Assignment One Report

Zhijun Yang

Faculty of Business and
Information Technology (FBIT)
Ontario Tech University (UOIT)
Oshawa, Canada
zhijun.yang@ontariotechu.net

Team LOGO

Junneng He

Faculty of Business and
Information Technology (FBIT)
Ontario Tech. University
(UOIT) Oshawa, Canada
Junneng.He@ontariotechu.net

I. Control

W key: Move front.
S Key: Move back.
A Key: Move left.
D Key: Move right.
Z Key: Save current position.
X Key: Load object to saved position.
C Key: Undo function.
Q Key: Create Object.
E Key: Create Object.
B Key: Change objects' color to black.
N Key: Change objects' color to blue.
Left-mouse key: Spawn ball.

II. Table of contents for each outlined section with references to each team member's contributions

Zhijun Yang: Command Design Pattern, Factory Design Pattern, DLL(From individual assignment 1), Scene.

Junneng He: Two Optional Design Pattern, Command Design Pattern UML, Report.

III. Describe the math, algorithms and logic behind the implementation.

A. Math:

According to the position of object, we made function for move speed as 0.5f. When player plays the game, object will move to what direction player selects by 0.5f of speed. Factory Design pattern just create an object and create an empty object as launcher inside the main object, when player presses Q or E, the object will be created on launcher's position, objects have two types, sphere and capsule, according to what key player pressed to create.

B. Algorithms:

Command design pattern: Creating controller function to call each function. When move function is called, it will save all commands and do a negative direction function for undo function. When undo function is called, the saved command list will change, last one will change to previous command, and object will go back last command.

Factory design pattern: create a factory class called object, and two child class inherit main class. Each created object will response in console and show what object is created.

One Optional Design pattern & Bonus:

The Observer pattern defines a one-to-many dependency between objects. When an object's state changes, all objects that depend on it are notified and automatically updated.

Singleton pattern involves a single class that is responsible for creating your own objects while ensuring that only a single object is created. This class provides a way to access its unique object, which can be accessed directly without the need to instantiate an object of that class

C. Logic:

Player press button to call function. Move functions for move left, right, back, and front by 1.0f of speed, and command list will save these commands for undo function. When undo function is called, object will read previous command and go back to previous position that object moved, and the command list will remove last one command. Factory pattern creates a main class and when player create object, it responses what object will create in scene. Observe design pattern create a class and control all object which bases on this class. Singleton design pattern create a class and only one object can be controlled by its class.

Reference for code.

"singleton leering," singleton patterns learning. [Online]. Available: http://www.runoob.com/design-pattern/singleton-pattern.html.

"Game programming patterns in Unity with C#," Habrador. [Online]. Available: https://www.habrador.com/tutorials/programming-patterns/1-command-pattern.

G. Dev, "Learn to Program with C# - SINGLETON DESIGN - Intermediate Unity Tutorial," YouTube, 17-Apr-2015. [Online]. Available: https://www.youtube.com/watch?v=EI1KJv8owCg&t=311s.

G. Dev, "Learn to Program with C# - DELEGATE & EVENTS - Advanced Unity Tutorial," YouTube, 21-Dec-2016. [Online]. Available: https://www.youtube.com/watch?v=qwQ16sS8FSs&t=31s