**Lab 2**

1. **Project Description**

The project is a 2D top down arcade shooting game with an endless mode. The game supports up to 2 players. The player will be hunting down animals and will be gaining points upon killing them.

The player will progress after achieving a certain amount of points in a round and proceeds to the next. As the player progresses to a higher level, the difficulty of the level increases. The game will end when the player loses all his lives.

1. **Application Name**

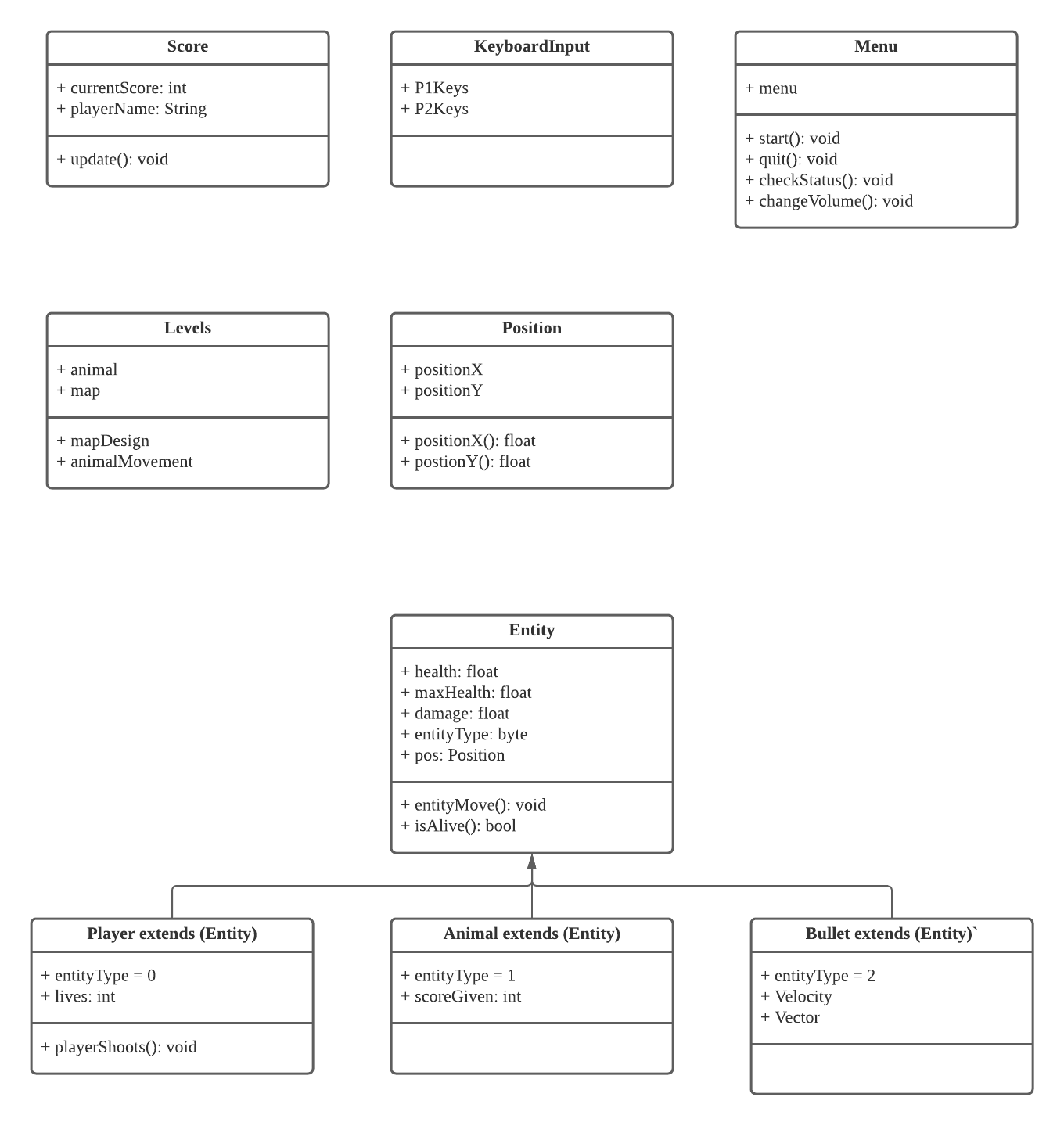
Animal Hunter 2021

1. **MoSCoW Matrix**

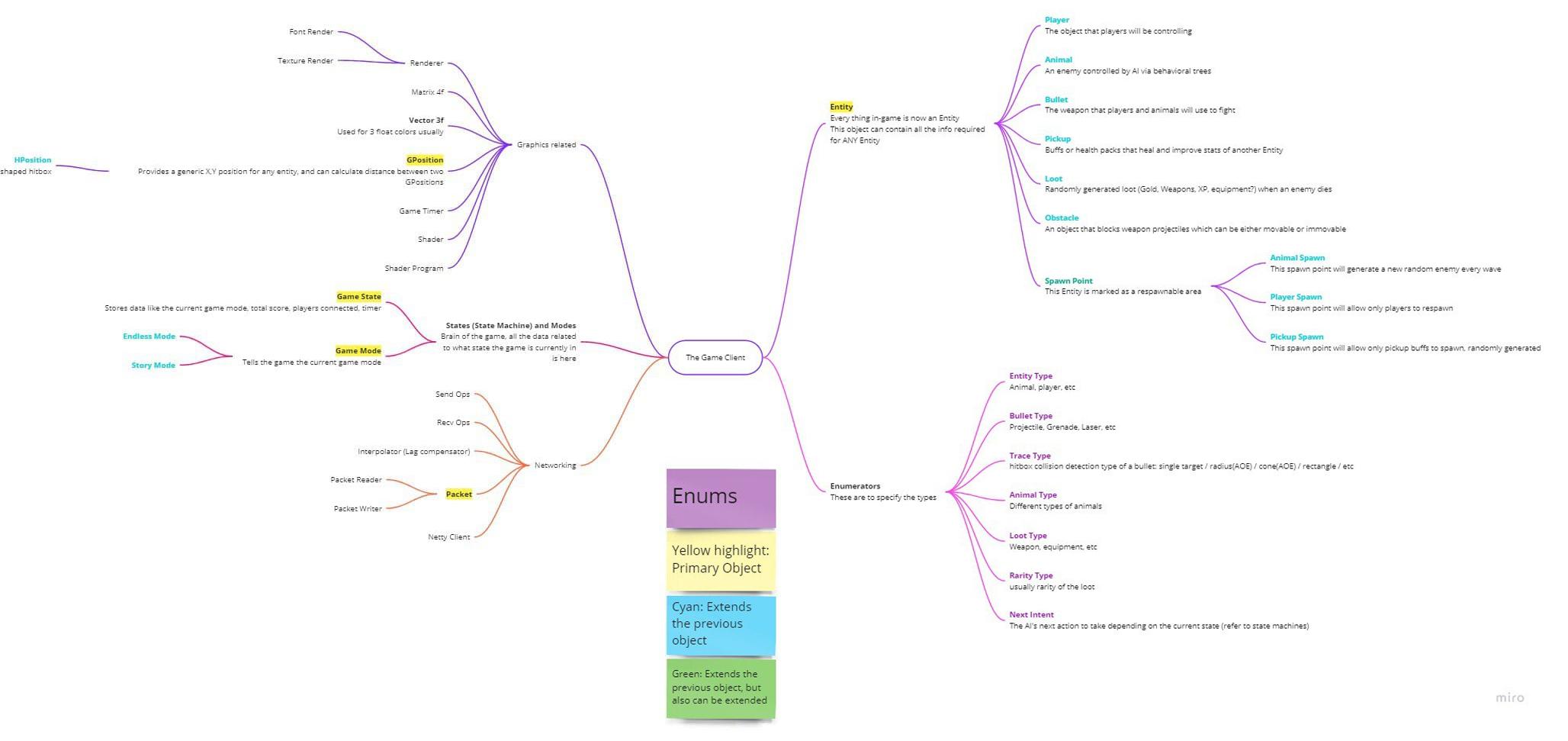
|  |  |
| --- | --- |
| **Must have**   * GUI * 2 players * Separate engines: Game engine, physics engine, maths engine etc. * Score-board * Player Lives * Visually pleasing * Report * UML Diagram * Video | **Should have**   * 2D graphics * Menu * Different levels * Background music * Settings * Keyboard controls |
| **Could have**   * Inventory * Leaderboard * Weapon upgrades * Extra lives * Power ups * Online multiplayer * TImer for each level * Randomised levels * Player statistics | **Won’t have**   * 3D graphics * Save feature * Controller support * Microtransactions * >2 players support |

**Lab 3**

1. **UML Diagram and Inheritance Relation**

****

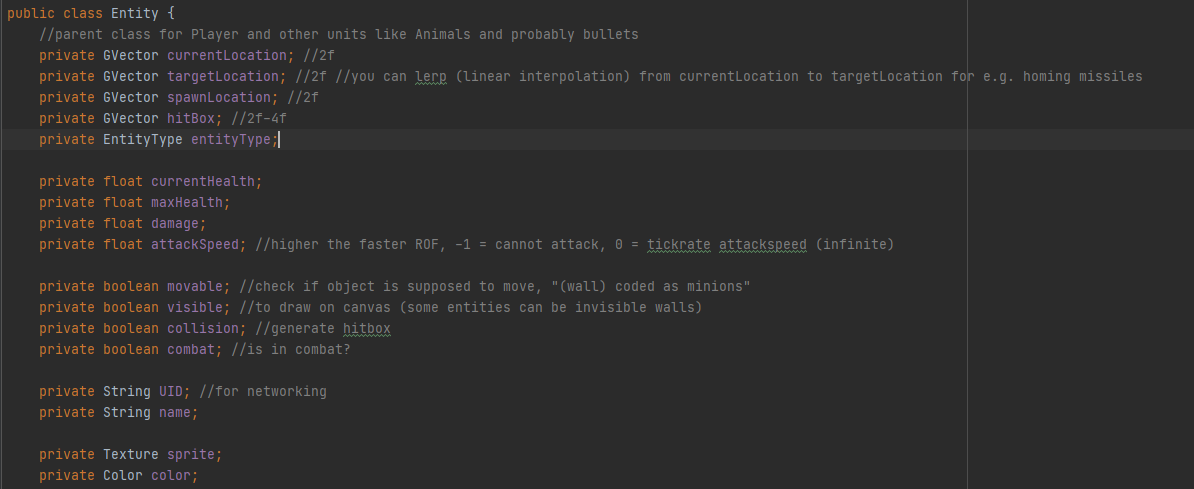
**Game Client Components**

****

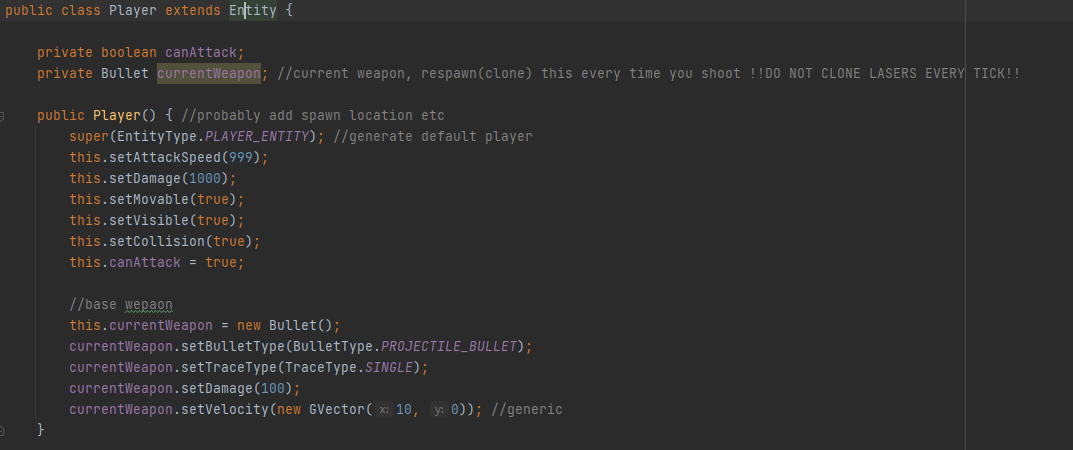
**Lab 4**

1. **Polymorphism**

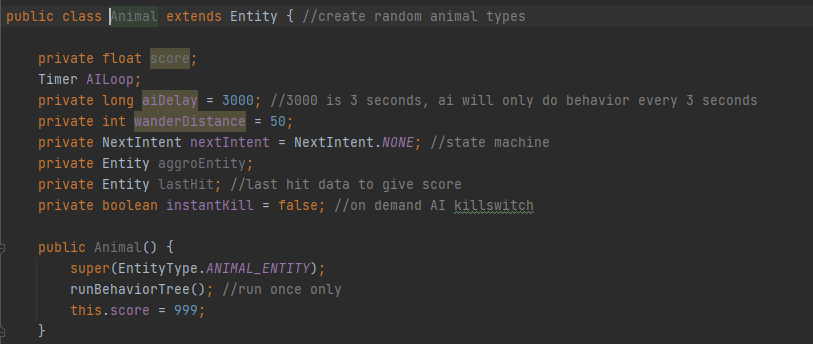
* Present at least two examples employing polymorphism in your game
* Entity (Base Class)



* Players extends Entity

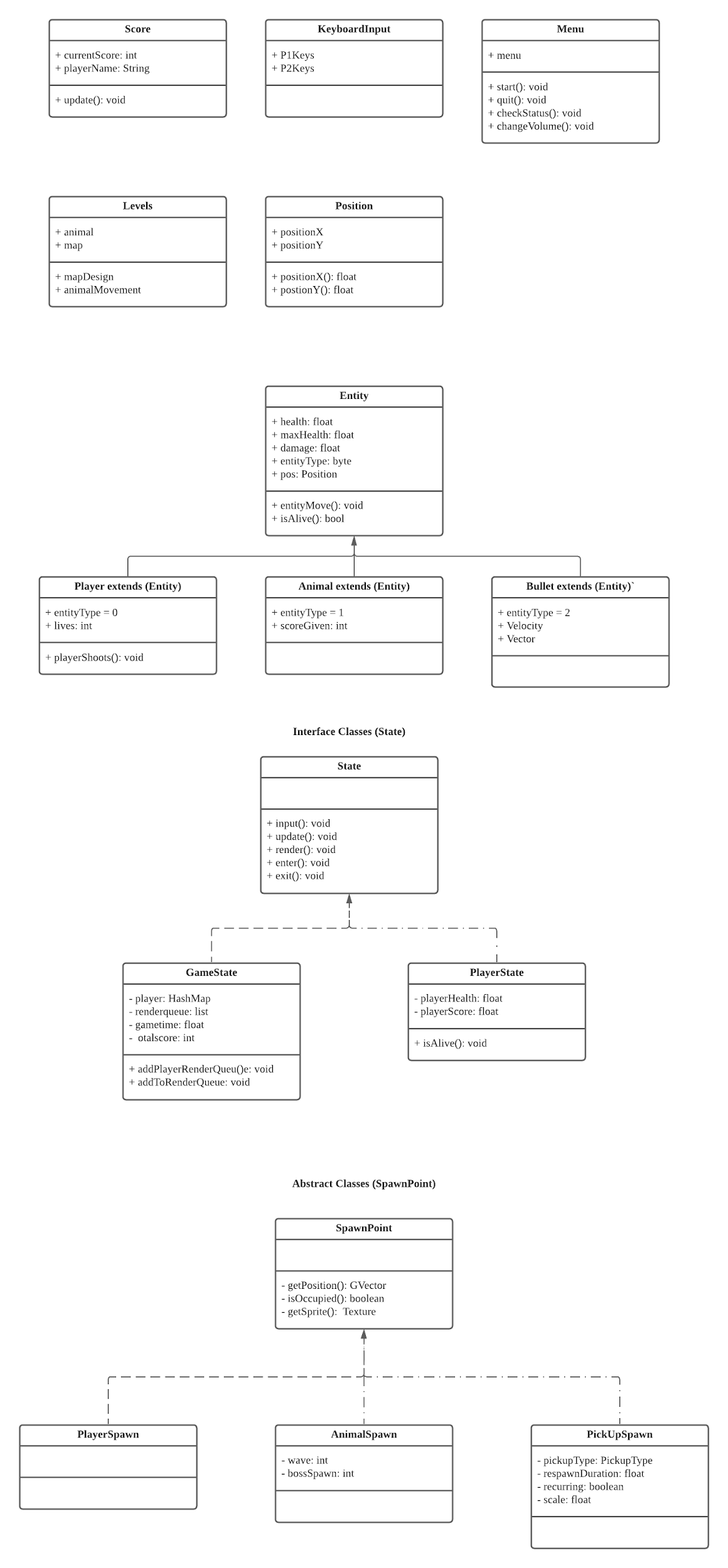


* Animals extends Entity



1. **Abstract Classes**

* In the class UML that your team created last week, mention and add abstract classes to it



1. **Interfaces**

* Present at least two examples employing interfaces in your game
* State (Base interface)



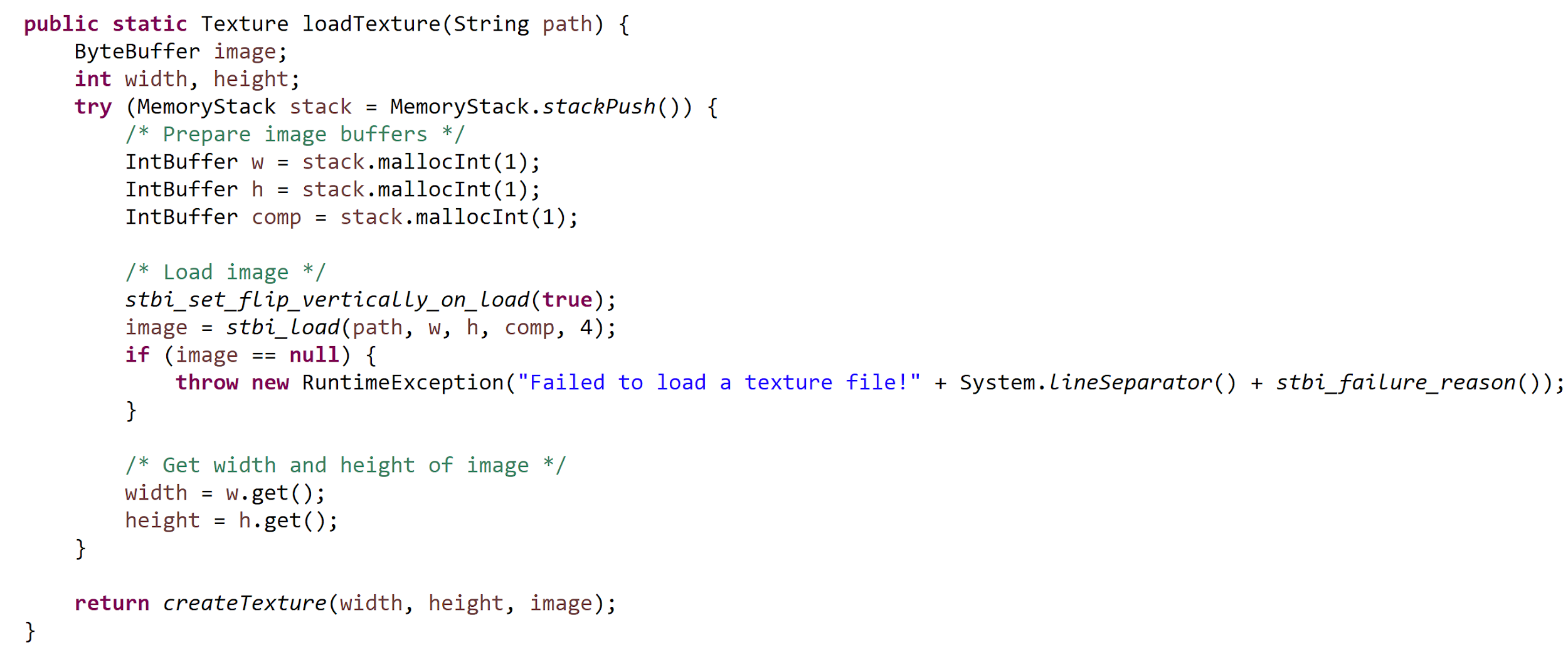
* GameState (Implemented interface)



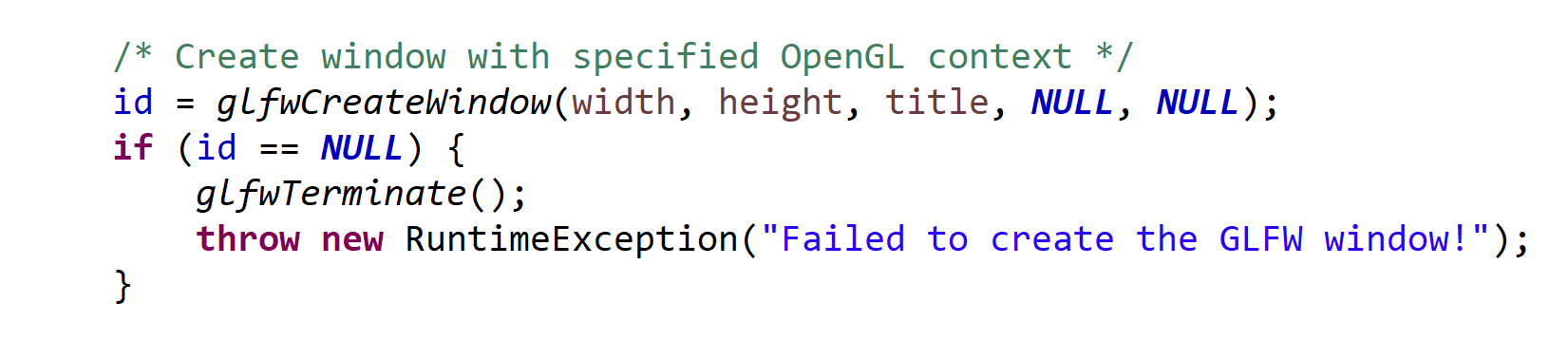
**Lab 5**

1. **Exception Handling**

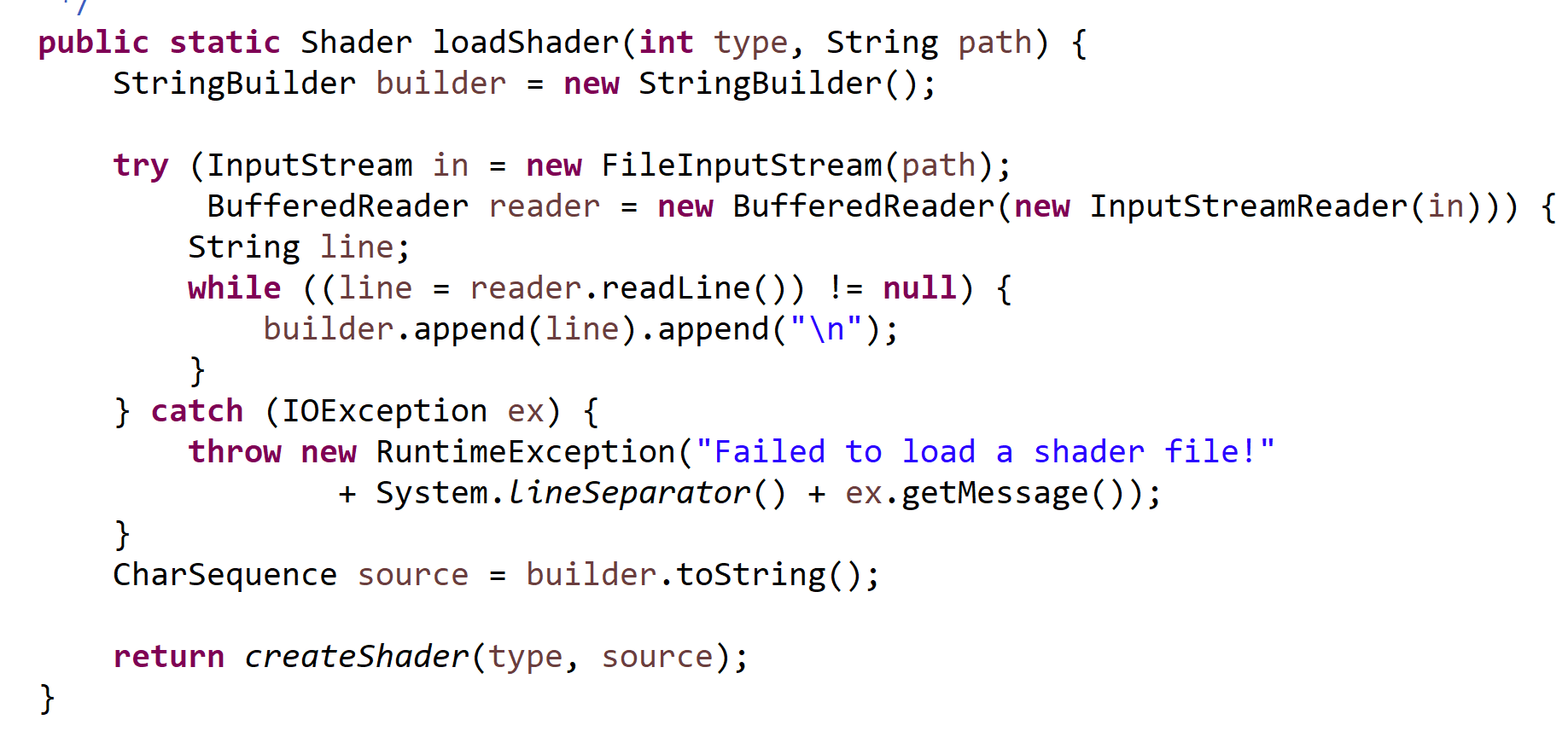
* Give three examples (classes) where you will be incorporating Exception Handling.
* Explain why it is vital for handling exceptions and those locations.
* Example 1



* It is important to handle a RuntimeException here to ensure that the texture has been loaded properly.
* Example 2



* It is important to handle a RuntimeException here to ensure that a window has been created.
* Example 3



* It is important to handle an IOException here to ensure that the shader is loaded correctly.