1 Getting Started

The ProjectFuji application is similar to common game engines or rendering applications. When the application is first opened, users are presented with a screen that should look like Figure 2.1. In its default state, the main camera is a freeroam camera, meaning that users can move it by using the W, S, A, D keys where W moves the camera forward, S backward, A to the left and D to the right. Additionally, keys Q and E can be used to move it downward or upward, respectively. Mouse is then used for camera rotation. **However, the mouse is not locked by default! Users can lock and unlock the mouse by pressing the C button or by clicking the "Consume Mouse" button in the UI view tab. Furthermore, the mouse wheel can be used to control the camera movement speed.** To switch to orthographic camera, users can use the same set of keyboard shortcuts as in Blender, where numpad key 5 switches the projection from perspective to orthographic and back, while numpad keys 1, 3, 7 (or 9) change the view to front view, side view and top view, respectively. Lastly, when orbit cameras are used, Q and E keys are used for orbiting around the scene. Majority of the application's shortcuts should be shown in parentheses in the UI. Only exception is the usage of + and - numpad buttons to increase or decrease particle count by 10k. Additionally, if Ctrl key is held, + and - buttons increase/decrease the particle opacity.

Brush Mode

Press B on the keyboard to switch to/from the emitter brush mode. **Beware that particles below CCL (i.e. particles that should not be visible, which will be most, if not all particles on the ground) are hidden by default. To make them visible, press Ctrl + B or change their settings in the cloud visualization tab under "Show Particles Below CCL".** When in brush mode, users can scale the active brush by using the mouse wheel. Furthermore, if Shift is held when scrolling the mouse wheel, number of emitted particles (i.e. brush opacity) is changed. If Ctrl is held, the range of profile indices (convective temperatures of emitted particles) is moved. Lastly, if Alt is held, the size of the profile index range is increased (mouse wheel up) or decreased (mouse wheel down). As an example, let us say that the user has currently a range of [20, 50]. By scrolling with Ctrl, he can move the range to [40, 70] or [10, 40] for example. By scrolling with Alt, he can change the size of the range to [30, 40] or [35, 35] or [0, 70] for example.

2 User Interface

Here we would like to familiarize the reader with the user interface of our application. The application has multiple tabs for customizing parameters of individual systems such as LBM, STLP, terrain, particle rendering and many more. There are also two viewport modes: 2D diagram viewport and a 3D viewport. Please note that both modes share the same UI panels. As shown in Figure 2.1, the 3D viewport of the application offers a main menubar, two sidebars, frame rate measurement head-up display, overlay textures, and an overlay diagram.

In the 2D diagram viewport, users can navigate the STLP diagram by moving around the diagram or by zooming in and out to view individual intersections or other details they are interested in by using the middle mouse button's click and scroll functionalities, respectively. Furthermore, curve editing (if enabled in STLP panel) is possible by dragging individual ambient temperature or dew point temperature curve vertices along their isobars.

Each sidebar of the UI can be set to one of twelve modes that pertain to individual systems or settings. These are LBM options, lighting settings, terrain customization and settings, sky settings, cloud visualization, STLP, emitter controls, view options, debugging pane, scene hierarchy, properties tab, and particle settings. We will shortly present all these tabs in their default settings. Please note that there is a large amount of contextual and pop-up menus in our UI and not all will be covered in the upcoming pages.

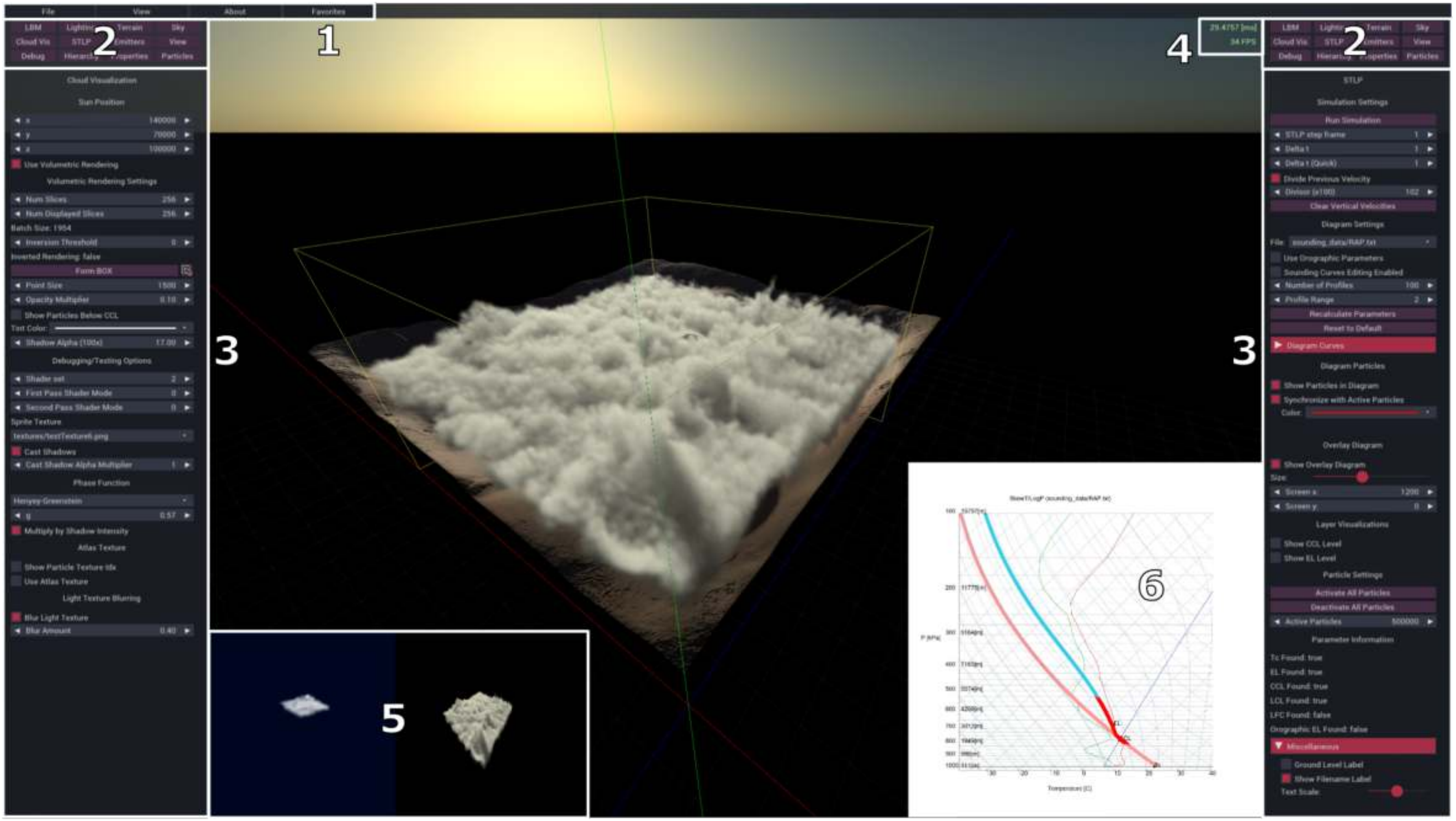

Figure 2.1: User interface of the application showing the horizontal toolbar (1), left and right sidebar content selectors (2), left and right sidebars (3), frame rate head-up display (4), overlay textures (5) and an overlay STLP diagram that displays particle positions on the adiabat curves as red points (6).

Figure 2.2: LBM Tab.

In the LBM tab, users can play or pause the simulation. Resetting the simulation is also possible by reinitializing the lattice to its default weights. Value of τ can be set. The same goes for a property named step frame. Step frame is an integer n that makes sure that the simulation is only run in frames with index that is a multiple of n. Inlet settings such as inlet velocity vector, respawn mode and active inlet walls are further customizable. Subgrid model can also be enabled and disabled at runtime. Velocity multiplier is the artificial multiplier applied when moving the cloud particles. Extended collision step can also be enabled, this is however **unstable** in its current form. If the simulation diverges, we recommend stopping the LBM simulation, then resetting the lattice to default and loading/refreshing particle positions.

LBM simulation area can be edited as well. The area can be positioned and scaled freely in the world space. Lastly, the area can be snapped to ground based on its four bottom corners. The LBM editing shows a secondary bounding box when enabled. Changes made to the area can be saved or discarded as shown.

Streamlines are also managed in the LBM tab. First, streamline instance must be created by defining their count and maximum length. After creation, options such as setting horizontal or vertical lines of seeds are present. Additionally, tracked particles can be reset to their initial positions. Finally, the generated streamline instance can be torn down and replaced with a new one with different streamline count and length arguments.

Figure 2.3: SkewT/LogP Tab.

In the SkewT/LogP (STLP) tab users can control both the STLP simulation as well as the diagram settings. Simulation can be played or paused. Similarly to LBM step frame, we provide an option to skip simulation step in the defined number of frames. Δt can also be set using one of the present widgets. Note that the value denoted with (Quick) only acts as a faster slider for convenience and does not have any impact on the simulation. “Divide Previous Velocity” denotes our damping factor whose 100x multiple can be set manually. The 100x multiplier is used so that users can fine-tune the simulation with high precision. An option to clear the vertical velocity (v_y) CUDA array to zeros is also provided.

In the diagram settings, users can select the sounding data file and whether to use orographic parameters. The number of convective temperature profiles to be used and their range in degree Celsius is also modifiable. Moreover, sounding curve editing can be enabled or disabled. After any changes to the diagram are made, “Recalculate Parameters” button must be used to recalculate all curves for the simulation. If a different sounding file was selected, “Load Sounding File” button (contextual) must be pressed for the changes to take place. In the diagram curves panel, groups of curves can be hidden and their color can be changed.

Diagram particle visualization controls are also present in this tab. Diagram particles can be hidden and their active count can be synchronized with the particle system. Moreover, their visualization color can be set. Overlay diagram settings are also available. The overlay diagram can be scaled and positioned on the screen. CCL and EL levels can be visualized in the 3D viewport using their perspective checkboxes. Additionally, particle settings and parameter information panel are shown. Lastly, a miscellaneous panel gives options for scaling diagram text and changing visibility of certain diagram labels.

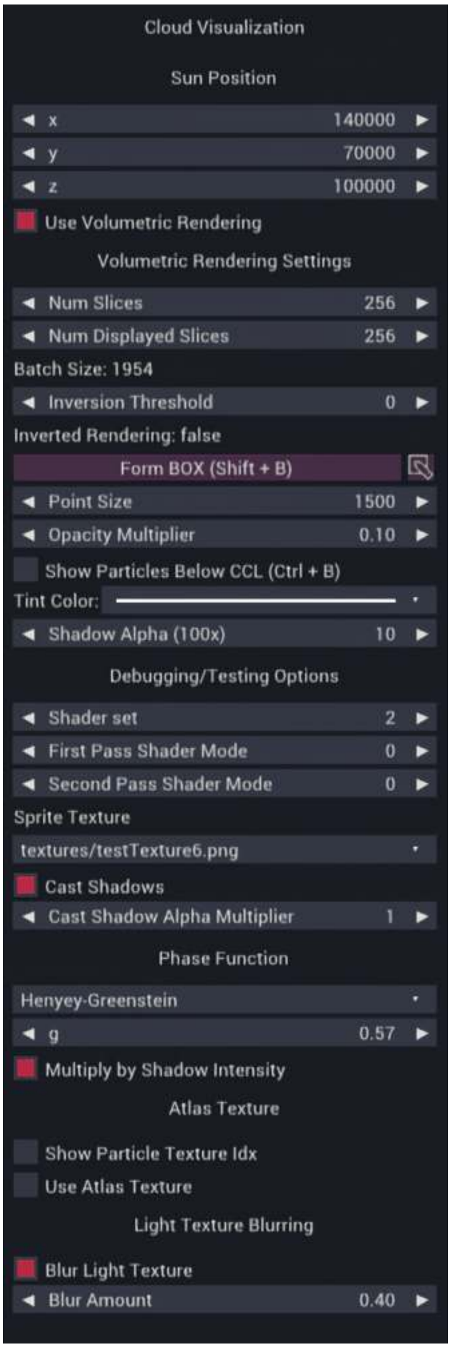

Figure 2.4: Cloud Visualization Tab.

In the cloud visualization tab, the sun position can be adjusted. The volumetric rendering can be enabled/disabled. In the image, options for the volumetric rendering are illustrated. In its settings, number of slices and number of displayed slices can be customized. Based on its count, the tab also shows what is the current particle batch size for a single draw call. Inversion threshold denotes a value against which the $\cos(\vec{l}, \vec{v})$ is compared. The panel provides information about whether the inverted rendering is currently in action. Debugging option to form a box out of the particles is also available. The box can be positioned and scaled by clicking the edit button on its right. Point size, opacity multiplier, tint color and shadow alpha determining how dark the occluded particles are, are also editable. “Shader Set” and “First/Second Pass Shader Mode” change the used shader set and debugging shader uniforms, respectively. Additionally, the point sprite texture can be changed as well.

Our cloud rendering approach also supports casting shadows which can be enabled or disabled freely. Multiplier of the cast shadow can be adjusted. Phase functions are also changeable at runtime. Appropriate settings for each phase function will be presented. An option to multiply the phase function result with shadow intensity is also available.

Experimental usage of texture atlases as sprite textures can be enabled including a visualization mode showing color coded particles based on their atlas texture indices. Lastly, light texture blurring can be enabled/disabled and its intensity can be set.

Figure 2.5: Lighting Tab.

In the lighting settings tab, the sun position, focus point and the sun's orthographic projection can be customized. Furthermore, settings for the exponential variance shadow mapping are present including shadow bias, whether to show shadows only, and whether to use gaussian blur on the generated shadow depth map. Fog settings are also present. Users can select fog mode (linear or exponential) and its settings: minimum and maximum distance for linear fog, falloff for exponential fog, and its intensity and color. Lastly, directional light settings such as its intensity (only for PBR shaders) and whether to sample the sky for tinting its color are shown.

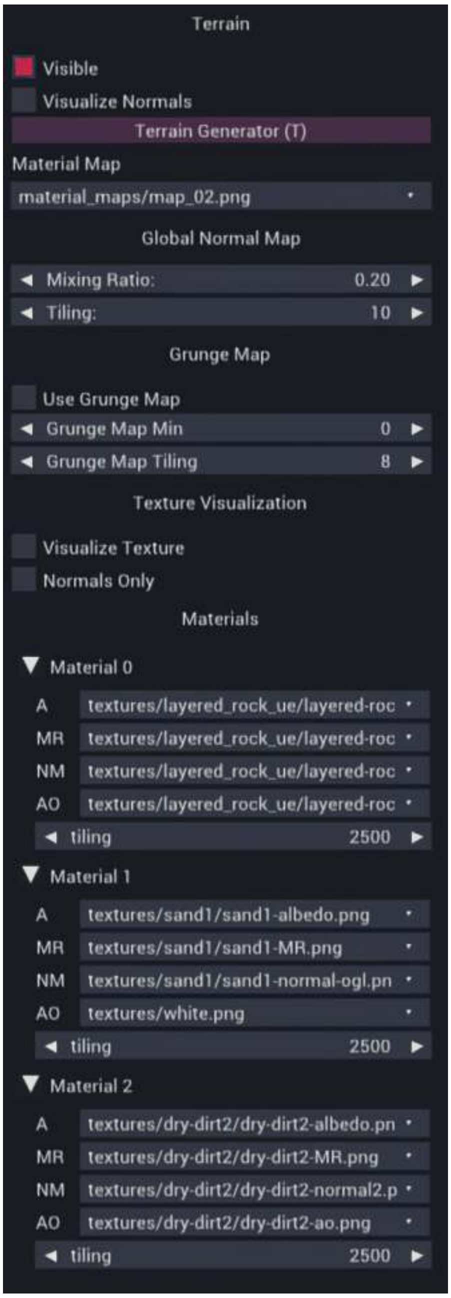

Figure 2.6: Terrain Tab.

In the terrain tab, users can hide the terrain, choose to visualize its normals (using a geometry shader), change its material map, or set properties of its global normal map and grunge map that break repeating patterns of its tiled materials. Furthermore, users can select any texture to be projected onto the terrain for easier debugging. Normal map visualization of the terrain is also present in two modes: display default normals of the terrain mesh or display normals generated from its normal map textures. Last but not least, panel of all materials is present where albedo, metallic roughness, normal map and ambient occlusion textures and their tiling can be changed for each material individually.

Lastly, a button to open terrain generation window is present. The terrain generator window then provides options to change altitude range of the terrain and the world size of texels. Terrain generation mode can be selected. Terrains can be either generated using heightmap texture or perlin noise. If perlin noise is selected, its settings are shown such as its frequency, number of octaves, persistence (stacked multiplier applied to each octave, e.g. if set to 1/2, second octave will have intensity 1/2, third octave 1/4, n-th octave $(1/2)^n$), and its mode (basic in range [-1, 1], normalized to range [0, 1], or turbulent (absolute value)).

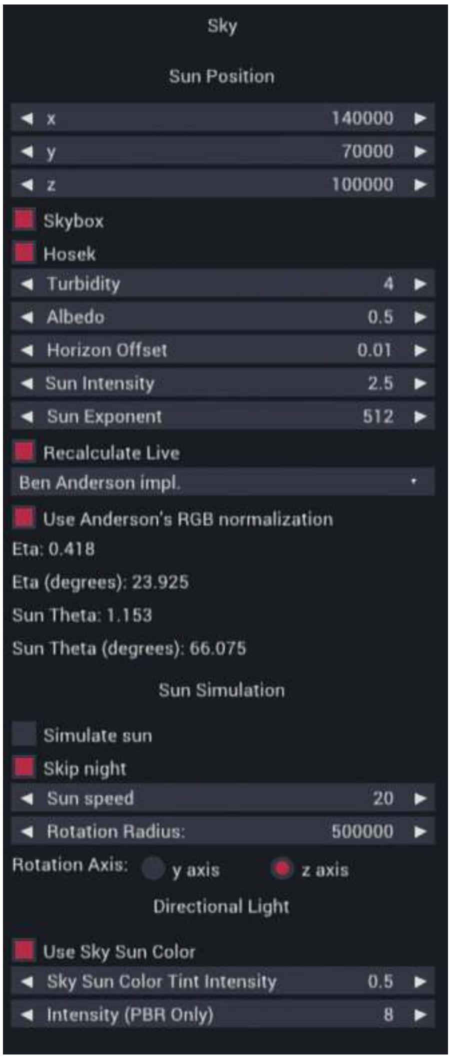

Figure 2.7: Sky Tab.

In the sky tab, users can select whether to use a skybox instead of a single color background. When a skybox is enabled, regular HDRI images or the dynamic model by Hošek and Wilkie can be used for its rendering. If the dynamic model is selected, options such as its turbidity, albedo and sun visualization are shown. Option to recalculate the model live (each frame if its properties such as sun elevation have changed) is present and recommended to keep enabled. Debugging information such as sun elevation are shown.

Furthermore, sun movement simulation option and its parameters such as speed of movement and rotation axis (x or z) are present. Lastly, the same directional light settings as in Figure 2.5 are provided.

Figure 2.8: Emitters Tab.

In the emitters tab, users are presented with an option to create new emitters by opening an emitter creation window. Brush mode can also be enabled or disabled.

If the brush mode is disabled, list of all emitters is shown. Each emitter in the list can be customized individually. Customization options differ based on the emitter type. All emitter share basic options such as whether it is enabled and visible, how many particles it emits per frame, and what is the range of profile indices the emitted particles fall into. The position and scale is also editable in case of positional emitters.

If the brush mode is enabled, users can select the active brush from all available positional emitters. If active brush is selected, users can click and draw/emit particles on the terrain. Using the mouse wheel, users can change the scale of the active brush. If shift is held down, the scrolling changes how many particles are emitted by the active brush. Furthermore, by holding control or the alt key, scrolling adjusts the profile indices of emitted particles.

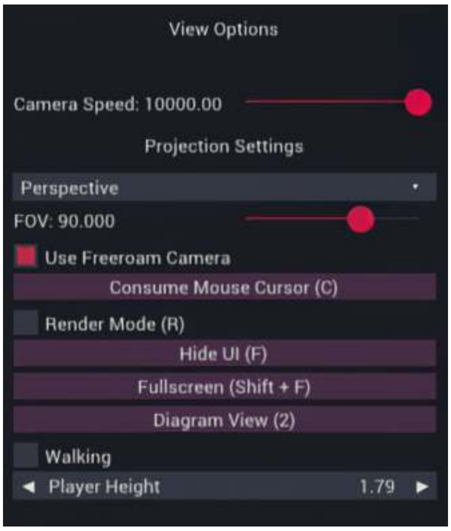

Figure 2.9: View Tab.

The view tab gives users basic options in changing camera speed, projection mode (perspective or orthographic), whether to use a freeroam camera, or whether the camera is snapped to ground, hence simulating walking on ground. Render mode can be also enabled which hides all helper structures such as grids or axes. Field of view (FOV) is also editable when perspective projection is used. On the other hand, if orthographic projection is used, users can set front, side and top views as in popular 3D modelers like Blender or Maya. Lastly, users can switch between the 3D and 2D viewport.

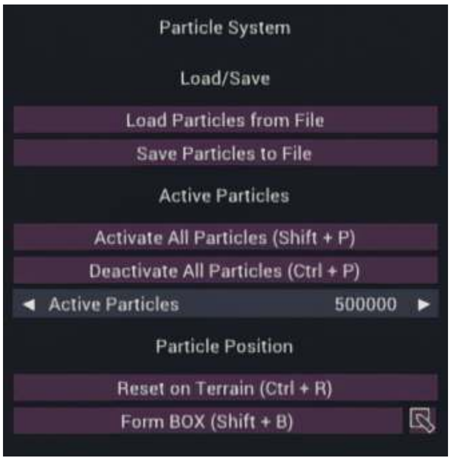

Figure 2.10: Particles Tab.

In the particles tab, users can load and save current particle positions to file. Particle positions can also be set by creating a box from the particles or by refreshing the particles uniformly across the terrain surface.

Figure 2.11: Debug Tab.

The debug tab contains useful functions and information for debugging the whole application.

Its first section shows all timers for the application. Each timer can be controlled individually and show all its accumulated data such as average time, frame time, maximum and minimum captured times, and options to enable GPU synchronization before and after it starts or ends timing. The GPU synchronization is further divided into two synchronization calls: `glFinish()` for synchronizing OpenGL pipeline and `cudaDeviceSynchronize()` for synchronizing CUDA kernel execution. Timers can also be operated all at once using global start, reset and end buttons. Benchmarking checkbox can be used if the user wants the application to automatically generate `.csv` files containing all frame times captured by the timers.

Second section of the debugging tab contains controls of overlay textures. Here, users can set visibility of each overlay texture slot, select any texture to be displayed within it and whether to show its alpha texture.

Lastly, general information about the application such as terrain resolution, camera position, lattice dimensions, etc., is provided.

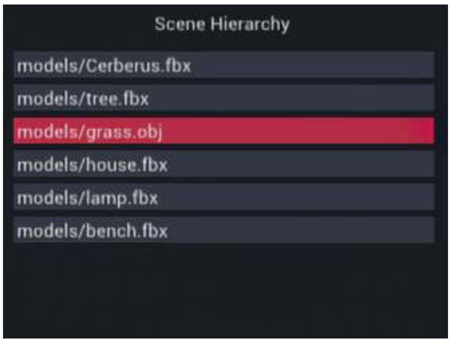

Figure 2.12: Hierarchy Tab.

The hierarchy tab simply contains the scene hierarchy as users would expect from any other game/rendering engine. Individual objects can be selected and their detailed properties are then shown in the properties tab (see Figure 2.13). Multiple objects can be selected at once and will all be shown in the properties tab.

Figure 2.13: Properties Tab.

Object details such as its transform (position, rotation, scale) and whether the object is visible or casts shadows are shown in the properties tab. Furthermore, options such as object snapping to ground or unparenting are present. If the selected object is an instanced model, it is also possible to refresh its instances by assigning them a new set of random positions on the terrain.