Picking3dAPI

Proposed 3D Picking API:

MouseEvent, MouseDragEvent, DragEvent, GestureEvent, ContextMenuEvent and TouchPoint classes:

```
    /**
     * Returns information about the pick.
     */
    PickResult getPickResult()

    /**
     * Depth position of the event relative to the
     * origin of the MouseEvent's source.
     */

    double getZ()
```

Add a PickResult argument to all their constructors. The existing constructors were not released yet so we can do without adding new constructors.

The semantics of the coordinates is as follows: the event's getSceneX(), getSceneY() values still represent the 2D coordinates of the mouse in the window content pixels (screen coordinates minus window position and decorations). The getX(), getY(), getZ() represent the intersection point in the local 3D space, so for scene they are no longer the same. I believe this is the natural extension (or even a bug fix) because

* for the 2D ParallelCamera case nothing changes, just the getZ() returns the Z coorinate of the intersection point
* for obtaining mouse position within the scene, the getSceneX(), getSceneY() are still available
* the getX(), getY(), getZ() now contain the correct position in local space of the event handling node/scene (instead of a complete nonsense)

Add PickResult class:

```
/**
 * A container object that contains the result of a pick event
 */
PickResult

    /**
     * An undefined face. This value is used for the intersected face
     * if the picked node has no user-specified faces.
     */
    public static final int FACE_UNDEFINED = -1;

    /**
     * Returns the intersected node.
     * Returns null if there was no intersection with any node and the scene
     * was picked.
     */
    Node getIntersectedNode()

    /**
     * Returns the intersected point in local coordinate of the picked Node.
     * If no node was picked, it returns the intersected point with the
     * projection plane.
     */
    Point3D getIntersectedPoint()

    /**
     * Returns the intersected distance between camera position
     * and the intersected point. Returns POSITIVE_INFINITY in case of
     * parallel camera.
     */
    double getIntersectedDistance()

    /**
     * Returns the intersected face of the picked Node, FACE_UNDEFINED
     *         if the node doesn't have user-specified faces
     *         or was picked on bounds.
     */
     int getIntersectedFace()

    /**
     * Return the intersected texture coordinates of the picked 3d shape.
     * If the picked target is not Shape3D or has pickOnBounds==true,
     * it returns null.
     */
     Point2D getIntersectedTexCoord()
```

```
/**
 * Creates a new instance of PickResult.
 */
public PickResult(Node node, Point3D point, double distance, int face, Point2D texCoord)

/**
 * Creates a new instance of PickResult for a non-3d-shape target.
 * Sets face to FACE_UNDEFINED and texCoord to null.
 */
public PickResult(Node node, Point3D point, double distance)

/**
 * Creates a pick result for a 2D case where no additional information is needed.
 * Converts the given scene coordinates to the target's local coordinate space
 * and stores the value as the intersected point. Sets intersected node
 * to the given target, distance to POSITIVE_INFINITY,
 * face to FACE_UNDEFINED and texCoord to null.
 */
public PickResult(EventTarget target, double sceneX, double sceneY)
```