

Code Smell #1: The update method in the *Player* class was too long and did way too many different tasks such as updating the player's direction, sprite, position on the map as well as checking collision between players and other objects and entities.

Solution: We divided up the method into smaller chunks and created multiple helper functions that would be called by the *update* method in *Player*. This resulted in code that was significantly cleaner and easier to read and understand.

Commit: [39cf607227950df13216dbfad22c655be59453e0](#)

Code Smell #2: The player and egg interaction was implemented inside the *objectInteraction* method inside of the *Player* class instead of *Obj_Egg*, making our code less cohesive.

Solution: We moved the code from *objectInteraction* method from the *Player* class to the *Obj_Egg* class by overriding the *update* method and rewrote parts of that function to work inside the egg object. However, since *update* was already being used to decrement the egg's expiry timer, we had to move that code into a new method *updateEggTimer* as well as updating our *Map* class to use the new method. This led to more cohesive and cleaner code.

Commit: [41a79c688a7b0907039cf2662ae8868a4405299b](#)

Code Smell #3: The player and key interaction was implemented inside the *objectInteraction* method inside of the *Player* class instead of *Obj_Key*, making our code less cohesive.

Solution: We moved the code from the *objectInteraction* method from the *Player* class to the *Obj_Key* class by overriding the *update* method and rewrote parts of it to work inside that object. Resulting in the *Player* and *Obj_Key* classes being more cohesive.

Commit: [b44ff71011ac1c86a6982fd02819cfab0f5d61e4](#)

Code Smell #4: The player and trap interaction was implemented inside the *objectInteraction* method inside of the *Player* class instead of *Obj_Trap*, making our code less cohesive.

Solution: We moved the code from the *objectInteraction* method from the *Player* class to *OBJ_Trap* by overriding the *update* method and rewrote parts of it to work inside that object. Resulting in the *Player* and *OBJ_Trap* classes being more cohesive.

Commit: [ab4baa81c5046519f48431eef5818f628828b503](#)

Code Smell #5: The *objectInteraction* method in the *Player* class used a switch case to determine the type of interaction with an object. This is unnecessary as collisions with every object call the same *update* method, except for *OBJ_Gate*.

Solution: We replaced the switch case with a simple if statement that checked if the object is a gate or not. If it was, it would call its own *update* method. Otherwise, for objects that weren't gates such as traps, eggs and keys, it would also call their respective *update* methods but they would be set to *null* as well to remove the object from the game world. Consequently, our code for the *objectInteraction* method is much cleaner and easier to read.

Commit: [f7abb169806c716521a7dfa92ca71220011e67ab](#)

Code Smell #6: We had several unused files and folders inside our game's source directory.

Solution: We deleted the unused files and folders which included temporary game screens, a template for our levels and the zip folder for the tile editor we used. This resulted in our file system being less cluttered and easier to navigate.

Commit: [9463e7e0069fe40cff2420d99ceb7caab6b727af](#)

Code Smell #7: Non-functional gates, *OBJ_NonFuncGate* and *OBJ_NonFuncStoneGate*, were objects that extended functioning gates, *OBJ_Gate* and *OBJ_StoneGate* respectively. These extra classes make our class hierarchy more confusing.

Solution: We got rid of the non-functional gate classes and replaced them with tiles instead. To accomplish this, we had to create new tiles with non-functioning gates on them and update each level to use the tiles instead of objects. The *createObject* method in *ObjectManager* also had to be updated to remove the unused non-functional objects. This led to our class hierarchy being simpler.

Commit: [cc7178cfda943aa7af5d5d1c4c07c51772dd96fe](#)