Computer Graphics Project: Sketch-Based Animation System

Team 14

HARSH VARDHAN SINGH and PRANAY GUPTA

1 INTRODUCTION

In the computer graphics domain, we present our project, "Sketching Dynamic and Interactive Illustrations," a fusion of artistic expression and digital innovation. This endeavour seeks to recreate a platform such as "Kitty", which provides a robust platform for individuals, from artists to designers, to create, animate, and interact with their visual narratives.

Our project will build upon the accomplishments of platforms like "Kitty" to take dynamic illustrations to a new level. Rather than focusing solely on static imagery, our goal will be to introduce interactive features that expand the creative possibilities.

Within our digital canvas, we will offer a range of tools to support the sketching process. These tools will include various brushes, erasers of different sizes, colour-filling options, colour pickers, magnification tools, a diverse colour palette, and the ability to integrate images seamlessly.

Our project will go beyond traditional sketch illustration to explore kinetic textures, which will be dynamic particle systems influenced by prior works like "Draco." We will also introduce interactive graph structures enabling users to connect events, eliminating the need for manually synchronizing animations on a global timeline. These interactive graphs will infuse vitality and motion into our creative canvas.

2 LITERATURE REVIEW

The sketch-based animation and dynamic illustrations field has seen significant advancements in recent years, as evidenced by several noteworthy contributions. While we primarily focus on recreating a platform that closely resembles "Kitty" using OpenGL, we recognize the broader landscape of research that informs our work. This literature review provides a snapshot of critical results and their contributions.

"Kitty" by Kazi et al. [1] is our project's cornerstone. It introduces an interactive system for sketching dynamic illustrations, showcasing the power of free-form ink-based drawings in an interactive digital environment. This work inspires our efforts to allow users to draw and animate their visual narratives.

"Kitty" offers diverse interactive tools, including a unique graph structure that connects visual entities and parameters, providing an intuitive way to create interactivity. The system employs force-directed layout algorithms to ensure minimal distances between nodes, simplifying the placement of visual elements. It utilizes a colour-coded node system to distinguish between elements, such as objects, emitting textures, and oscillating textures. This approach enables a user-friendly, interactive graph-based creation process.

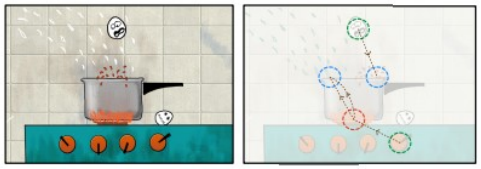

Fig. 1. Visualization of the colour-coded node system to distinguish between different types of elements: objects (green nodes), emitting textures (blue) and oscillating textures (orange)

Authors' address: Harsh Vardhan Singh, harsh20202@iiitd.ac.in; Pranay Gupta, pranay20228@iiitd.ac.in.

Additionally, "Animation Sketching: An Approach to Accessible Animation" by Moscovich [3] is an essential source of inspiration for the "Kitty" project. This work explores motion recording techniques and simplifies the animation process, providing valuable insights into accessible animation for communication and planning.

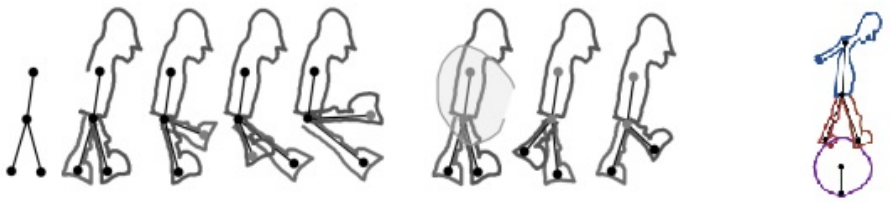

Fig. 2. Visualization of free-form skeletal control

"Draco" [2], a non-interactive but influential system, explores kinetic textures, which can add dynamic elements to static illustrations. While not the central focus of our project, it presents valuable insights into the dynamics of visual elements.

"Draco" introduces the concept of kinetic textures, allowing for incorporating elements like particles and fluid dynamics into static illustrations. This non-interactive system emphasizes adding dynamic features to visual composition, providing a foundation for exploring dynamic textures and animations.

In our pursuit of interactivity, we aim to enhance user experiences with features such as various brushes, drag-and-drop mechanisms, and masking, enabling the creation of multiple layers.

Our project seeks to combine these inspirations and concepts to create a platform that empowers users to seamlessly sketch, animate, and interact with dynamic illustrations.

3 MILESTONES

S. No.	Milestone	Member
	Mid evaluation	
1	Sketching and conversion to discrete objects (brush strokes to entity definition)	Harsh
2	Layered object manipulation (Drag, Drop and Delete for existing entities)	Pranay
3	Rigid Body Animation with local timeline implementation	Harsh
4	Defining and Implementing Kinetic Textures (Emitting, Oscillating and Granular)	Pranay
	Final evaluation	
5	Relational Graph Structure Implementation (Initial setup and parameter definition)	Harsh
6	Relational Graph Structure Implementation (Shortest Paths and Transitive Relations)	Pranay
7	Time Control Presets for the animation timeline	Harsh
8	Chain Effects Implementation	Pranay
9	Importing existing images *[Tentative]*	
10	Export Logic *[Tentative]*	

4 APPROACH

Let us now discuss the main methodologies.

- Implementing sketches: We create an algorithm where a person can create sketches, continuously adding points on the canvas. It stores the points in a temporary list, passed to a main list, which stores all of these temporary lists. The temporary list adds points once the User presses the mouse, passes these points to the final list, and clears itself once the User releases the mouse. To avoid adding new duplicate points when the User is not dragging their mouse but keeping their mouse pressed in one place, we check if the mouse is stationary and, accordingly, don't add points. We also devise an algorithm to view these sketches when they are not being edited
- List of illustrations: We create a simple manager which allows us to add, delete and edit a selected drawing.
- Editing sketches: We can move, rotate and scale any chosen images using a 2D homogeneous transformation matrix.
- Animating sketches: Each illustration has a manager that manages its animations. We interpolate between frames to animate drawings utilizing a timer and a 2D homogeneous transformation matrix. We can keep a list of spirits for the same object. We can play/pause/reset these animations by selecting one at a time.
- Kinetic textures (Emitting textures): This has four sub-parts: emitting surface, Emission direction, Emission particle, and Mask. The emitting surface is where the Emission particle will emit in the specified emission direction. The particles will only be visible when they are in the masked area. A manager allows us to have multiple kinetic textures in the same scene.
- Relational Graph Structure Implementation: The graph comprises nodes and edges. The nodes represent the sketches (static sketches, kinetic textures, etc, where a unique node can represent each type). A node represents each sketch. So, for N total sketches, we will have N nodes (nodes are instantiated as we instantiate sketches). Considering that the edges do not have any directions, we can have a maximum of N*(N-1)/2 edges. The User can add these edges, and the User can specify the initial and end nodes and the kind of relation between these nodes (lines/square/cubic/exponential). Each node can be connected to the rest using these edges
- Time control presets. Here, the animation of one node can be controlled using the animation of another node, given that the appropriate edge has been created. The User can specify the relationship between the run times of these animations. It has been constrained to a linear relation but can also be extended to the square/cubic/exponential relations.
- Chain effects. Once a proper relational graph has been set up, we can extend the functionality to have chain reactions where one node catalyzes the other node, which further catalyzes another node. This can be achieved using recursion (not implemented).

5 CHALLENGES FACED

We faced a few challenges while coding ...

- While colouring the sketches, we kept a single global colour for all of our sketches. Which might not be desirable. However, it was not hindering without algorithms, so we reserved changing this for a later stage and focused on other algorithms.
- Applying recursion to the graph turned out to be challenging, and also applying these graphs to multiple animations (Each object could have multiple animations; the current graph takes into consideration the first animation of each sketch only)
- We did not use quaternions to rotate objects, which would have been better than primitive rotations techniques, which sometimes give us weird artefacts.

- Our implementation is not as interactive as we envisioned, where most of our manipulations are being controlled using a toolbox rather than direct manipulations, a design choice we had made earlier.

6 RESULTS

- We achieved a simple program that allows us to interact with different sketches and combine them to create relations. These relations can be worked upon extensively to bring more variations and introduce efficient methods.

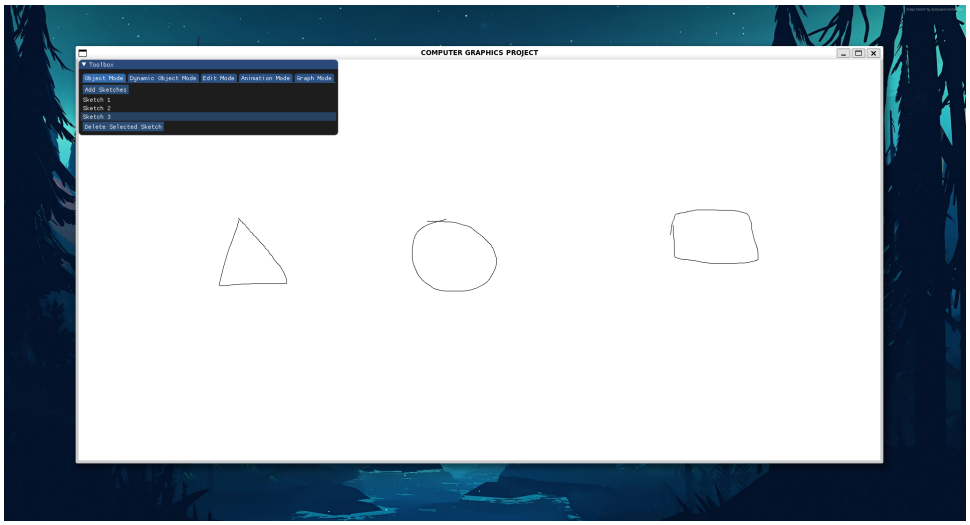

Fig. 3. User can add hand-drawn sketches with each as a separable object, which can be edited or deleted at any stage of the program run

7 CONCLUSION

To merge artistry with innovation, we developed a dynamic illustration platform inspired by "Kitty" and pivotal research in sketch-based animation. Our approach spanned sketch creation algorithms, animation management, kinetic textures, and relational graph structures, culminating in a versatile canvas for users to sketch, animate, and interact seamlessly. Challenges surfaced but served as learning opportunities, guiding our focus. The system's results showcased seamless sketch manipulation, interactive connections, and orchestrated animations, demonstrating the potential of our platform. This project marks the convergence of art and technology, offering a glimpse into the expansive possibilities of dynamic illustrations while hinting at ample room for future enhancements.

REFERENCES

[1] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 395–405.

[2] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014. Draco: bringing life to illustrations with kinetic textures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 351–360.

[3] Tomer Moscovich and John F Hughes. 2001. Animation sketching: An approach to accessible animation. *Unpublished Master's Thesis, CS Department, Brown University* 5 (2001).

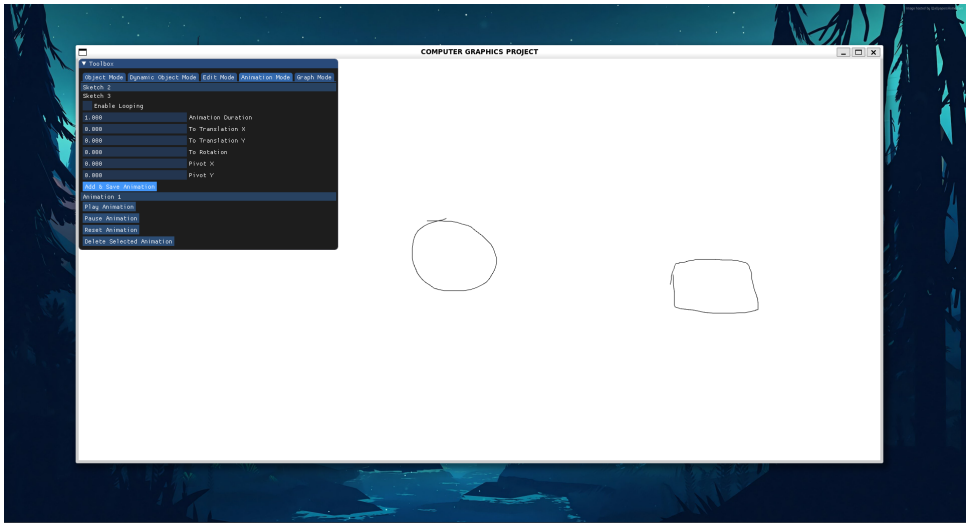

Fig. 4. User removed one of the sketches on the left and edited the position of one of the sketches on the right

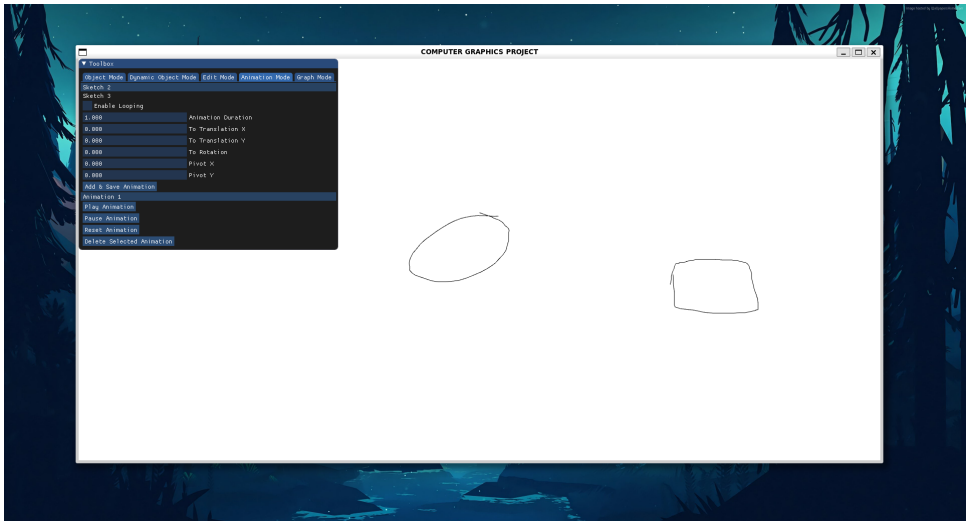

Fig. 5. User animated the figure in the centre

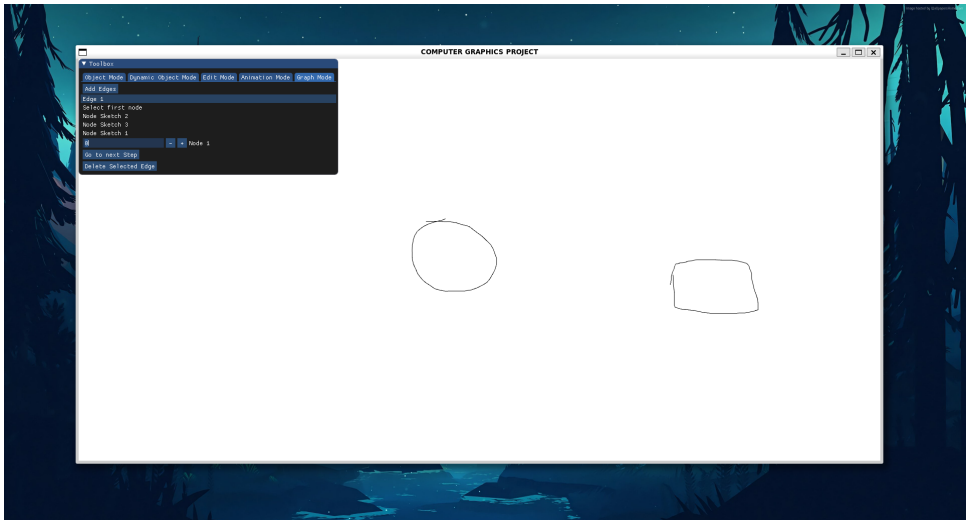

Fig. 6. User is creating an edge between the two sketches nodes

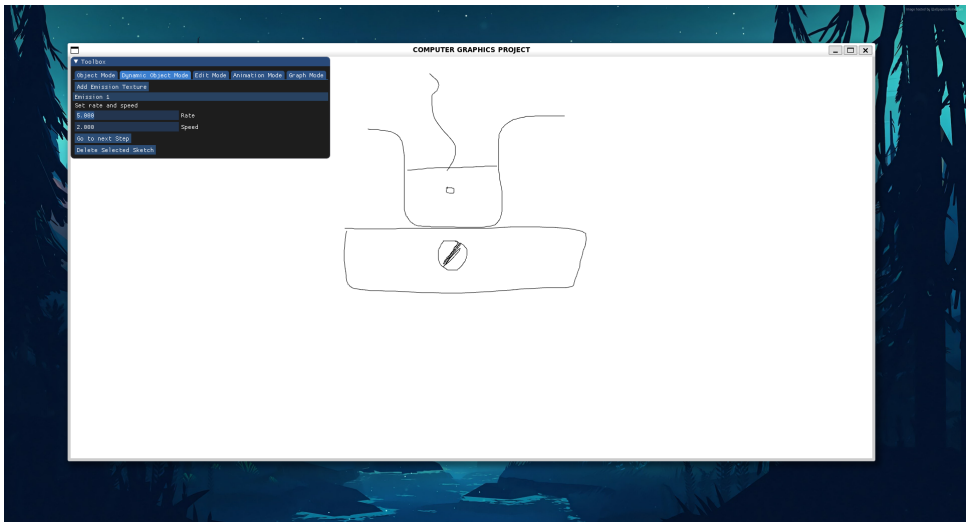

Fig. 7. User creates a stove and an emitting texture which replicates boiling water

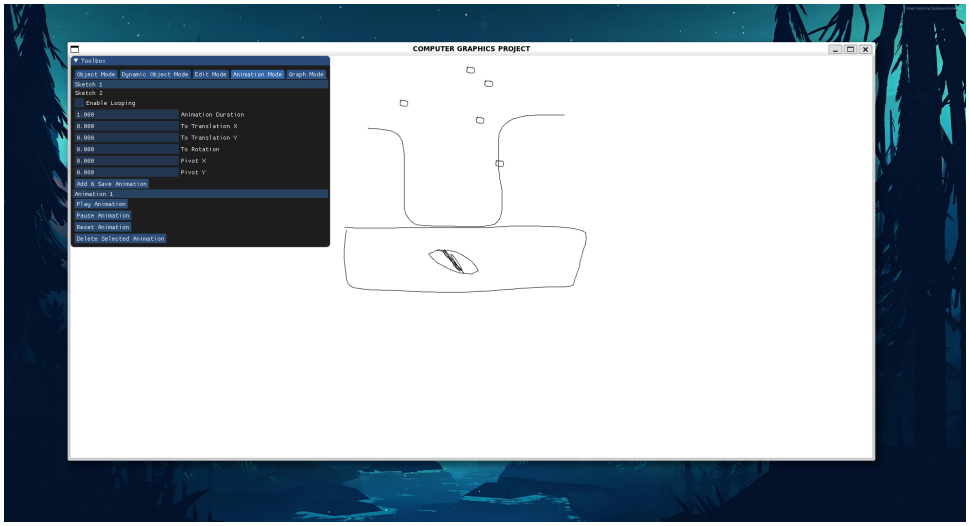

Fig. 8. User uses all the above methods to create an interactive scene which boils the water when user players the gas stove animation