

KingSchlock /
COS30031-2023-103071494

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

COS30031-2023-103071494 / 08 - Spike - Game State Management / GameStateManagement
/ StateManager.h

...



unknown Finished Task 08 - Spike 'Game State Management'



last week



91 lines (76 loc) · 1.44 KB

Code

Blame

Raw



```
1  #pragma once
2
3  enum STATES {
4      WELCOME,
5      MAIN_MENU,
6      ABOUT,
7      HELP,
8      SELECT_ADVENTURE,
9      GAMEPLAY,
10     NEW_HS,          //New Highscore
11     VIEW_HoF,        //View Hall of Fame
12     QUIT,
13 };
14
15 class State {
16 public:
17     virtual STATES update() = 0;
18     virtual void render() = 0;
19 };
20
21
22 class MainMenu : public State {
23 public:
24     STATES update() override;
25     void render() override;
26 };
27
28 class AboutMenu : public State {
29 public:
30     STATES update() override;
31     void render() override;
32 };
33
```

```
33
34  ✓ class HelpMenu : public State {
35      public:
36          STATES update() override;
37          void render() override;
38      };
39
40  ✓ class AdventureSelectMenu : public State {
41      public:
42          STATES update() override;
43          void render() override;
44      };
45
46  ✓ class GameplayState : public State {
47      public:
48          STATES update() override;
49          void render() override;
50      };
51
52  ✓ class NewHighScoreMenu : public State {
53      public:
54          STATES update() override;
55          void render() override;
56      };
57
58  ✓ class HallOfFameMenu : public State {
59      public:
60          STATES update() override;
61          void render() override;
62      };
63
64  ✓ class QuitState : public State {
65      public:
66          STATES update() override;
67          void render() override;
68      };
69
70  ✓ class StateManager
71      {
72      private:
73          MainMenu _main_menu;
74          AboutMenu _about_menu;
75          HelpMenu _help_menu;
76          AdventureSelectMenu _adventure_select_menu;
77          GameplayState _gameplay;
78          NewHighScoreMenu _new_highscore_menu;
79          HallOfFameMenu _hall_of_fame_menu;
80          QuitState _quit;
81
82          State* _current_ptr = &_amp;main_menu;
```

```
83         STATES _current_state = STATES::MAIN_MENU;
84
85
86     public:
87         StateManager();
88         bool isRunning();
89         void update();
90         void render();
91     };
```

KingSchlock /
COS30031-2023-103071494

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

COS30031-2023-103071494 / 08 - Spike - Game State Management / GameStateManagement
/ StateManager.cpp

...



unknown Started Task 09 - Spike 'Game Data Management'.

4 days ago



181 lines (144 loc) · 5.51 KB

Code

Blame

Raw



```
1  #include <iostream>
2  #include "StateManager.h"
3
4  //State manager definitions
5  StateManager::StateManager() {
6      _current_ptr->render();
7  }
8
9  bool StateManager::isRunning() { return _current_state != STATES::QUIT; }
10
11 void StateManager::update() {
12     //Wrapped up all nice and cozy
13     _current_state = _current_ptr->update();
14
15     if (_current_state == STATES::MAIN_MENU) { _current_ptr = &_amp;main_menu; }
16     else if (_current_state == STATES::ABOUT) { _current_ptr = &_amp;_about_menu; }
17     else if (_current_state == STATES::HELP) { _current_ptr = &_amp;_help_menu; }
18     else if (_current_state == STATES::SELECT_ADVENTURE) { _current_ptr = &_amp;_adventure_se
19     else if (_current_state == STATES::GAMEPLAY) { _current_ptr = &_amp;_gameplay; }
20     else if (_current_state == STATES::NEW_HS) { _current_ptr = &_amp;_new_highscore_menu; }
21     else if (_current_state == STATES::VIEW_HoF) { _current_ptr = &_amp;_hall_of_fame_menu; }
22     else if (_current_state == STATES::QUIT) { _current_ptr = &_amp;_quit; }
23
24 }
25
26 void StateManager::render() { _current_ptr->render(); }
27
28 //Main menu definitions
29 STATES MainMenu::update() {
30     int choice;
31     std::cin >> choice;
32
33     std::cout << std::endl;
```

```
33         std::cout << std::endl;
34
35         switch (choice) {
36         case 1:
37             return STATES::SELECT_ADVENTURE;
38         case 2:
39             return STATES::VIEW_HoF;
40         case 3:
41             return STATES::HELP;
42         case 4:
43             return STATES::ABOUT;
44         case 5:
45             return STATES::QUIT;
46         default:
47             return STATES::MAIN_MENU;
48         }
49     }
50
51     void MainMenu::render() {
52         std::cout << "Zork(ish) :: Main Menu" << std::endl;
53         std::cout << "-----" << st
54
55         std::cout << std::endl << "Welcome to Zorkish Adventures!" << std::endl;
56
57         std::cout << std::endl << "1. Select Adventure and Play" << std::endl;
58         std::cout << "2. View Hall of Fame" << std::endl;
59         std::cout << "3. Help" << std::endl;
60         std::cout << "4. About" << std::endl;
61         std::cout << "5. Quit" << std::endl;
62
63         std::cout << std::endl << "Selected 1-5: " << std::endl;
64     }
65
66
67     //About menu definitions
68     STATES AboutMenu::update() {
69         return STATES::MAIN_MENU;
70     }
71     void AboutMenu::render() {
72         std::cout << "Written by: Thomas Horsley (103071494)" << std::endl;
73         system("pause");
74         std::cout << std::endl;
75     }
76
77
78     //Help menu definitions
79     STATES HelpMenu::update() { return STATES::MAIN_MENU; }
80     void HelpMenu::render() {
81         std::cout << "The following commands are supported: " << std::endl;
82         std::cout << ">> quit" << std::endl << ">> highscore (for testing)" << std::endl;
```

```
83         system("pause");
84         std::cout << std::endl;
85     }
86
87
88     //Select Adventure definitions
89     STATES AdventureSelectMenu::update() {
90         int choice;
91         std::cin >> choice;
92         std::cout << std::endl;
93
94         switch (choice) {
95             case 1:
96                 std::cout << "This is a pretty neat world" << std::endl;
97                 return STATES::GAMEPLAY;
98             case 2:
99                 std::cout << "Wow this worlds pretty cewl" << std::endl;
100                return STATES::GAMEPLAY;
101            case 3:
102                std::cout << "This is the coolest shit i've ever seen :0" << std::endl;
103                return STATES::GAMEPLAY;
104        }
105    }
106
107    void AdventureSelectMenu::render() {
108        std::cout << std::endl << "Zork(ish) :: Select Adventure " << std::endl;
109        std::cout << "-----" << st
110
111        std::cout << std::endl << ">> 1. World" << std::endl;
112        std::cout << ">> 2. Cool World" << std::endl;
113        std::cout << ">> 3. Even COOLER World" << std::endl;
114    }
115
116
117    //Gameplay state definitions
118    STATES GameplayState::update() {
119        std::string command;
120        std::cin >> command;
121
122        if (command == "highscore") { return STATES::NEW_HS; }
123        else if (command == "quit") { return STATES::QUIT; }
124        else { return STATES::GAMEPLAY; }
125    }
126    void GameplayState::render() {
127        std::cout << std::endl;
128        std::cout << "Gameplay stuff goes in here!" << std::endl;
129        std::cout << "Allowed stage 1 commands can be found in the help screen" << std::endl;
130        std::cout << "Please enter test commands" << std::endl;
131        std::cout << ">> ";
132    }
```

```
132     }
133
134
135     //New Highscore state definitions
136     STATES NewHighScoreMenu::update() {
137         std::string name;
138         std::cin >> name;
139
140         return STATES::MAIN_MENU;
141     }
142     void NewHighScoreMenu::render() {
143         std::cout << std::endl << "Zork(ish) :: Select Adventure " << std::endl;
144         std::cout << "-----" << std::endl;
145
146         std::cout << "Holy... someone's jacked and cracked at zorkish, NEW HIGHSCORE!" << std::endl;
147
148         std::cout << "World: [World here]" << std::endl;
149         std::cout << "Score: [Score here]" << std::endl;
150         std::cout << "Moves: [Move count here]" << std::endl << std::endl;
151
152         std::cout << "Please type your name and press 'Enter': " << std::endl;
153         std::cout << ">> ";
154     }
155
156
157     //Hall of fame definitions
158     STATES HalloOfFameMenu::update() { return STATES::MAIN_MENU; }
159     void HalloOfFameMenu::render() {
160         std::cout << std::endl << "Zork(ish) :: Select Adventure " << std::endl;
161         std::cout << "-----" << std::endl;
162
163         std::cout << "1. [Name], [World], [Score]" << std::endl;
164         std::cout << "2. [Name], [World], [Score]" << std::endl;
165         std::cout << "3. [Name], [World], [Score]" << std::endl;
166         std::cout << "4. [Name], [World], [Score]" << std::endl;
167         std::cout << "5. [Name], [World], [Score]" << std::endl;
168         std::cout << "6. [Name], [World], [Score]" << std::endl;
169         std::cout << "7. [Name], [World], [Score]" << std::endl;
170         std::cout << "8. [Name], [World], [Score]" << std::endl;
171         std::cout << "9. [Name], [World], [Score]" << std::endl;
172         std::cout << "10. [Name], [World], [Score]" << std::endl << std::endl;
173
174         system("pause");
175         std::cout << std::endl;
176     }
177
178
179     //Quit state definitions
180     STATES QuitState::update() { return STATES::QUIT; }
181     void QuitState::render() { std::cout << "Quitting zorkish!" << std::endl; }
```


KingSchlock /
COS30031-2023-103071494

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights



COS30031-2023-103071494 / 08 - Spike - Game State Management / GameStateManagement



/ main.cpp



unknown Nearly finished, bug with menu's double rendering.



last week



21 lines (16 loc) · 379 Bytes

Code

Blame

Raw



```
1  /*      Task 08 - Game State Management
2  *      Author - Thomas Horsley (103071494)
3  *
4  *      The state manager was built implementing the OO State pattern.
5  *      See spike report! */
6
7  #include <iostream>
8  #include <string>
9  #include "StateManager.h"
10
11
12  int main() {
13      StateManager _game_manager;
14
15      while (_game_manager.isRunning()) {
16          _game_manager.update();
17          _game_manager.render();
18      }
19
20      return 0;
21  }
```

