

```
1 #include "stdafx.h"
2
3 #include "ServiceLocator.hpp"
4 #include "Positions.h"
5 #include "Typing.h"
6
7 #include "Item.h"
8
9
10 Item::Item(sf::Sprite* sp)
11 {
12     m_counter = 0;
13
14     m_sprite = sp;
15
16     m_type = ENTITY_ITEM;
17
18     m_visible = true;
19
20     m_collider = new Collider(m_x, m_y);
21     m_collider->SetParent(this);
22     m_collider->SetWidthHeight(m_sprite->getGlobalBounds().width, m_sprite->getGlobalBounds().height);
23
24 }
25
26
27 Item::~Item()
28 {
29 }
30
31 void Item::Update(float deltatime)
32 {
33     if (m_y < -76)
34     {
35         m_type = ENTITY_ITEM;
36         SetInvisible();
37         SetVisible();
38     }
39
40     if (m_hold == false && m_type == ENTITY_ITEMPROJECTILE)
41     {
42         SetPosition(GetX(), GetY() - (GetSprite()->getGlobalBounds().height * deltatime * 20));
43     }
44 }
45
46 m_collider->Refresh();
47
48
49 m_sprite->setPosition(m_x, m_y);
50 }
51
52 sf::Sprite* Item::GetSprite()
53 {
54     return m_sprite;
55 }
56
57 Collider* Item::GetCollider()
58 {
59     return m_collider;
60 }
61
62 float Item::GetX()
63 {
64     return m_x;
65 }
66 float Item::GetY()
67 {
68     return m_y;
69 }
70
71 void Item::SetInvisible()
72 {
73     if (m_visible == true)
```

```
74     {
75         m_visible = false;
76     }
77
78
79     }
80 }
81 void Item::SetVisible()
82 {
83     if (m_visible == false)
84     {
85         m_visible = true;
86
87         Positions* pos = ServiceLocator<Positions>::GetService();
88         Typing* type = ServiceLocator<Typing>::GetService();
89
90         m_pos = pos->setItemPos();
91
92         SetPosition(pos->GetItemPos(m_pos).x, pos->GetItemPos(m_pos).y);
93
94         if (m_pos != -1)
95         {
96             type->newWord(m_word, m_pos);
97         }
98     }
99 }
100 }
101 }
102 bool Item::IsVisible()
103 {
104     return m_visible;
105 }
106
107 EEntityType Item::GetType()
108 {
109     return m_type;
110 }
111
112 void Item::SetType(int entity)
113 {
114     m_type = (EEntityType)entity;
115     if (entity == ENTITY_ITEMPROJECTILE)
116     {
117         Positions* pos = ServiceLocator<Positions>::GetService();
118
119         pos->removeItem(m_pos);
120     }
121 }
122
123 void Item::SetPosition(float x, float y)
124 {
125     m_x = x;
126     m_y = y;
127     m_collider->Refresh();
128 }
129
130
131
132 /* bool IsGrounded() = 0;
133 void SetGrounded(int i) = 0;*/
134
135 bool Item::IsHolding()
136 {
137     return m_hold;
138 }
139
140 void Item::Hold(bool yes)
141 {
142     m_hold = yes;
143 }
144
145 Weakness Item::GetWeakness()
146 {
```

```
147     return (Weakness)1;
148 }
149
150 void Item::setWord(std::string std)
151 {
152     m_word = std;
153 }
```