Recycle Michael

Philip Caraher B00699092

COM629 Mobile Game Development

11/03/2019

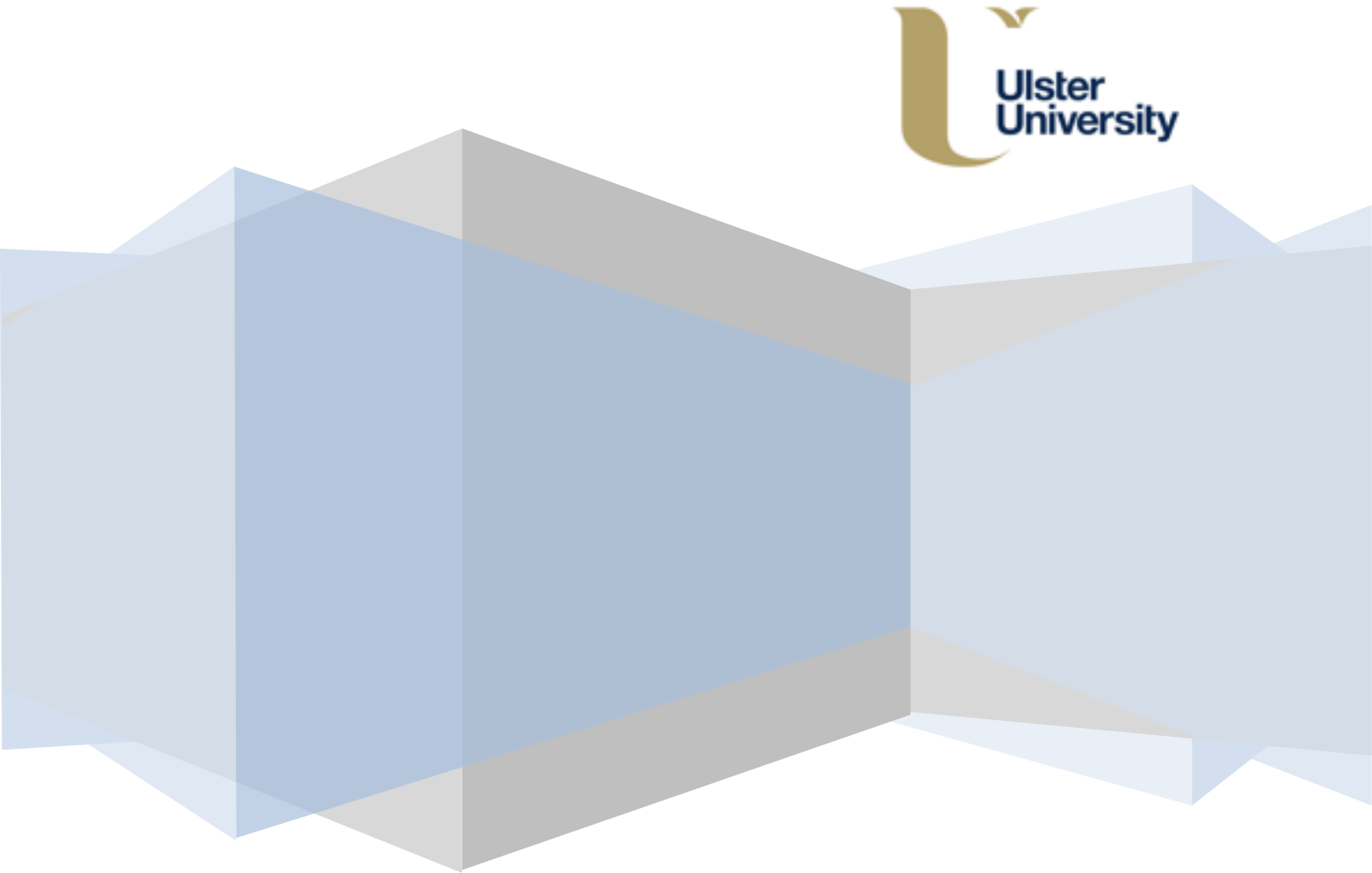


Table of Contents

Overview 3
Objective 3
Genre/ Target audience 3
Game World 3
Gameplay 4
Challenges and actions 4
Win & Lose Conditions 5
Loops 5
Monetization Strategy 5
Resources 5
In App Purchases (IAP) 6
Prototype UI design 7
Technical Design 8
Use Cases 8
Use Case Descriptions 8
Start Game 8
Select/Purchase Contract 9
Play Game 9
Sorting Facility 10
Customize Facility 11
UML Class Diagram 12

Overview

You are the manager of *Mike's Recycling* based in Capital City and there is a trash epidemic, so there's plenty of money to be made in the waste disposal business! Acquire contracts to clean up the city and earn some cash! Upgrade your facility to unlock new contracts and earn special rewards!

Objective

The objective is to sort trash into the appropriate bins to earn cash and diamonds. The player uses energy to fulfil the contracts. The trash enters the facility on conveyor belts and the player must use swipe controls to sort the trash. Each session length is dependent on the amount of trash the contract brings in. The player can upgrade their facility using the cash and diamonds earned from sorting. The upgrades allow the player to sort new types of trash; Paper, Plastic, Glass, General, Compost, etc. The player can acquire contracts that pay out more and have a higher chance of diamonds. The player needs to upgrade their facility to have access to special contracts.

Genre/ Target audience

The game is a simulation game, simulating a recycling facility in a simple cartoonish style.

Targeted towards casual gamers that like varied play session length with an underlying grind like progression. In games such as Clash of Clans, or more recent examples like Epic Seven, shows that impatience is worth money. Both have a system in which a key resource that is used to play the game is in limited supply that can be regenerated over time. The player can wait or spend money to instantly recharge it. This target audience is more willing to spend money to continue playing without waiting.

Game World

The player works as the manager at *Mike's Recycling* in a city overrun with trash. There are multiple organisations in the city from whom the player can purchase contracts. The organisation determines the type of trash in the contract and the cost of it. For example; the player has purchased a contract from a school. It cost $10 to acquire the contract and will use 10 energy to start it, and the majority of the trash in the contract will be appropriate to that found in a school i.e. paper, plastic, food, and maybe some rare trash.

Gameplay

Conveyors bring in trash into the sorting facility. They're carrying assorted trash that the player needs to swipe into the correct bins. It will test their reflexes as the conveyors won't stop, and once you've filled up a bin it must be emptied before it can be used again. There is only a certain amount of trash per contract so there are no second chances, once the trash comes through the facility if it isn't sorted it's gone.

Bins	Trash
General waste	Anything can go in the general waste bins
Paper	Regular paper, cardboard, newspaper, drink cartons, some food packaging
Plastic	Milk containers, food packaging, product packaging, drinks packaging
Glass	Drink bottles, broken glass, jars,
Food waste	Any biodegradable products, compost
Electronics	May need to be broken done into components, phones, computers, TVs, gadgets, speakers,
Clothes/Textiles	Shoes, hats, T-shirts, pants, sheets etc
Metal	Aluminium cans, Brass and bronze, Cast Iron, Copper, Steel, Tin
SAVE bin	This bin is used to save scavenged items you find so you can try to sell them for more than their recycle value.

Challenges and actions

Challenges

The primary mechanic of the game is using drag and drop touch controls to sort items based on the appearance of the item.

There is a penalty for incorrectly sorting items e.g. if the player puts a metal object into a glass recycling bin then they lose money.

The sorting bins in the facility have a base capacity that can be upgraded. During gameplay they can be filled to capacity, but the player can empty the bin during gameplay too. This doesn't affect the rewards already gained but it does make more room to fill up the bin again. The player needs to wait while the bin is being emptied before using it again ~3 seconds.

There is only a certain amount of trash in each contract so the player needs to make the best of it to maximize their earnings.

Actions

The player can buy new bins for sorting different types of trash, this opens up access to more contracts that have the same type of specialized trash e.g. when the player buys a bin for electronics, this opens up access to a range of organisations with this kind of trash.

The player can upgrade the max capacity of the bins they have.

Outside of the gameplay the player can choose what bins they want to have in their facility before starting a contract. This equip/un-equip systems comes into play later in the game as the player acquires more bins than they can fit in their facility at any one time. They have to prepare which ones to bring in before a contract.

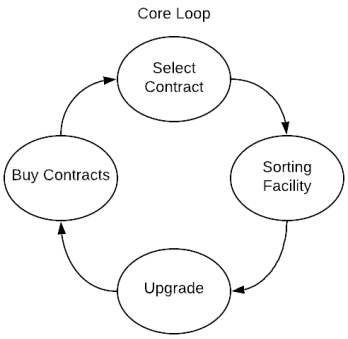
The player can save items they scavenge during sorting in order to sell later on in order to try and get more money than the recycle value. However, it takes time to sell an item i.e. charge time.

Win & Lose Conditions

There is no win condition as the player simply continues to upgrade their facility and customize. The game will get bigger and bigger as the player gains access to more parts of the city so there could be daily and weekly challenges or even quests to clean up parts of the city. There is no lose condition but there are conditions where the player will get reduced rewards.

Loops

The core loop consists of selecting a contract to bring into the sorting facility, playing the sorting game and acquiring resources, using the resources to upgrade your facility, and buy new contracts.



The Retention Loop is used to bring players back to the game and hopefully get them to spend more. Ways to gain retention in the proposed game would be daily/weekly challenges to entice players to come back for easy resources, or quests to get players to come back and grind for.

Selling saved items: Saved items can be sold for potentially more than their recycle value/worth in parts but it takes time so players would come back to collect their rewards from selling the item.

Monetization Strategy

The internal economy is inspired by Strategy, Role-Playing, and Gacha style games, such as Clash of Clans or Epic Seven, that have the player spending resources on almost every action in the game with the means to earn the resources through regular gameplay. However, the player can supplement with IAP to boost their resources.

Resources

Resources	Source	Uses
Cash	Cash is earned through playing the game. Every job that is completed rewards cash based on the amount and type of recycled materials. Daily/weekly challenges. Below are the cash values for sorted materials and the percentage chances of getting a diamond from the sorted material. Standard and Uncommon – 0.1% General waste - $1	Upgrade bin capacity. Buy contracts. Speed up selling times. Costs cash to remove bins from the facility i.e. when the player wants to replace a general waste bin with a different bin, they have to use cash to remove it (un-equip costs).

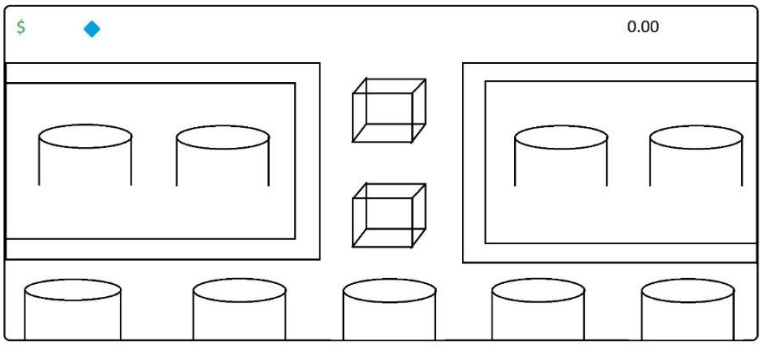
	Paper - $1.50 Metal - $1.50 Plastic - $1.50 Glass - $1.50 Food waste - $1.50 Rare – 0.2% Electronics - $2.00 Clothes/Textiles - $2.00 Special – 1% Video-Games - $5.00 Consoles - $10.00 Phones - $15.00 Relics - $20.00 Trophies - $15.00 Adult Magazine - $5.00 – 1.5% Scavenge/save The only way to get the full price from special items. Using a processor on big items gives more cash and higher chance of diamonds	
Diamonds	Can be randomly acquired from sorting regular trash – Rare. Better contracts have higher chances of giving diamonds. Using a processor gives a higher chance of diamonds.	Used to upgrade your facility e.g.; Buy bins. Different bins allow the player to sort different types of trash. Buy contracts.
Energy	Energy is a basic resource that recharges with time; 5 minutes for 1 energy. Can also be gained from daily rewards or quests.	Energy is used for sorting sessions to bring the trucks into the facility. Depending on the contract type the energy cost is different. Standard – 5 Energy Uncommon – 8 Energy Rare – 12 Energy Special – 20 Energy

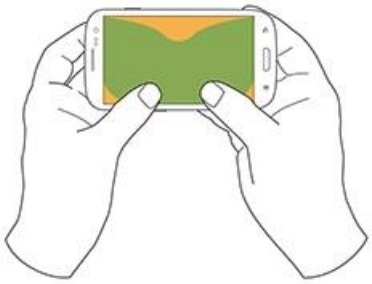
In App Purchases (IAP)

In app purchases will offer the player cash and diamond supplements (via Resource Packs) to the amount that can be earned in the game through regular gameplay. The player can also refill their energy by spending some of their cash so they can start more contracts. The first purchase offers the player the contents of a more expensive pack for a fraction of the price to make it tempting and therefore easier to commit to their first purchase.

	Cost	Contains
Pack 1 (Available once)	£0.80	$100 and 3 Diamonds
Pack 2	£2.00	$100 and 3 Diamonds
Pack 3	£4.00	$250 and 7 Diamonds
Pack 4	£8.00	$625 and 17 Diamonds
Pack 5	£16.00	$1500 and 40 Diamonds
Energy Pack	$25	Refills players energy

Prototype UI design



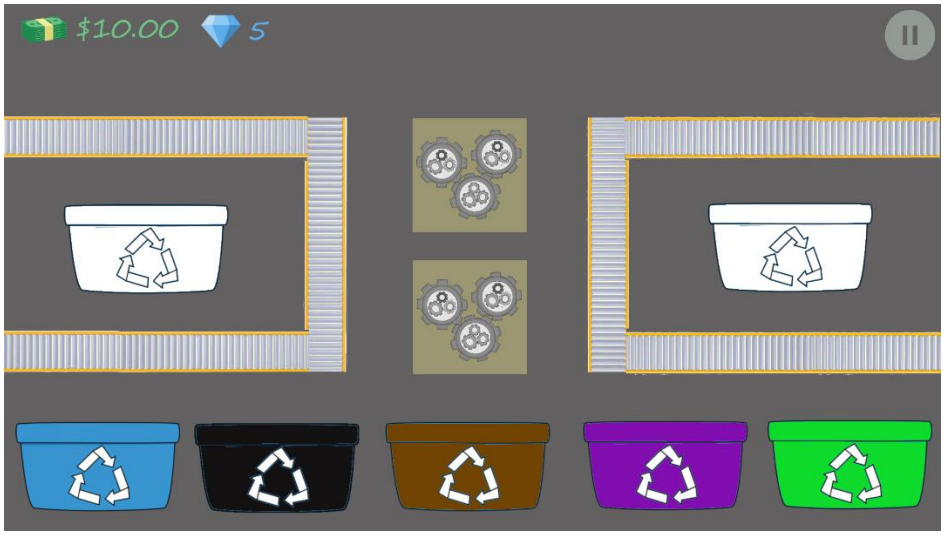


Above is a wireframe prototype UI design for the game. The conveyer belts enter from the side of the screen where the player likely be holding the device. The sorting bins can be placed in the centre of the conveyor belts or along the bottom of the screen. The spaces in-between the conveyor belts are reserved for special machines that provide a different function than the bins.

This UI design was chosen to keep in mind the radius of reachable space when the player is holding their device in the landscape orientation. During gameplay the majority of interaction will take place in the more comfortable regions.

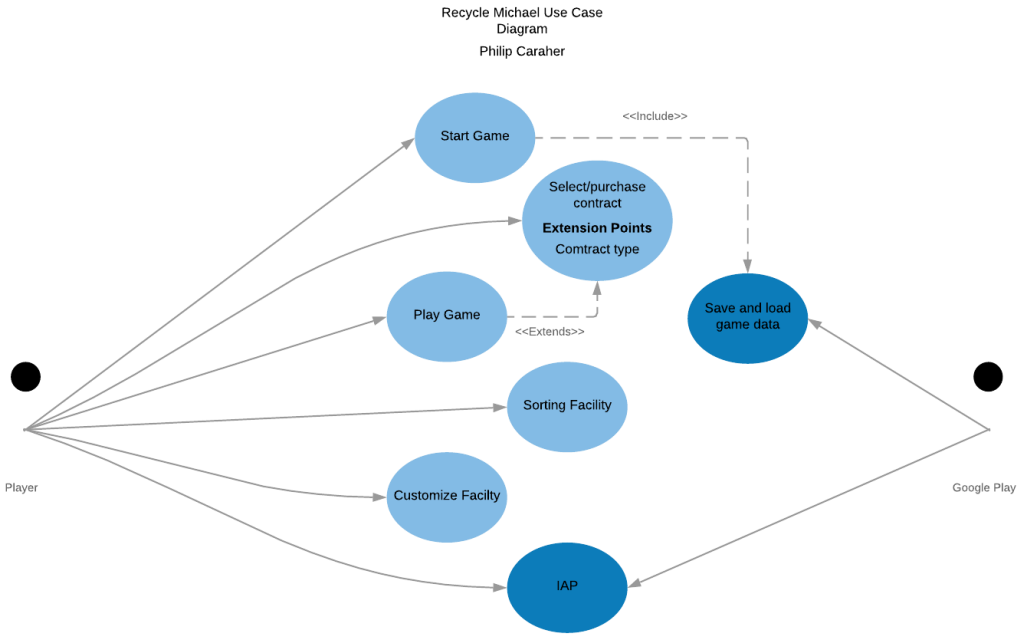
The proposed control scheme is touch based drag and drop interactions for sorting the trash into the bins. The player then has to touch the bins to empty them once they reach capacity. The placement of the bins and the conveyor belts allows the user to easily interact with the game in landscape orientation.

The UI below shows a more fleshed out look at the main gameplay screen



Technical Design

Use Cases



Use Case Descriptions

Start Game

Name: Start Game
Identifier: UC00
Description: Starting the game is the first thing the player does. It loads into the main menu screen.
Goal: To load the game and present the player their options
Preconditions: (*List the state(s) the system can be in before this use case starts)*

1. The application has been opened

Assumptions: (*List all assumptions that have been made)*

1. The player has installed the application.
2. The player has a compatible device.

Frequency: *(How often is this use case realized e.g. once a week, 500 times a day etc.)*
This use case is realized ideally multiple times per day.
Basic course: *(Describe the "normal" processing path, aka, the Happy Path)*
This use case begins when the player opens the application and ends when they load into the main menu screen.
Post conditions: *(List the state(s) the system can be in when this use case ends)*

1. The main menu UI and the players save data has been loaded

Actors: The player
Included Use Cases: *(List of use cases that this use case includes)*

1. Save and load game data by Google Play. The start game use case includes the functionality to load game data from the Save/Load use case

Extended Use Case: *(The use case, if any, that this use case extends)*
None

Select/Purchase Contract

Name: Select/Purchase Contract
Identifier: UC01
Description: The player can use their resources to buy contracts. Contracts vary in cost depending on the location in the city the contract is coming from. The player can then start this contract whenever they want using the facility they have prepared.
Goal: To provide the player with a scenario for gameplay
Preconditions: (*List the state(s) the system can be in before this use case starts)*

1. The player may have no contracts other than standard ones

Assumptions: (*List all assumptions that have been made)*

1. The player has loaded into the main menu
2. The player has resources to use

Frequency: *(How often is this use case realized e.g. once a week, 500 times a day etc.)*
Frequently. The player needs to use a contract to enter the gameplay.
Basic course: *(Describe the "normal" processing path, aka, the Happy Path)*
If the player has resources, they can spend them on contracts to vary the trash that gets delivered to the facility. Once purchased it can then be selected to take to the sorting facility and the sorting gameplay will begin based on the specifications from the contract.
The contracts define the amount of trash to be sorted, the type of trash that will be generated, and the percentage chances of rare trash.
Post conditions: *(List the state(s) the system can be in when this use case ends)*

1. The players list of purchased contracts is updated.

Actors: The player
Included Use Cases: *(List of use cases that this use case includes)*
None
Extended Use Case: *(The use case, if any, that this use case extends)*
Play Game

Play Game

Name: Play Game
Identifier: UC02
Description: Play game starts the gameplay scene using the contract information that the player selected. It loads the players facility and starts the generation of trash coming into the facility.
Goal: To load the gameplay scene using contract specifications.
Preconditions: (*List the state(s) the system can be in before this use case starts)*

1. The player has selected a contract.

Assumptions: (*List all assumptions that have been made)*

1. The players facility and all assets have been loaded.

Frequency: *(How often is this use case realized e.g. once a week, 500 times a day etc.)*
Frequently realized as the player has to use the play game case to get into the gameplay.
Basic course: *(Describe the "normal" processing path, aka, the Happy Path)*
The process of generating the trash objects to pass through the facility will use the information from the selected contract to determine the amount, types of trash, and amount of rare trash.
The trash will enter the players screen in random order on the conveyor belts and will finish once all of the trash has passed through the facility.
Post conditions: *(List the state(s) the system can be in when this use case ends)*

1. The gameplay will have started and the player will begin sorting in their facility.
2. The contract will be removed from the players contract list.

Actors: The player.
Included Use Cases: *(List of use cases that this use case includes)*
Extended Use Case: *(The use case, if any, that this use case extends)*
Select/Purchase Contract.

Sorting Facility

Name: Sorting Facility
Identifier: UC03
Description: The sorting facility is the scene where the main gameplay takes place. The player sorts the trash from the incoming contract and collects cash for the various actions they perform in the sorting process. The sorting process contains the player interaction/control scheme and the mechanics of the sorting process.
Goal: To sort the trash from the contract and earn cash/resources.
Preconditions: (*List the state(s) the system can be in before this use case starts)*

1. The player has selected a contract and chose to start it
2. The generation script has created all the trash to sort.

Assumptions: (*List all assumptions that have been made)*

1. All of the assets have been loaded successfully.

Frequency: *(How often is this use case realized e.g. once a week, 500 times a day etc.)*
Very frequently during the primary game mode as this how the player interacts with the game.
Basic course: *(Describe the "normal" processing path, aka, the Happy Path)*
The default control scheme is touch based, drag and drop. The camera is a top down view onto a 2D world so the camera is static.
As the trash comes in on the conveyor belts the player drags and drops the trash into the sorting bins. The device will check for touches and if the player is touching a trash object, they can move it and release the touch to drop the object.
Once the sorting bins are full the player touches the bin to empty it to make more room.
Post conditions: *(List the state(s) the system can be in when this use case ends)*

1. The trash count has been changed

2. The sorting bins value has been changed
3. The players cash has changed

Actors: The player
Included Use Cases: *(List of use cases that this use case includes)*
None
Extended Use Case: *(The use case, if any, that this use case extends)*
None

Customize Facility

Name: Customize Facility
Identifier: UC04
Description: The player can customize their facility by upgrading the bin capacities, buying more bins of different types, and equipping/un-equipping bins in their facility.
Goal: To upgrade the players game for more variation.
Preconditions: (*List the state(s) the system can be in before this use case starts)*

1. The player must have spendable resources.
2. Must be in the main menu

Assumptions: (*List all assumptions that have been made)*

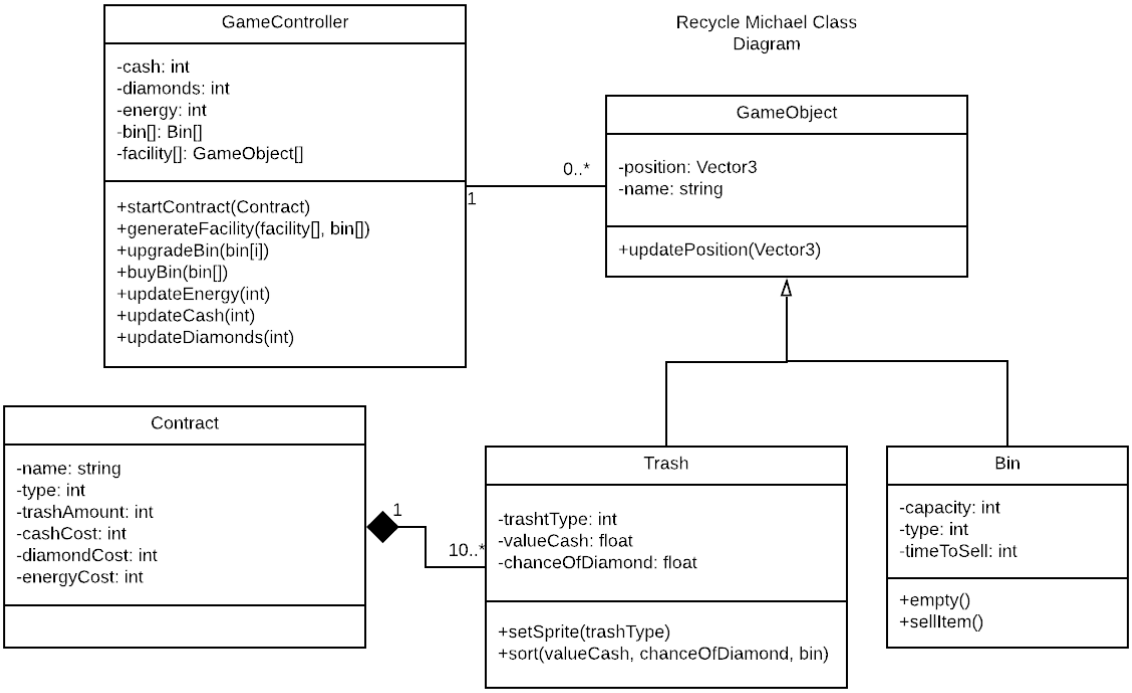
1. The player knows that they can upgrade

Frequency: *(How often is this use case realized e.g. once a week, 500 times a day etc.)*
Often. The ideal scenario is the player will manage their facility set up to maximize their earnings every contract.
Basic course: *(Describe the "normal" processing path, aka, the Happy Path)*
After the player has acquired enough resources, they can spend it on upgrades. This will all be tracked within the game using player prefs to save changes to the players bins and facility layout.
Post conditions: *(List the state(s) the system can be in when this use case ends)*

1. The players resource count has changed.
2. The properties of their own bins have changed
3. The layout of their facility has changed.

Actors: The player
Included Use Cases: *(List of use cases that this use case includes)*
None
Extended Use Case: *(The use case, if any, that this use case extends)*
None

UML Class Diagram



Above is a UML Class Diagram for the game. It shows the key classes with their attributes, functions, and their relationships with each other. As shown, the GameController will be responsible for a lot of the functionality from updating values to generating the trash that will pass through the facility.