**App Builder Key Terms**

**Buttons:**

These are perhaps the simplest way in which a game developer can get input from the game player. Buttons can be used to create a joystick (with controls for moving up, down, left, right), and to perform any other actions in response to a button click.

**Canvas:**

Canvas components are useful for multiple reasons. First, they are the component where game objects are drawn -- shapes such as rectangles, ovals, etc. can be drawn on these using the methods provided by AppInventor in the Blocks Editor, and images can be uploaded as backgrounds. Second, they are containers for other components known as sprites (described below), and provide a set of bounds within which sprite objects can be drawn and moved around. Third, Canvas objects can be used as a source of touch-input, and they provide a "dragged" and "touched" event that the app developer can respond to. Fourth, these objects can be used as a spacing mechanism in layouts, e.g. if a developer wishes to center a button on the screen, he/she can add an empty/transparent canvas on each side within a Horizontal Layout, and set the width property of each canvas to "fill parent" -- magically, the end result will be the centering of the button or whatever other object is between the two outside canvases.

**Clock:**

Clocks are instrumental for game development since they are the time-keepers. They have an Interval property which the game developer can define and modify at any time, and they fire an event after each interval expires. The game developer can easily react to the firing of these clock timer events in the Blocks Editor in order to perform certain tasks (such as updating the location of a Sprite) every few milliseconds.

**Sound:**

Sound effects are easy to add in App Inventor and can make a significant impact on perceived game quality. The 'Sound' component is an invisible component that allows the developer to easily select a source sound file and to easily play this file in response to other actions from the Blocks Editor.

**Ball:**

The ball component is perfect for games that actually require balls that need to move around on a canvas. The ball component supports rectangle-based collision detection and implements the physics of ball movement/bouncing. The game developer can provide it a direction (called a heading), a speed, a radius, and the time interval the ball should wait between each pair of movements.

**Image Sprite:**

Sprites are a fundamental part of any game that requires animation or motion. They are essentially more generic 'ball' components in that they can be visualized by any image that the developer uploads (animated gifs not supported). They support rectangle-based collision detection and are restricted to movement within the bounds of the canvas in which they reside (they can't go off-screen).

**Game Client:**

This component is the foundation for building multi-player games with AppInventor. It allows for web-based communication between multiple clients (phones) and a single game server that the game developer deploys independently.