HighGround class created and extended by *WarpPipe* class, and it allows the player to jump onto it. To follow the Don't Repeat Yourself Principle, all the implementation that relates to the jump action will be implemented inside the *HighGround* class. By doing this, it will also be easy for maintenance or extension if there is another high ground object added into this game.

HighGround---<<creates>>---> JumpAction. In this game, there are different high ground objects that allow the player to perform jump action on them. Based on object-oriented, the high ground is the object. We can simply create JumpAction inside the Player class, however by doing this, we need to know which object the player wants to jump, it will require additional dependency between Player and different high ground. Besides, we also need to check whether the ground allows the player to jump onto it or not, checking the object classes using the if-else statement will also increase dependency. To align our design with the Reduce Dependency Principle, we discard this alternative. *HighGround---<creates>---> JumpAction* can avoid checking the objects that the player wants to jump using the if-else statement.

MemoryManager has two roles. One is to remember the last location that player on the current map before moving to the lava zone. So that when player wants to move back to safe zone, it can use the memory to bring him back to the last location that player on the safe map before moving to the lava zone. Another role is to remember the game map that has added into the system.