The main character (Doji) performs attacks based on the directional keys pressed immediately before the attack button is pressed. For instance, if the player presses Down->Right->Attack, Doji will perform that attack mapped to the set of keys matching Down->Right->Attack.

In the game engine, the container that holds all registered set of attack key combinations is a multimap. The entries in a map contains a pair between:

* A vector of key or keys
* The name of a registered attack

To recap, the character’s move list is stored in a std::multimap container, with each entry a std::pair of one vector of keys and one registered attack name:

std::multimap<std::vector<sf::Keyboard::Key>, std::string> mMoveList;

The vector of keys of typedef’d as KeyVector.

For instance, Doji has an attack called “Impact” that is performed when the player pressed Down + Attack. To add this attack to the move list:

1. Create a vector of keys (KeyVector) called Impact.
2. Push the Down key into the KeyVector.
3. Finally, make a pair in the move list between the Impact KeyVector and the registered attack name.

KeyVector Impact;   
Impact.push\_back(sf::Keyboard::Down);  
mMoveList.insert(std::pair<KeyVector, std::string>(Impact, Attacks::PLAYER\_IMPACT));

|  |  |
| --- | --- |
| mID | unique global identifier |
| mName | game object name |
| mParent | pointer to parent object |
| mLayerName | name of layer for rendering and collision detection |
| mChildren | vector of child GameObjects |

|  |  |
| --- | --- |
| mRenderComponent | maintains visual properties of the game object, such as textures |
| mInputComponent | enables and controls input for game object |
| mBoxColliderComponent | assigns collider box to object for AABB collision handling |
| mLogicComponent | provides custom behavior for game object |

|  |  |
| --- | --- |
| GameObject | constructor with [GameObjectDesc](http://fatalfarrowprod.blogspot.com/p/techdocssolsticescargameobjectdesc.html) as argument |

|  |  |
| --- | --- |
| addChild | adds passed-in GameObject\* as child of this GameObject |
| addComponent | adds specific component to GameObject |

|  |  |
| --- | --- |
| getRenderComponent() | returns pointer to GameObject's render component |
| getLogicComponent | returns pointer to GameObject's logic component |
| getInputComponent | returns pointer to GameObject's input component |
| getBoxColliderComponent | returns pointer to GameObject's box collider component |