Game Premise

City Builder is a strategy game where players take on the role of a city planner and build their own metropolis from scratch. In the game, players are given a certain amount of money to start with, which they can use to buy resources, buildings, and other assets needed to grow their city. As players progress through the game and their city grows, they earn additional points that can be used to purchase even more resources and expand their city further.

City Builder Game Documentation

Overview

City Builder is a game where players build and manage their own city. The game was built using the Windows Presentation Foundation (WPF) framework in C#, with guidance from tutor Thomas Margraff. It consists of six classes, utilizing foreach loops and List data structures.

Gameplay

To play City Builder, players start by placing buildings and assets on the game board using the buttons provided in the MainWindow. Buildings and assets cost money, which is deducted from the player's available funds. However, placing these items also generates income and boosts the population count, which can lead to more funds in the future.

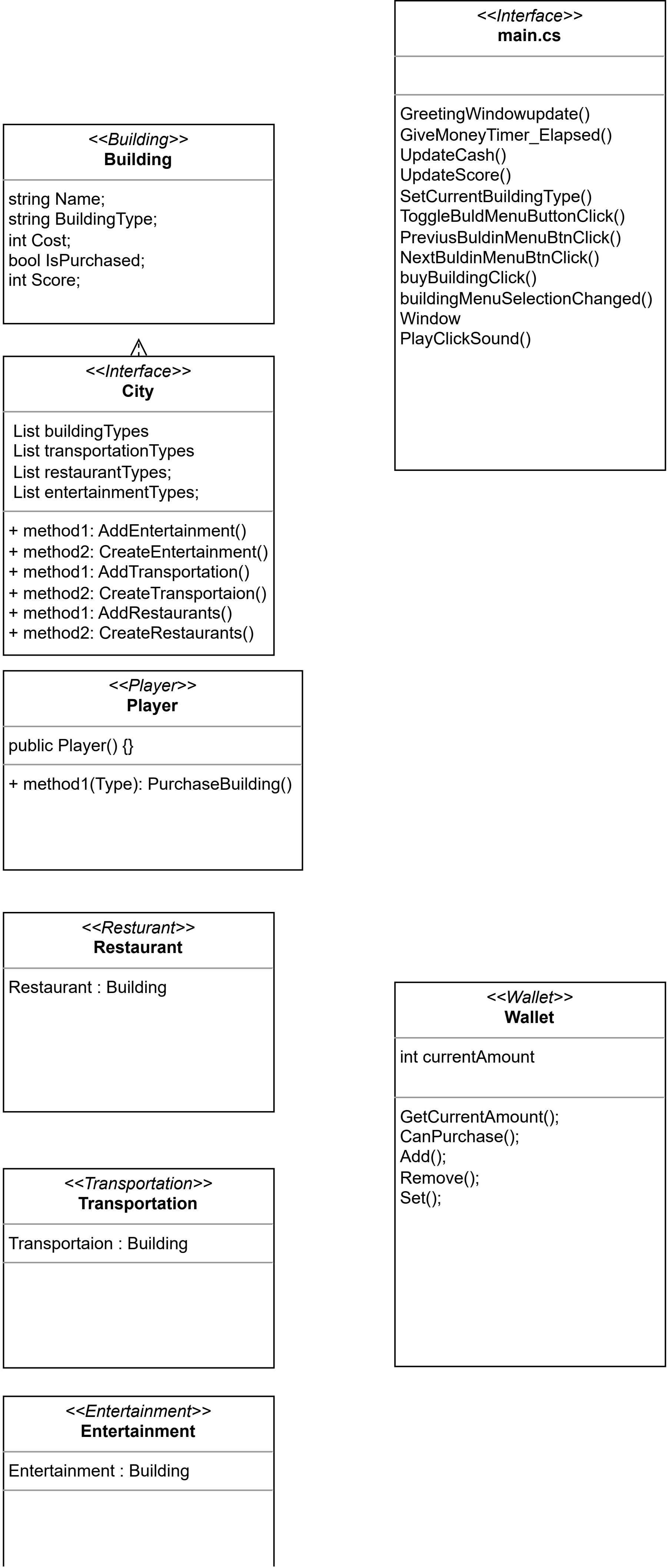
The game board is represented by a grid of cells, each of which can hold one building or asset. Players must carefully choose where to place items on the board to maximize their benefits and minimize their costs.

As the city grows, players can earn more funds and purchase additional buildings and assets. However, they must also manage their available resources carefully, as overspending can lead to bankruptcy and game over.

Conclusion

City Builder is a fun and engaging strategy game that challenges players to build and manage their own city.
With:
6 classes,
if else statements
foreach loops,
List data structures
Audio File,
Media Files

it provides a solid foundation for further exploration and development.
Thanks to Thomas Margraff for his tutoring and expertise throughout the project https://github.com/gurrenm3
He was helping me with a Timer setup, that players' money could be added every 10 minutes, he also explained many XMAL features that I didn't know about, and showed me how different I can instantiate classes and some other advanced feature in this game.



<<*Interface*>>
main.cs

GreetingWindowupdate()
GiveMoneyTimer_Elapsed()
UpdateCash()
UpdateScore()
SetCurrentBuildingType()
ToggleBuldMenuButtonClick()
PreviusBuldinMenuBtnClick()
NextBuldinMenuBtnClick()
buyBuildingClick()
buildingMenuSelectionChanged()
Window
PlayClickSound()

<<*Building*>>
Building

string Name;
string BuildingType;
int Cost;
bool IsPurchased;
int Score;

<<*Interface*>>
City

List buildingTypes
List transportationTypes
List restaurantTypes;
List entertainmentTypes;

+ method1: AddEntertainment()
+ method2: CreateEntertainment()
+ method1: AddTransportation()
+ method2: CreateTransportaion()
+ method1: AddRestaurants()
+ method2: CreateRestaurants()

<<*Player*>>
Player

public Player() {}

+ method1(Type): PurchaseBuilding()

<<*Resturant*>>
Restaurant

Restaurant : Building

<<*Wallet*>>
Wallet

int currentAmount

GetCurrentAmount();
CanPurchase();
Add();
Remove();
Set();

<<*Transportation*>>
Transportation

Transportaion : Building

<<*Entertainment*>>
Entertainment

Entertainment : Building