

Семинар #3: Полиморфизм. Домашнее задание.

Задача 1. Ползунок:

Создайте класс `Slider`, который будет описывать элемент интерфейса ползунок. Графическое оформление на ваше усмотрение, в качестве примера:



Ползунок должен работать также как и обычный ползунок в ОС Windows или Linux. При нажатии на сам ползунок и зажатии кнопки, он переходит в состояние перемещение и остаётся в нём до момента отпущения клавиши мыши (даже если курсор вышел далеко за пределы полоски). Другой способ регулирования положения ползунка - это нажатие на саму полоску. В этом случае ползунок сразу перемещается в выбранное место. При изменении положения ползунка должен меняться текст, показывающий числовое значение. Минимальное и максимальное значение ползунка должно задаваться в конструкторе и храниться в частных переменных.

- Создайте круг и 1 ползунок. При изменении положения ползунка должен меняться радиус этого круга.
- Создайте ещё 3 ползунка. При изменении положения этих ползунков должен меняться цвет круга (RGB компоненты).

Объедините классы `Slider` и `Button` в единую иерархию наследования. Класс `Button` можно найти в папке `simple_gui/button`. Оба класса должны наследоваться от абстрактного класса `Widget`. Создайте программу, которая будет использовать несколько объектов класса `Button` и несколько объектов класса `Widget`, используя полиморфизм.

Задача 2. Дерево навыков

В папке `skilltree` находится реализация простого дерева навыков для компьютерной игры.

- **Node** – абстрактный класс, который описывает узел дерева навыков. Узел может быть в трёх состояниях
 - **Blocked** – заблокированный узел. При нажатии на такой узел ничего не должно происходить.
 - **Unblocked** – разблокированный узел. При нажатии на такой узел он активируется. При активации узла, все его дети разблокируются.
 - **Activated** – активированный узел.
- **HitNode** – абстрактный класс, который описывает круглый переключаемый узел дерева навыков. При нажатии на этот узел, если он разблокирован, узел активируется. При повторном нажатии на этот узел, он деактивируется и все его потомки становятся заблокированными.
- **ShieldSkillNode**, **HandSkillNode** и другие – конкретные классы узлов.

Подзадачи

1. **Новый навык** – добавьте новый навык – огненный шар (файл изображения `icon_fireball.png`). Добавьте этот навык в дерево навыков.
2. **Накопительный навык** – создайте новый узел `AccumulativeNode`, наследник класса `Node`. В отличие от узла `HitNode` в этот узел можно вкладывать несколько очков навыка. При нажатии на этот навык, если он разблокирован, он становится активированным, а все его дети разблокируются. При повторном нажатии на этот левой кнопкой мыши у этого узла увеличивается уровень активации до некоторого максимального значения. При нажатии правой кнопкой мыши уровень активации должен уменьшаться на 1. Если уровень достигнет нуля, то узел перестаёт быть активированным, а все его потомки становятся заблокированными.



Визуально этот узел должен быть квадратным (учтите это при определении столкновения при нажатии кнопки мыши). Внизу этого узла должен быть написан уровень активации в виде a / b , где a – текущий уровень активации, а b – максимальный уровень активации.

3. **Наследники накопительного навыка** Создайте 3 класса наследника `AccumulativeNode`. Изображения для этих навыков можно найти в папке `icons` (`icon_rect_sword.png`, `icon_rect_freeze.png` и `icon_rect_chain.png`). Добавьте эти узлы в дерево навыков.
4. **Абстрактное дерево навыков** – создайте абстрактный класс дерева навыков `SkillTree`. Этот класс должен описывать всё дерево навыков, а не один узел. Соответственно, объекты этого класса должны содержать указатель на корень дерева. Также они должны хранить количество свободных очков навыка. При активации какого-либо навыка количество свободных очков навыка должно уменьшаться на 1. При деактивации навыка, все свободные очки должны возвращаться. Если свободных очков нет, то получить новый навык нельзя. Количество свободных очков навыка должно также отображаться на экране.
5. **Деревья навыков** Создайте классы `MageSkillTree`, `WarriorSkillTree` и `RogueSkillTree`, наследники абстрактного класса `SkillTree`. Эти классы должны создавать деревья навыков для персонажей воинов, магов и разбойников соответственно. Все деревья должны содержать как узлы типа `HitNode` так и узлы типа `AccumulativeNode`.