	5
4.3	3D modeling software	5
4.3.1	Bézier curves & Bézier surface	5
4.3.2	Non-Uniform Rational Basis Spline curves (NURBS) & surface	6
4.3.3	Digital sculpting	6
5	Deliverable	7
6	Schedule	7
7	Conclusion	9
	Reference	10

1 Introduction

1.1 Background

3D printing technology has been emerging and popularizing over the past decades, which has led to a huge demand on powerful 3D modeling software. Although there is a batch of traditional software that are able to handle creation and rendering of complex 3D objects, they often have scaring user interfaces containing tons of tools and options and require steep learning curves. And since traditional software need installation, there is a lack of portability and a hinder for sharing. All these result to the rising of a call for light-weight web-based 3D modeling software, especially in the field of school children education and personal usage.

1.2 Problem

There are various online software providing 3D modeling functionalities with different levels of quality in performance. However, most of them lack the capability of forming curved surface and achieving advance deformation, and there is a defect in the algorithm behind the grouping and punching operation, which might lead to a displacement among the objects grouped and a long-time processing. These weaknesses greatly impair the operability and practicality of the web-based modeling tool. Detailed comparisons among currently existed traditional software, web-based software and our project product are made in the second part - related works of this report.

2 Related works

2.1 TinkerCAD

TinkerCAD is an easy-to-master web-based 3D design application belonging to Autodesk, Inc. The software offers shapes, characters and symbols as basic building components, which the user can pile or arrange to build complicated objects. Basic modeling functions such as scaling, rotating and grouping are also provided to adjust each building block. However, curved surface can only be approximately constructed by exploiting the surface of spheres, cylinders, paraboloids and those blocks containing

arched surface. And a fine relative position amongst objects might cause error after grouping, for instance, linking two sharp ends to form a funnel.

2.2 Blender

Blender is a robust open-source 3D creation toolset written in mainly C and C++. Besides the basic building blocks and transformation operations provided in TinkerCAD, it provides tools to create Bézier curves and Non-Uniform Rational Basis Spline curves (NURBS) and tools to extend those paths to corresponding curved surface or blocks in 3D space. It also contains a group of modifiers which apply deformation effects to objects without destruct the original shapes, such as distortion or subdivision surface that requires overwhelming work if manually editing. Another useful feature offered is 3D sculpting, which allows pushing, pulling, pinching and similar real-life behaviors that can be performed in clay. Other powerful add-ons such as particle system are out of the scope of our project and are not discussed in this report.

Although Blender is of high practicality and performance, it requires installation and holds an unfriendly user interface due to massive modeling options. The former reduces the portability of the production process and the latter leads to a steep learning curve.

3 Objective

This project is a further development based on a prototype developed by Dr. Chan Bin, and a current functioning version can be found here. The prototype contains the building blocks of basic shapes and modeling operations provided in TinkerCAD. Additionally, it provides a curve editing window for creating curved surface (Figure. 1). However, this editing option only applies to a few specific building components and no free-form curved blocks are allowed. Besides, errors of wrong displacements occur when there is a change in the viewpoint or a grouping of objects (Figure. 2).

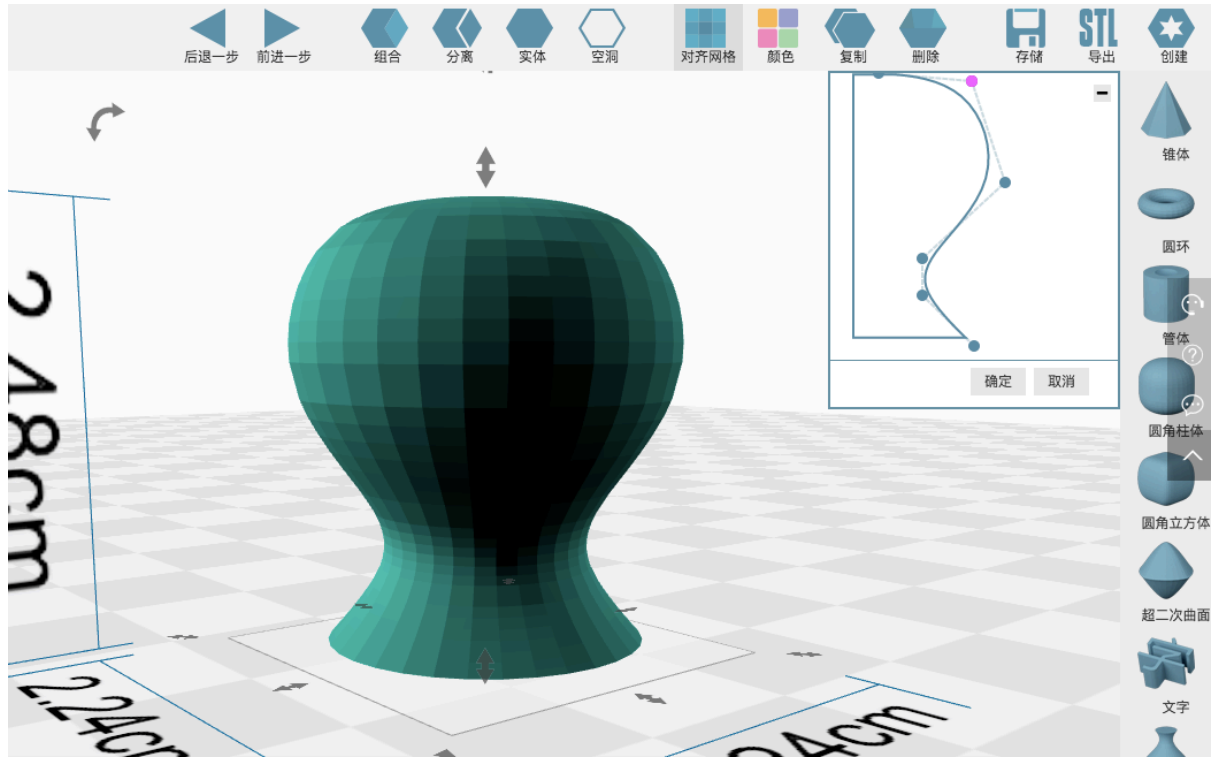


Figure. 1 The 3D modeling prototype with a curve editing window on the top-right corner. The screenshot is obtained from the workspace of azura3d.com.

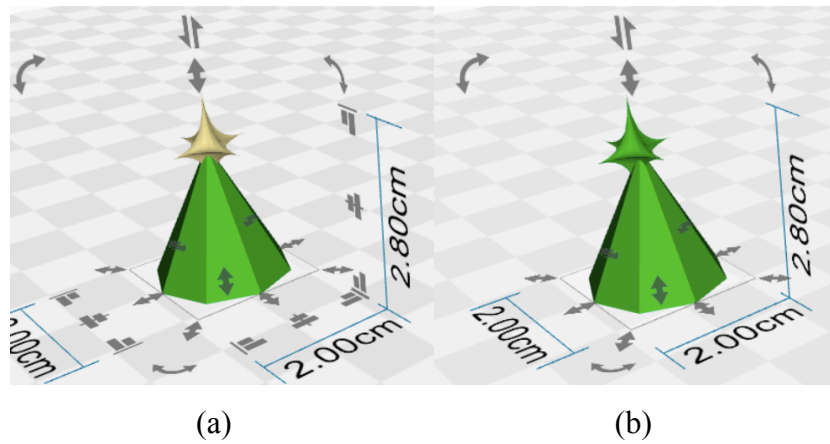


Figure. 2 Two images showing the relative position between a star and a pyramid before and after the group operation. (a) Before grouped, the star shares about the same vertical axis with the pyramid. (b) After grouped, an obvious dislocation occurs to the star object.

The screenshot is obtained from the workspace of azura3d.com.

The goal of the project is to fix the existing bugs during the viewport transition and object grouping, decomposing and aligning, then add advanced 3D curving operation by adding canvas tools such as a pen and an path/surface extension handler. Further improvements include implementing a brush tool to perform digital sculpting and researching to improve the speed and accuracy of the transformation and rendering algorithms.

4 Methodology

This project contains both front-end and back-end work, as well as the 3D modeling software to be embedded in the website. Most of the effort lies in the development of the software, and then the back-end data storage technique.

4.1 Front end

The front-end code construct the web layout to display the embedded 3D modeling software as well as the development log, written in mainly HTML, CSS and JavaScript. It also implements a simple login system to preserve work for each corresponding user. This system does not include password encryption and session cookie check for simplicity, since these are not the focus of this project.

4.2 Back end

The back-end side, written in primarily PHP and MySQL, maintains the database of user accounts, as well as their corresponding modeling work. It implements the real-time data storage technique to avoid loss of user work due to a browser breakdown or other failures.

4.3 3D modeling software

The software is written mostly in C, relying heavily on Open Graphics Library (OpenGL), which is the application programming interface (API) most widely adopted in professional and industrial CG application development. WebGL, which is based on OpenGL and targets at web browser Graphics processing, provides a JavaScript API for use in the HTML5 <canvas> element.

4.3.1 Bézier curves & Bézier surface

A Bézier curve is a free-form curve extensively used in Computer Graphics, an n degree Bézier curve is generally denoted as [1]:

$$r(t) = \sum_{i=0}^n b_i B_{i,n}(t), \quad 0 \leq t \leq 1.$$

Its curvature can be adjusted by the $(n+1)$ control points b_i and the Bernstein basis polynomials [2] $B_{i,n}$. In programming practice, De Casteljau's algorithm can be used to generate Bézier curves using recursion [3]. And curved surface can be formed simply by manually dragging a Bézier curve towards one direction across the space. An advanced alternative is utilizing Bézier surface [3]. As the property held by Bézier curves, the surface can be well tuned intuitively by moving the control points.

4.3.2 Non-Uniform Rational Basis Spline curves (NURBS) & surface

NURBS curves offer higher flexibility and precision during modeling. While a Bézier curve tends to possess a convex hull property, where the curve flows along the contour of the control polygon consisting of the control points, a NURBS curve extends according to the relative position of the closet control points set (Figure. 3). Consequently, by adjusting single control point in a Bézier curve and a NURBS curve respectively, the whole curve of the former goes through some changes while only local effects are applied to the nearest two curve segments in the latter [4].

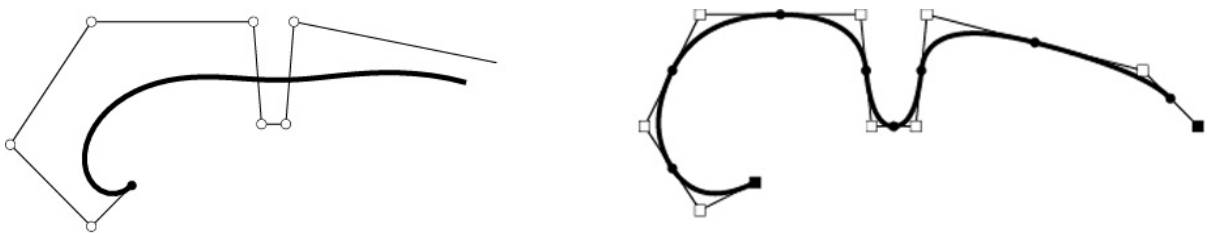


Figure. 3 Two mathematical generations of curves using similar control polygons. (a) A Bézier curve. (b) A NURBS curve. Source retrieved from Courseware.DeadCoderSociety.org [4].

4.3.3 Digital sculpting

3D Sculpting is an advanced modeling technique allows the users to perform manipulation of high degree of freedom on an object as if doing sculpting. A method to achieve this technique is to exploit the subdivision solid geometry [5].

An web-based application implementing sculpting can be found here, written in WebGL and JavaScript.

5 Deliverable

The final product is a 3D modeling software embedde in the corresponding final year project website: <http://i.cs.hku.hk/fyp/2017/fyp17038/>. Development log and progress can be checked on the website as well.

6 Schedule

The project is divided into mainly two phases.

Phase 1 mainly focuses on modifying the current exist features of the prototype, such as fixing the bugs of displacement, upgrading the flexibility of the curved surface modeling and ensuring the web accessibility and manipulability.

Phase 2 is of higher difficulty. There are two directions can go and since both requires heavy effort on research, only one can be guaranteed to be achieved taking time constrain into consideration. Due to current lack of practice of building modeling engine, the final decision on which direction to go is left to be decided by the end of Phase 1. Phase 2.1 and 2.2 explain the two directions.

Phase 2.1 Add digital sculpting operation.

Phase 2.2 Modify the matrix and boolean operation behind transformation and rendering algorithms to speed up the calculation process.

Task	Start Date	End Date	Duration	Milestone
3D Graphics mathematics & Research paper study	Sep 1 st	Sep 30 th	30 days	

Prototype installation & environment setup	Oct 1 st	Oct 5 th	5 days	
Understanding source code of the prototype	Oct 6 th	Oct 22 nd	17 days	M1. Able to modify the code
Front & Back end setup with login system	Oct 23 rd	Oct 27 th	5 days	
Embedding the prototype to the website and realize real-time storage of current work	Oct 28 th	Nov 10 th	14 days	M2. Realization of web-based feature
Fixing displacement bugs due to the viewport changes	Nov 11 th	Nov 30 th	20 days	
Fixing displacement bugs due to the grouping operation	Dec 1 st	Dec 17 th	17 days	M3. Fix existing bugs of the prototype
Implementation of curved path	Dec 18 th	Dec 31 st	14 days	
Implementation of curved surface & blocks	Jan 1 st	Jan 7 th	7 days	M4. Implemente Curving operation
1 st presentation	Jan 8 th			
Sculpting OR transformation algorithm research*	Jan 13 th	Feb 1 st	20 days	M5A. Decide and research on the advanced part of the project
Implementation of Sculpting OR transformation algorithm*	Feb 2 nd	Mar 27 th	55 days	M5B. Implement the advanced part of the project
Testing, debugging and documentation	Mar 28 th	April 15 th	18 days	M6. Finalization of tested implementation
Final presentation	April 16 th			

* Since both the implementation of digital sculpting and the modification of deformation algorithm require high-level understanding of the current project and OpenGL skills, and both takes large-scale of time, only one of them is to be implemented. And the decision is to be made after gaining the experience during the development of Phase 1.

7 Conclusion

This project aims at providing a user-friendly web-based 3D modeling toolset. Basic building blocks of shapes such as spheres, cubes, cones and so on are offered. And the users can perform basic operations on the objects to transform each building blocks. The set of transformation operations include but are not limited to scaling, rotating, grouping, discomposing and punching a hole. Advanced 3D creation is available by the presence of curve tools. The users are free to change their perspectives to obtain a panoramic vision. A further improved feature can be: (a) Sculpting tools which allow the users to perform fine deformation on objects, such as pulling, pushing, twisting, pinching, etc. (b) A modification on transformation and rendering algorithm which speeds up the calculating procedure.

This project is a further development based on a previous web-based 3D modeling prototype developed by Dr. Chan Bin.

Reference

- [1] N. M. Patrikalakis, T. Maekawa, W. Cho. (2009, December). *Shape Interrogation for Computer Aided Design and Manufacturing* [Hyperbook]. Available: <http://web.mit.edu/hyperbook/Patrikalakis-Maekawa-Cho/>. [Accessed: 2017, Sep 17]
- [2] P.P. Korovkin. (n.d.). *Bernstein polynomials* [Encyclopedia of Mathematics]. Available: http://www.encyclopediaofmath.org/index.php?title=Bernstein_polynomials&oldid=13289. [Accessed: 2017, Sep 17]
- [3] G. Farin, *Curves and Surfaces for CAGD*, 5th ed. San Francisco, CA: Academic Press, 2001.
- [4] Bezier Curves, B-Splines & NURBS. (n.d.). Courseware.DeadCoderSociety.org [Online]. Available: http://courseware.deadcodersociety.org/csis4466-computer_graphics/Graphics/10.%20Bezier%20Curves,%20B-Splines%20&%20NURBS.pdf. [Accessed: 2017, Sep 17]
- [5] K. T. McDonnell, H. Qin and R. A. Wlodarczyk. “Virtual Clay: A Real-time Sculpting System with Haptic Toolkits”.