# Video Game Development Glossary (terms)

**AI**: the logic that is used to determine the computer’s decision.

**Assets**: Including images, sound effects, fonts, 3D models and textures, these are any external pieces of content you’ll be importing into your game.

**Collision Detection**: Detecting an intersection between one or more objects in your game. Player & enemy, enemy & weapon, player & wall, etc.

**Boundary collision detection** is done by putting a rectangle around game objects and checking if the rectangles intersect. This isn’t always exact due to different shaped objects in your game.

**Pixel perfect collision detection** can make boundary collision detection more exact by checking the pixels involved in the rectangle intersection. This is an expensive operation for your game so don’t check more pixels than you have to!

**Game Engine**: An engine is a system designed to for the creation and development of games. An engine allows for reuse by letting the developer use different art and data sets. You can use a game engine that’s already available or build your own.

**Game Loop**: This is what makes your game run. The game loop run continuously, updating your game based on player input and drawing the changes to the display. If you use a game engine, the game loop might be provided for you. It consists of the core methods: load, initialize, update and draw. For more information check out Game Development 101: Part 2.

**Game State**: All the information that defines an object at a particular time. This covers everything from where an object is positioned to whether a gamepad button is being pressed. Often you’ll compare one state to another and take some action based on the result, such as firing a weapon when the space bar is pressed and released.

**NPC**: Non-Player Character. This is a character that is not controlled by the person playing the game. This often refers to character that the player talks to as part of the story in the game.

**Player Input**: A game has to have a way to take player input. This can be from the keyboard, a game controller, mouse, pen, touch or even audio commands. If you port games from one platform to another, you may have to update how to accept player input based on hardware form factors.

**Sounds**: Of course you know what sounds are, but having sounds and audio are an important part of a game. Sound effects, background music, level music, battle music, etc. are all great ways to give the player a more immersive game experience.

**Sprite**: Several 2D graphics images of one object or character for each of it’s possible positions and states. The game engine displays the correct image based on the object’s or character’s position.

**Texture**: Images that are loaded into your game. Textures are used for backgrounds, enemy images, bullet images, etc.

**Vector**: A vector is a set of two or more numbers, often used to indicate a position (on an imaginary grid) or a direction. A two-component vector can be used to for two-dimensional space: (x,y); a three-component vector can be used for three-dimensional space (x,y,z); and so on. You might use a vector to define the location of a character on screen and you would update the vector value to reflect movement.