It is really the most challenge project at this semester, since almost all the definition, the package, the method etc. are totally new to me. Professor suggested me to use tkinter to create my pinball game, but actually at first I didn’t even know what it is. So I searched on Internet, which was what I did for final project check in, and knew that tkinter is a module built in Python. It doesn’t need to download anything from Internet and I can easily import it in Python. I tried to create some canvas, shapes, label and button and it worked. And it is also helpful to my pinball game, since the game must be presented on interface and enable user to see the ball and paddle.

After knowing the basic method to create the interface, my next step is to enable the ball and paddle to move. I watched some existing codes on the Internet, but to be honest, what they wrote, like class, def \_\_init\_\_ were totally wired and strange to me. My current Python knowledge was not enough for me to understand them. At that time, I believed that I should not try to write codes immediately, I should read some books and get to know the methods they used.

I read chapter 9 and 10 in Zelle’s book. Chapter 9 shows me how to build the structure to design a simulation. Before getting into the specific solution to the project, I need to have a bigger version instead of focusing on small problem and then gradually add features in it. It also works for my project. My pinball game includes one bouncing ball and one movable paddle. The ball will hit around the interface, and if it hit the paddle, user will get one point. If it hit the wall, it will bounce to another direction. But if users fail to catch it by controlling the paddle and the ball hit to the bottom, game is over and show user’s total score. That’s my big picture of my project and next I should focus on how to solve the smaller problem.

I noticed that most of the Python game programs include different class, so I read chapter 10 to get to know about how to define new class. I gradually knew about the previous unknown terms and it was the time for me to try by myself.

First I imported tkinter and create the simple interface and make it unadjustable and topmost. Then I started to define class ball. The first method should be def \_\_init\_\_() which provides initial values for the instance variables. Other than self parameter, it should also have canvas, paddle and color parameters to connect with other objects. Then I created a circle representing the ball and regulated the initial speed of it. To prevent it bounce out of the canvas, I also regulated the range of motion for it.

Next step is to def hit\_paddle method. The coordination of the ball and paddle can be used to check whether the paddle catch the ball. If the x-coordinate of ball is within the x-coordinate of the paddle and y-coordinate of ball is within the y-coordinate of the paddle, the ball is caught by the paddle and hit\_paddle will return true.

For draw() method, it regulates different situation when the ball hit different objects. When the ball hit the top, it will change to direction to positive. If it hit the bottom, hit\_bottom method will be true(it will be defined later). If the ball hit the paddle, it will bounce to other direction. If it hit the left-most or right-most, the direction will also be changed to opposite.

Next part is to define class paddle. In \_\_init\_\_ method, some parts are similar to ball’s. But the paddle need to be controllable, so it should bind with the direction keys on keyboard, which enable users to move the paddle by using left-arrow key or right-arrow key. It is quiet a difficult task for me, so searched the Internet to find the proper methods. Besides, I also need to def two new methods for turn\_left and turn\_right and set the speed of the paddle with them. For draw(), it regulates the movable range of the paddle.

Up till now, my main class is almost done. I tried to call on my methods and write the while loop to execute the program. If the hit\_bottom is false, the ball continues bouncing. If the hit\_bottom is true, the program will break.

At first, the program didn’t go well since there were some spelling and grammatical mistakes. After fixing them, the simple version of pinball game was done. Next, I started to add some new features on it.

First, there should be a scoreboard calculating the user score. Every time the ball hit the paddle, score will add 1. So I create a text box on the top right, and create class score. The initial score should be 0 and the text box will update the score every time the ball hit the paddle. I was going to add a pop-up window to show the score if the game is over, but I got an error when I try to convert the score to string.

Then I realized that the interface should have some instruction to guide users how to control the paddle. So I added two more text boxes to show the basic instructions. I also noted that the game automatically started when I click the run in Python, which is such in hurry. So I decided to enable user to start and pause the game by themselves. So I assigned the up-arrow and down-arrow to control these functions and display “game start” and “game end” texts on the top right. I also need to add self.started = Flase to prevent the game starts automatically. “game start” will show up when users click down-arrow to start the game and will disappear when game is over. “game over” will show up when users fail to catch the ball.

Besides, I also think users can change the color of ball and paddle by themselves. So I define two new functions to enable users to input their wanted color for ball and paddle. I also changed the previous parameter name ‘color’ to ‘ball\_color” and “paddle\_color”, which better distinguish different color setting. However, I encounter an error cause I forgot to add the parameter for ball.draw() inside the while loop.

Adding color options also inspired me to ask users to choose their difficulty for the game. Similarly, users can input the speed of ball and paddle to select the difficulty. But this time, I got stuck. I tried to change the speed of paddle by change the parameter of turn\_left and turn\_right function to

def turn\_left(self,evt, paddle\_speed):

self.x = -paddle\_speed

But it didn’t work and the error said I missing the paddle\_speed parameter. It is so strange since I already had the parameter. If I remove the evt parameter, the error showed that I must have the evt parameter. Then I asked my friend who suggested me to define new function named set\_speed to get the paddle speed. I defined a new method called set\_speed with a value parameter. In this way, I don’t need to add new parameter in turn\_left or turn\_right method. Then I just need to change the numerical speed to the variable ‘value’ and can control the speed of the paddle.

Now, my pinball game allows users to input any speed and color of ball and paddle. It is more flexible and interesting. I know that there are many other interactive features can be added on the game, but I don’t have enough time to learn more before the deadline.