**Contributions**

I am responsible for helping in the making of sprites and arts (player sprite and in-game background image sprite). I am also responsible for coding of player movements and actions.

**Sprites**

 I created the background image with microsoft paint (lacks of image editing app, not because I want to). To start it, I created a green rectangle as the foundation. then I created dark grey, light grey, and blue rectangle on the top of green rectangle. to make it more “realistic”, I created multiple yellow stripes on the dark grey rectangle that represents as a road where all the cars will be queued. The light grey rectangle will act as the place for the gas station and the blue rectangle with yellow circle on it will be the sky and the place for placing the HUD.

 For player sprite, I created it using pixilart (<https://www.pixilart.com>). For reference, it’s based on one of the renowned indonesian’s gas station attendant uniform with red as its main color. There are 3 sprites in total, 1 for standing still animation while the other 2 are for walking animation (left foot and right foot).

**Player movements and actions**

All player’s input commands are done through mouse clicks as the input media and all the input commands will control the player sprite’s movements and actions. Player’s sprite has pathfinding as its behavior. Pathfinding is one of the sprite’s behaviors in construct 2 used for finding shortest path to the target. It uses built-in algorithm to find the path.





For input commands related directly to player’s sprite, there are 3 commands in total. First, when the vehicle already has been put in the gas station (defined by TakingOrder value being true). Avatar (player’s sprite) will find path to the pre-determined coordinate (which is the vehicle) and move along the found path. IsTargeted of the targeted vehicle will be set to true to avoid having other vehicle order be taken. Second, when left mouse is clicked and the cursor is over the distillation tank. Third, when the cursor is over the trash bin. Those 2 commands have the same action as the first one, avatar will find path to the pre-determined coordinate, move along the found path, and set the IsTargeted variable of the targeted object to true to avoid conflict with the same kind of object (vehicle, distillation tank, trash bin other than targeted one).



When moving along the found path, the avatar will play walk animation which consists of 2 animation frames.



When the avatar overlaps with the vehicle at pre-determined position, it will then check whether the avatar has a fuel block in hand (determined through HandFull boolean) and the overlapped vehicle is the targeted one. It will only trigger once to avoid looping. After all the conditions are met, it will basically compare the order of the targeted vehicle with the fuel block in hand. If it matches, it will play correct order audio (“Nise”), add 200 to Money variable, and set the symbol to “right”. Otherwise, it will play wrong order audio (“Despacito”), subtract 100 from Money variable, and set the symbol to “wrong”. TotalCar value will be subtracted based on the number of vehicles that have been served (1 for each served vehicle). After done serving, the inventory of the player will be reset to “None”, HandFull boolean reset to False, the served vehicle will be destroyed, and the GasDispenser’s IsTaken value will be reset to False, indicating that another vehicle can be placed on it.



When the avatar overlaps with the distillation tank at pre-determined position, it will then check whether the avatar already has a fuel block in hand (determined through HandFull boolean) and the overlapped tank is the targeted one. After all the conditions are met, it will set the HandFull boolean to true, create a fuel block and pin it alongside the avatar, and reset the IsTargeted value of the distillation tank to False. It will then check the DistillType of the distillation tank. For instance, if the avatar overlaps with the distillation tank that produces Petrol, it will set the inventory of the avatar to “Petrol” and set the fuel block animation to the one that belongs to the Petrol.



When the avatar overlaps with the trash bin at pre-determined position, it will then check whether the avatar has a fuel block in hand (determined through HandFull boolean) and the overlapped trash bin is the targeted one. After all the conditions are met, it will set the HandFull boolean to False, set the inventory of the avatar to “None”, destroy the fuel block, reset the IsTargeted value of the trash bin to False, and subtract 25 from Money variable.