Mobile Touch Camera

How to use

Camera

Attach the MobileTouchCamera script to the game camera. Done. Ensure proper parameter set-up. For sidescroller cameras, make sure CameraAxes is set to XY. For top-down cameras, make sure XZ is set as axes.

Note: For perspective cameras it is recommended that the ground of your scene is rooted at y = 0 for top-down cameras or at z = 0 for side-scrolling cameras. In case this cannot be ensured, the proper offset must be set in the Ground Level Offset inspector variable.

Parameters

- **Camera Axes:** You need to define whether your camera is a side-view camera (which is the default when using the 2D mode of unity) or if you chose a top-down looking camera. This parameter tells the system whether to scroll in XY direction, or in XZ direction.
- **PerspectiveZoomMode:** When using a perspective camera, the zoom can either be performed by changing the field of view, or by moving the camera closer to the scene.
- **Cam Zoom Min/Max**:
  - For orthographic cameras this value denotes the minimum and maximum camera size that will not be surpassed when zooming in or out. Example values: 4/12
  - For perspective cameras with field of view based zoom, this value denotes the minimum and maximum field of view. Example values: 30/90
  - For perspective cameras with translation based zoom, this value controls the minimum and maximum distance of the camera from the ground. Example values: 5/30
- **Cam Overzoom Margin**: The cam will overzoom the min/max values by this amount and spring back when the user releases the zoom.
- **Cam Overdrag Margin:** When dragging the camera close to the defined border, it will spring back when the user stops dragging. This value defines the distance from the border where the camera will spring back to.
- **Boundary**: These values define the scrolling borders for the camera. The camera will not scroll further than defined here. Depending on the Camera Axes this either affects the XY or the XZ position of the camera. The boundary is drawn as yellow rectangular gizmo in the scene-view when the camera is selected. You can modify the boundary using the gizmo handles.

- **Cam Follow Factor**: The lower the value, the slower the camera will follow movement updates. The higher the value, the more direct the camera will be. This value is necessary for keeping the camera smooth when the framerate is not in sync with the touch input update rate. For most cases it can be left at its default setting.
- **Auto Scroll Damp Mode:** When dragging quickly, the camera will keep autoscrolling in the most recent direction. The autoscrolling will slowly come to a halt. This value allows to choose between 2 preset slow-down behaviours (Default – quickly comes to a halt, Slow Fade Out – slows to halt a little more smoothly than Default) or to create your own behaviour (by selecting Custom). When creating your own behaviour, the following tweakables become visible in the inspector:
    - **Auto Scroll Damp:** When dragging quickly, the camera will keep autoscrolling in the last direction. The autoscrolling will slowly come to a halt. This value defines how fast the camera will come to a halt.
    - **Auto Scroll Damp Curve:** This curve allows to modulate the auto scroll damp value over time.
- **Ground Level Offset**: The camera assumes that the scrollable content of your scene (e.g. the ground of your game-world) is located at y = 0 for top-down cameras or at z = 0 for side-scrolling cameras. In case this is not valid for your scene, you may adjust this property to the correct offset.
- **Enable Rotation:** When enabled, the camera can be rotated using a 2-finger rotation gesture.
- **Enable Tilt:** When enabled, the camera can be tilted using a synced 2-finger up or down motion. Enabling this feature sets the following tweakables to visible in the inspector:
    - **Tilt Angle Min:** The minimum tilt angle for the camera.
    - **Tilt Angle Max:** The maximum tilt angle for the camera.
- **Enable Zoom Tilt:** When enabled, the camera is tilted automatically when zooming in or out. Enabling this feature sets the following tweakables to visible in the inspector:
    - **Zoom Tilt Angle Min:** The minimum tilt angle for the zoom tilt.
    - **Zoom Tilt Angle Max:** The maximum tilt angle for the zoom tilt.
- **On Pick Item (2D)**: Here you can set up callbacks to be invoked when an item is tapped on.
- **On Pick Item (2D) Double Click**: Here you can set up callbacks to be invoked when an item is double-tapped on.

Editor Keys

The following key combinations can be used in the editor to control the camera:

Ctrl or Alt + Num Pad Plus/Minus … Zoom

Ctrl or Alt + Arrow Key Left/Right … Rotate (Only available when the feature is enabled)

Ctrl or Alt + Arrow Key Up/Down … Tilt (Only available when the feature is enabled)

Picking

For item picking & dragging, attach the MobileTouchPickable script to any pickable item that also has a collider component. Once this component is attached, you can select the item by tapping on it, and then drag it around once it's selected.

In case the collider of a pickable item is not placed at the root of the item but on a child instead, you may want to set "Pickable Transform" to the root gameObject that you want to move by dragging.

For advanced settings, attach the MobilePickingController script to the game camera. You can then tweak the following variables.

Mobile Picking Controller Parameters

- **Snap To Grid**: When set to true, the position of dragged items snaps to discrete units.
- **Snap Unit Size**: Size of the snap units when snapToGrid is enabled.
- **Snap Offset**: When snapping is enabled, this value defines a position offset that is added to the center of all objects when dragging. When a top-down camera is used, these 2 values are applied to the X/Z position.
- **Snap Angle**: When set to Straight, picked items will be snapped to a perfectly horizontal and vertical grid in world space. Diagonal snaps the items on a 45 degree grid.
- **Is Multi Selection Enabled:** When this flag is enabled, more than one item can be selected and moved at the same time.
- **Require Long Tap For Move:** When setting this variable to true, pickables can only be moved by long tapping on them first.
- **On Pickable Transform Selected:** Here you can set up callbacks to be invoked when a pickable transform is selected.
- **On Pickable Transform Selected Extended:** Here you can set up an extended callback to be invoked when a pickable transform is selected. This callback is triggered at the same time as OnPickableTransformSelected but has more details about the selection in the passing parameter.
- **On Pickable Transform Deselected:** Here you can set up callbacks to be invoked when a pickable transform is deselected.
- **On Pickable Transform Move Started:** Here you can set up callbacks to be invoked when the moving of a pickable transform is started.

- **On Pickable Transform Moved:** Here you can set up callbacks to be invoked when a pickable transform is moved to a new position.
- **On Pickable Transform Move Ended:** Here you can set up callbacks to be invoked when the moving of a pickable transform is ended. The event requires 2 parameters. The first is the start position of the drag. The second is the dragged transform. The start position can be used to reset the transform in case the drag has ended on an invalid position.

In addition to the global snap offset setting on the Mobile Picking Controller, an additional local offset can be defined for particular pickables by tweaking the Local Snap Offset variable on the Mobile Touch Pickable component.

More Information

https://bitbendergames.wordpress.com/unity-asset-store-support/

Credits

Programming: BitBender Games

Asset Store Background: http://opengameart.org/content/trees-bushes

Demo Scene Graphics: http://opengameart.org/content/lpc-a-shootem-up-complete-graphic-kit

Testing Devices: Rarebyte OG

Version history

v2.4
- Removed compiler warnings.
- Add better auto-scroll algorithm that fixes scale-dependent behaviour.

v2.3
- Fix jumping-bug when using touch input on laptops with touchscreen.
- Add WebGL to default list of keyboard & mouse input platforms.

v2.2
- Added tool script to disable touch camera input on all UI.
- Add tweakables for mouse & keyboard input in PC builds.

v2.1
- Fixed null ref exception and other issues when setting the Unity Timescale to 0.
- Added expert mode tweakable to allow changing camera overdrag in horizontal and vertical direction independent from eachother.

v2.0

- Multi-Selection. It's now possible to enable a multi-selection mode for pickables where more than 1 item can be selected and dragged around at the same time.
- Fixed minor issue with local snap offset when moving pickables.
- Misc minor fixes.

v1.9
- Extended custom editor to allow modifying the boundary via handles in the scene.
- Added option for automatically tilting the camera when zooming in and out.
- Improved Camera Focus on Item script.
- Warnings in inspector when boundary values are not correct.

v1.8
- The boundary is drawn as yellow rectangular gizmo when the camera is selected.
- Added a tool-script that allows to fetch the camera boundary from a box collider.
- Added expert mode tweakables to Mobile Picking Controller for fine-tuning the selection/deselection behaviour.
- Some more internal tweakables exposed via expert mode of the custom inspector.
- When adding the Mobile Touch Camera to a new Game Object, default values for the camera-axis- and zoom-value-settings are derived from the camera type and orientation. Translation based zoom is now the default value for perspective cameras.
- Adding ability to set a terrain-collider in the expert mode of the Mobile Touch Camera. When set, picked items are aligned to the terrain when they area dragged by a user.
- Tweaked the tilt behaviour to reduce cases where the camera would go into tilt mode when zoom or rotation was intended.

v1.7
- Added an Expert Mode that allows to tweak hidden internal variables through the inspector.
- Fixed issue with offset of dragged items.
- Added ability to select and move items via a long tap instead of 2 separate taps.
- Added event for finding out if a pickable item has been selected through regular tap, double tap, or long tap.

v1.6
- Added 2-finger gesture for tilting the camera.
- Added more freedom to tweak auto-scroll damp and added ready-to-use presets.
- Fixed issues with boundary computation code.

v1.5
- Added camera rotation gesture.
- Added some more events for item dragging. Useful for showing special markers when the item drag starts, or for resetting item position when the drag ends on invalid positions.

v1.4
- Added translation-based pinch zoom for perspective cameras.
- Added events for item touble tap.
- Added event for selected item updating position due to being dragged by the user.
- Added helper script FocusCameraOnItem for auto-focusing the camera on a selected item or collider.
- Switched to using a custom inspector for the MobileTouchCamera component.

v1.3
- Proper tooltips added to all inspector variables.

- Added snapping of dragged items.
- Reworked boundary clamping algorithm.

v1.2
- Moved all position update code to LateUpdate().
- Added position interpolation to fix stutter problems on iPads running games at targetFrameRate=60.
- Warnings when the camera doesn't look in the direction defined by the axes.
- Fixed position offset of picked items in orthographic camera mode.
- Fixed pinch to zoom in unity remote-testing and win-standalone mode.

v1.1
- Made the manual zoom-factor obsolete. Zooming-speed now purely depends on the distance of the fingers when pinching starts.

v1.0
- Initial release.