1. A description of the control flow for the interaction of a player avatar and a bank square. Where in the code is the co-location of the two objects detected, and what happens from that point until the interaction is finished? Which functions of which objects are called and what do they do during the handling of this situation?

Every square checks if it's position coincides with the position of each player. It determines if the player has landed (wasn't here two ticks ago, was here last tick and this tick, roll is 0) or player passes (wasn't here last tick, here this tick, roll not 0) .

* Player is determined to be here or not using a function in StudentWorld that check if the x and y location is the same on the two given objects (usually itself and peach or yoshi).
* Given the player has landed, runs playerLandsAction, unique for each square type.
* Given the player is passing, runs playerPassAction, unique for each square type.

For a bank square, PlayerLandsAction asks the world for the bank balance, asks the world to reset bank balance, gives the player (accessed through the world) coins, and has the world play the sound.

For a bank square, PlayerPassAction checks if the player coins is 5 or more.

* case 5 or more: reduces coins by 5 from player, increases bank by 5.
* otherwise: takes all coins from player, gives bank same amount

the player is always accessed through a pointer returned by the student world for all situations, and then a player member function is called to change coins or other quantities.

1. A list of all functionality that you failed to finish as well as known bugs in your classes, e.g. “I didn’t implement the Vortex class.” or “My Bowser doesn’t work correctly yet so it behaves like a Boo right now.”

If a direction square is replaced with a dropping square by bowser, the direction square ceases functionality but the location does not start to operate as a fork.

Not a bug, but the Baddie class is treated as a limited version of the playerAvatarClass, and does not deal with stars, vortexes, coins, or requesting player actions, but deals with walking and uses the player functionality for that.

1. A list of assumptions you made; e.g., “It was not specified what to do in situation X, so this is what I decided to do.”

Upon game start player activates the coin square they are on (same as given game).

Game only runs with two players.

Get action can only be run once per tick.

Board is assumed to work or game functionality will break down.