PIC 10B Discussion
Week 4

Professor: Michael Lindstrom
TA: Thomas Tu
Exercise: Inheritance and Polymorphism

• Let’s make a game (again)
• GameObject: parent class
  • Has a position(private): stored as vector<int>
  • Has a virtual void update() method: override it
  • Has a private const char* art: defaults to " "
  • Weak_ptr to Board it is stored in(so you can check for collisions)
• Rock: is a GameObject
  • Art: "*"
  • Update(): does nothing
Exercise: Inheritance and Polymorphism

• Board
  • hasObj(int x, int y): returns if there is an object at that location
  • Update(): updates all GameObjects

• Player: is a GameObject
  • Update(): get an up, down, left, or right from cin, and change position accordingly
  • Make sure to check for rocks/other objects
  • Assign art for Player to be “P”
Casting Errors

• What happens when you cast incorrectly?
• static_cast: no runtime check, you must be right
  • Typically, used for void*
  • Void*: A pointer that can point to anything (for better or for worse)
• dynamic_cast: checks for downcasting, sidecasting, slicing
  • On a pointer: returns null pointer
  • On a reference: throws bad_cast error
Figuring Out Types

• RTTI: Run Time Type Information
  • Find out the type of an object at run time
  • Dynamic cast uses this

• typeid(UnidentifiedObject).name()

• Dynamic behavior: gets most derived type
  • e.g. object of type Base&, but is actually a Derived
  • Will print Derived
Forcing Types to Work

• I have Apples, Bananas, and Clowns, and really want to put them into the same container
• Unfortunately, it’s someone else’s code, I can’t just give them all a new parent class

• static_cast<void*>(obj_ptr)
• static_cast<type>(void_ptr)  //that hopefully points to the right type
• Undefined behavior if you mess up
  • Don’t mess up