METU, Department Of Computer Engineering

Graduation Project
Proposal Form

Important Notes

A project could be proposed by (i) a student group, (ii) a company, or (iii) a faculty member of the department by filling in this form and submitting it to 49x-proposal@ceng.metu.edu.tr by e-mail. For a project proposal, there might be a sponsoring company supporting the project and providing some form(s) of resources for the project.

If your proposal might contain a patentable idea or any type of intellectual property, please first make sure to follow appropriate steps (apply for a patent, etc.) before sending your idea to us. Once this form is received from you, the instructor(s) and the department has no responsibility regarding to intellectual properties of your project/idea.

All sources and documentation developed for this course are assumed to be public domain (GPL, CC or similar license) by default. If you need any exception for license and disclosure of project work, please specify this in detail in IP section of the form.

Please note that source codes, documents and issue tracking should be kept in department servers. No restrictions can be requested for limiting faculty and assistants access to student work.

Project Information

Title

Blender based digital geometry processing tool

Target

Public [ ] Restricted [X]

Only group-3 is eligible for this project.

Proposer Information

Name(s)	*Emre Baris Toyan* *Furkan Odluyurt* *Dicle Ayzit* *Uğur Yanıkoğlu*
E-Mail(s)	*e1819598@ceng.metu.edu.tr* e1819499@ceng.metu.edu.tr e1819051@ceng.metu.edu.tr e1881598@ceng.metu.edu.tr

IP (Intellectual Property) Information

Blender software is released under GPL; since it is build upon Blender, our project will be an open-source project under GPL as well. Asst. Prof. Yusuf Sahillioglu from METU Comp. Eng. Dept. is the supervisor of the project.

Project Description and Background Information

Description

List of abbreviations used in this document.

DGP Digital geometry processing
CG Computer graphics
GUI Graphical user interface
API Application programming interface
UX End-user experience
WYIIWYG What you implement is what you get (our own catchphrase)

Three-dimensional geometric models are the base data for applications in computer graphics, computer aided design, visualization, multimedia, and other related fields. Operations on those geometric models has always been the hot topic of academic research, such as; decimation, subdivision, regularization or sampling, deformation, (adaptive or not) mesh construction...

The project is a 3D graphics software, as well as an integrated development environment in mind, which is specially crafted for developing and testing those operations, or namely, DGP algorithms.

It will be build upon open-sourced Blender and will adapt WYIIWYG as in PixelBender Toolkit by Adobe.

Similar Products/Projects

Bioblender is a project run by a few researchers at SciVis, CNR, located in Italy. The concept in Bioblender is the scientific visualization of the molecules etc. and it is equipped by some famous algorithms in clinical physiology. Although the concept of our project has nothing to do with molecules, Bioblender is an example of how we build upon Blender. Same technique, different concept.

Justification of the proposal

There is no easy-to-use framework *for DGP specialists. 3D software have always been large-scale software including many modules integrated at once: modeling, rendering, animation, composition etc. Those software are*

1. ***mostly commercial*** *since companies put years of effort*
2. ***general-purpose*** *(i.e. not DGP specific) since many modules cooperate together*
3. ***hard-to-grasp*** *by DGP specialists since SDK for those large-scale apps come with complex API*

That way, we also hope to encourage not only DGP specialists but also anyone with basic programming background to focus on digital geometry processing. Providing an easy-to-use interactive user interfaces will help people not to get discouraged by difficulties of hard-to-grasp frameworks.

> *We also consulted students of* ***CENG789*** *where famous DGP algorithms are introduced. The environment in which students are encouraged to work (Open-Inverter, another hard-to-grasp framework) is not as easy as it seems to be; students suffer from integrating complex data-structures defined in open-inverter's framework to their own implementation homeworks. (We can not share a code sample here, for limitedly reasons. But you can always see implementation of vertex color change after vertex selection over GUI in Open-Inventer. For example in our tool, we plan to achieve this by a* ***single line of code; TOOL.APPLICATION.enable_vertex_selection*** *) Getting accustomed to a new framework is an indecency; because course's goal is only to introduce DGP algorithms*

What we have in mind at the end, is a 3D software with an integrated development environment console (thanks to Blender), helpful GUI to support WYIIWYG, and incredibly user-friendly API. Other than those application specific features, we will equip our framework with ***implementations of state-of-the-art DGP methods defined in actual papers****. (see. supervisor requested implementations in 'Technical Aspects' section.) And those are what a DGP specialist needs to develop | test | see immediate results of her algorithm.*

Contributions, Innovation and Originality Aspects of the Project

The project is innovative and original; because there is no any other ***user-friendly*** *API | framework | application that can be used to implement famous DGP algorithms and develop | test new ones. Also, we offer better solutions to the specialists that 3rd party frameworks have never provided. One example is we come up with a catchphrase* ***wyiiwyg*** *(what you implement is what you get), because we plan to visualize editable algorithm parameters through GUI of the project. That way, implementor will have a chance to try-and-see how his/her implementation works.*

Advantages:

Most frameworks for DGP are written in VC++, just like Open-Inventer. Therefore specialists are obliged to work in Windows only. However, Blender is cross-platform and our product will be cross-platform as well.

This project is open-source and is targeted to academic people. Moreover, mesh recognition and mesh retrieval are two academic hot topics in CG. As it gets mature further, we may support for those topics in our project.

Technical Aspects of the Project

When implementing our tool, we will use Blender-Python-Interface (BPY). Therefore our implementations are plugin-based, unless it is required to alter the C/C++ source of Blender. We can specify our service in terms of two categories. One set of API will be implemented for UX, the other set of algorithms will be implemented to truput comparison. That way, end-user makes sure his/her implementation is correct. Below is an rough idea how our API look like. We will need to ***implement a few papers*** *for supervisor_requested API.*

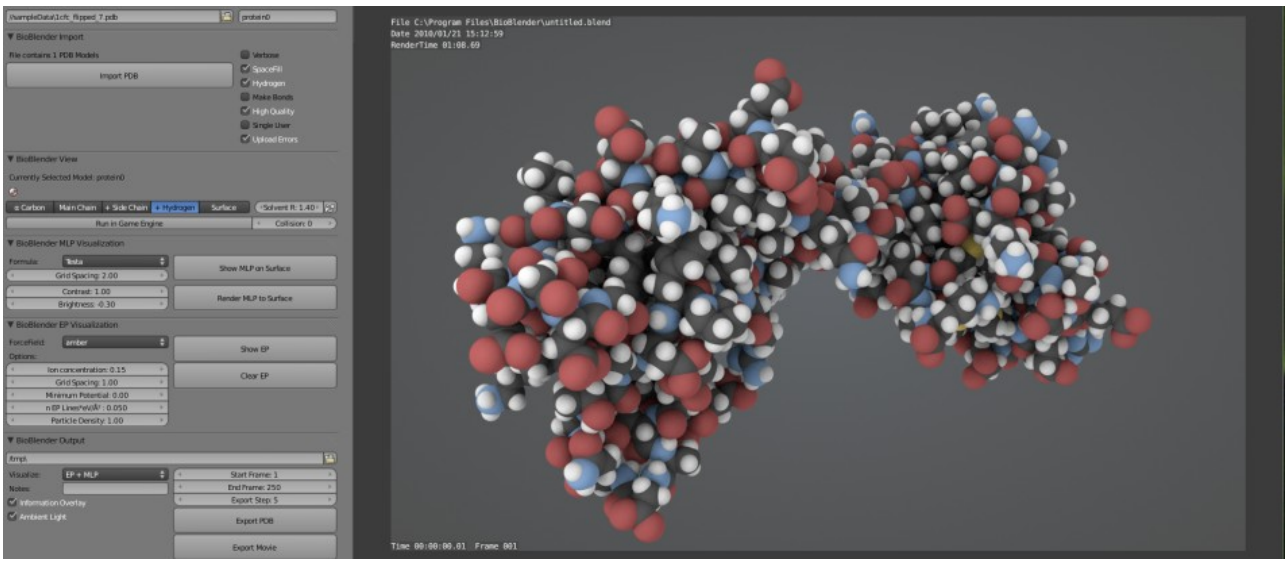
[EndUser] → [Our tool] → [Blender]

```
TOOL.APPLICATION.enable_camera_trackball                    //BPY implementation
TOOL.APPLICATION.enable_vertex_selection                    //BPY implementation
TOOL.APPLICATION.process_algorithm_just_after_run           //Another BPY implementation
TOOL.APPLICATION.enable_interactive_brush
…
TOOL.DGP_READY_TO_USE_ALGORITMS.smooth                      //Own implementation 4 truput comp.
TOOL.DGP_READY_TO_USE_ALGORITMS.move                        //Own implementation 4 truput comp.
…
TOOL.SUPERVISOR_REQUESTED.DESCRIPTORS.average_geodesic_distance
TOOL.SUPERVISOR_REQUESTED.DESCRIPTORS.global_point_signatures    [rustamov et al.]
TOOL.SUPERVISOR_REQUESTED.DESCRIPTORS.curvature
TOOL.SUPERVISOR_REQUESTED.SAMPLING.farthest_point
TOOL.SUPERVISOR_REQUESTED.SAMPLING.uniform
TOOL.SUPERVISOR_REQUESTED.DISTANCE.geodesic
TOOL. SUPERVISOR_REQUESTED.DISTANCE.diffusion
TOOL.SUPERVISOR_REQUESTED.DISTANCE.biharmonic                [rustamov et al.]
TOOL.SUPERVISOR_REQUESTED.VOXELIZATION.winding_number_method
TOOL.SUPERVISOR_REQUESTED.SPACE_DIVISION.voronoi_3d
```

Targeted Output, Targeted User/Domain Profile

This is a snapshot for Bioblender. (see. Similar Projects) This image ***closely looks like*** *what our product's GUI will look like at the end. If we need to specify further, we have*

1. one canvas (Interactive 3D GUI where end-user works in by rotating | zooming in/out mesh or meshes)
2. application panel, (where application parameters reside)
3. user panel, (where custom algorithm parameters reside)
4. and an integrated development console.
See SRS for further detail.



A snapshot from BioBlender

Our product mainly targets academic people who works on DGP. (DGP specialists are usually academic people.)

Project Development Environment

Pyhton3x, maybe C/C++
Cross-platform: Windows, Ubuntu, MacOS

External Support

We do not require any support from the department to develop our project. We do not plan to utilize external support for the project.

References

A similar solution for our project is Bioblender
http://www.bioblender.eu/

mzoppe@Ifc.cnr.it
Scientific Visualization Unit – Istituto di Fisiologia Clinica
Area della Ricerca San Cataldo
Via Moruzzi, 1
56124 Pisa (Italy)
Tel: (+39) 050 3153095

http://www.adobe.com/devnet/archive/pixelbender.html	//PixelBender by Adobe
http://ocw.nagoya-u.jp/files/34/trackball.c	//A trackball implementation by Silicon GFX, single line of call in our environment