Game Design Document

Profit at War Time

Assignment 2

Incremental Game

Jean-Francois Retief

2458318

1. Genre and subgenre analysis

The two main inspirations I will draw from are:

- Cookie Clicker [1]
- Universal Paperclips [2].

The inspiration for many mechanics in my incremental game is Cookie Clicker, where you buy easy-to-understand upgrades and generators that increase the effectiveness of each click. I will also take cues from Universal Paperclips, where eventually the clicking mechanic becomes obsolete (since per-second generators are far more effective), and automatic generation and management of that becomes the core of the late game.

At the core of Cookie Clicker and Universal Paperclips (or any incremental game), there is simply a value (or multiple values) that increase over some increment (whether it be per-second or per-click) and there are clear examples of the mechanics mentioned in the assignment’s design requirements (such as primary currencies, generators and multipliers) that will help development of my game.

2. Hypothesis / Interrogation / Design goal

My design goal is to create a simple incremental game themed around war profiteering. The player will create weapons to sell to the warring factions either by clicking or by purchasing factories that automatically generate weapons over time (i.e. the game will increment the value either per-click or per-second).

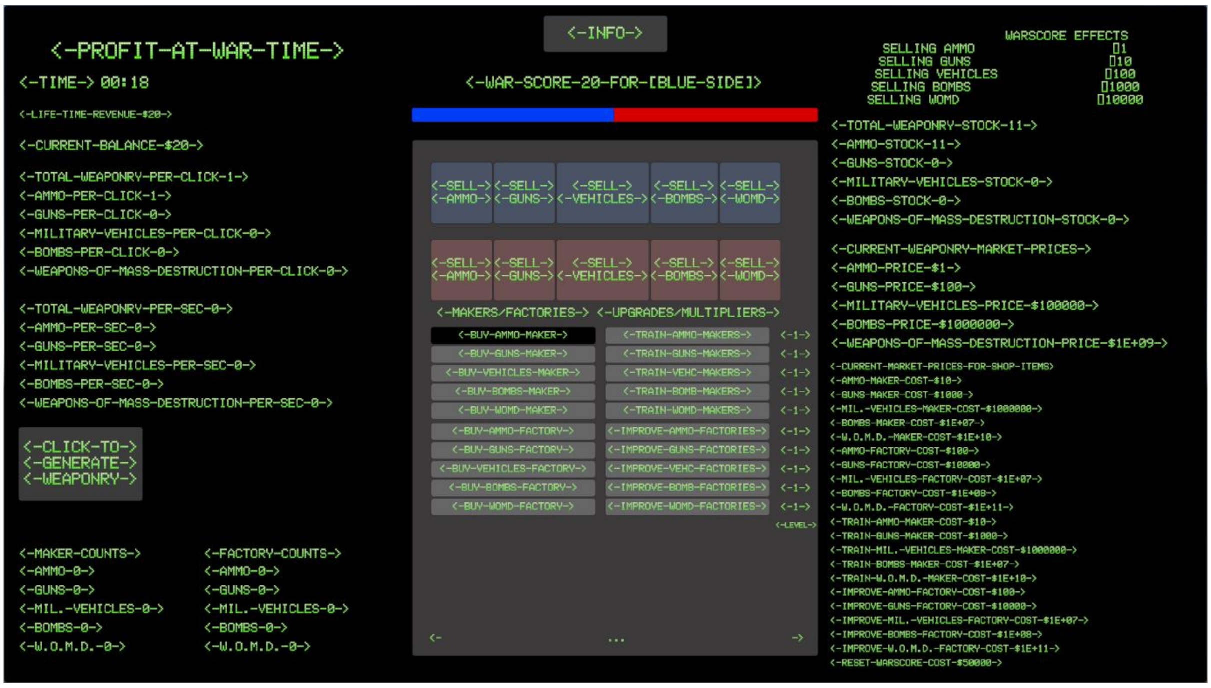
My plan for the economic progression: At the start, the player will only be able to manufacture 1 ammo per click and sell whatever they manufacture. Once they can afford upgrades, they will be able to purchase them. These upgrades will include: additional ammo per click, additional ammo per second or simple multipliers. Eventually the player will be able to afford other types of weaponry (such as guns, military vehicles, bombs and weapons of mass destruction) which will function as the five economic steps the game will have.

My plan for data management and feedback to the player: Throughout the game the following values will be kept track of: current balance, life-time revenue, current prices, the “WarScore” (a score between -1 000 000 and +1 000 000 that will be used to visually indicate which army is winning the war) and generation values per click / per sec. These values will always be shown to the player.

My plan for the core-user-experience: The player will have to balance to which side they sell weapons since their main goal is to make profit from the war, but if one side wins the war, there is no more profit to be made. This creates an important secondary goal for the player: keep the war going as long as possible, in service of their main goal (to make profit). There is also a risk-vs-reward scenario when one side is close to winning: the selling prices are increased so the player stands to make even more profit, but they run the risk of letting one side win.

3. Design notes & process

- After the initial subgenre analysis, I started an Excel spreadsheet to map out my core variables, functions, currencies, generators, relationships, etc.
- While most price increases are non-linear (NewPrice = OldPrice*2 for generators and NewPrice = OldPrice^2 for upgrades), the reset WarScore price has a linear increase. Since it prevents a game over, I wanted it to be more lenient, at least for this prototype.
- Having data-relationships and functions pre-mapped out in Excel, made the process of translating it into C# code for my unity build quite easy. In previous assignments, where data-handling and internal economies were not the focus of the project, I would usually have figured out these relationships on the fly (during the coding process), but it was helpful this time to have a pre-planned out economy ready to implement. It greatly sped up my workflow.
- After some playtesting, it was clear that reading the shop prices to see if you could buy something became tedious, very quickly (especially when hard to read floating point / scientific notation is used for large values). To remedy this I made the buttons react (turn black) to hovering your mouse over it only if the item is affordable. This allows players to just drag their mouse across the shop to check what they can buy.



```
<-PROFIT-AT-WAR-TIME->
<-TIME-> 00:18
<-LIFE-TIME-REVENUE-$20->
<-CURRENT-BALANCE-$20->

<-TOTAL-WEAPONRY-PER-CLICK-1->
<-AMMO-PER-CLICK-1->
<-GUNS-PER-CLICK-0->
<-MILITARY-VEHICLES-PER-CLICK-0->
<-BOMBS-PER-CLICK-0->
<-WEAPONS-OF-MASS-DESTRUCTION-PER-CLICK-0->

<-TOTAL-WEAPONRY-PER-SEC-0->
<-AMMO-PER-SEC-0->
<-GUNS-PER-SEC-0->
<-MILITARY-VEHICLES-PER-SEC-0->
<-BOMBS-PER-SEC-0->
<-WEAPONS-OF-MASS-DESTRUCTION-PER-SEC-0->

<-MAKER-COUNTS->          <-FACTORY-COUNTS->
<-AMMO-0->                <-AMMO-0->
<-GUNS-0->                <-GUNS-0->
<-MIL.-VEHICLES-0->       <-MIL.-VEHICLES-0->
<-BOMBS-0->               <-BOMBS-0->
<-W.O.M.D.-0->            <-W.O.M.D.-0->

<-WAR-SCORE-20-FOR-[BLUE-SIDE]>

<-1->
<-1->
<-1->
<-1->
<-1->
<-1->
<-1->
<-1->
<-1->
<-1->
<-LEVEL->

                        WARSCORE EFFECTS
      SELLING AMMO                  ▯1
     SELLING GUNS                  ▯10
    SELLING VEHICLES              ▯100
   SELLING BOMBS                 ▯1000
  SELLING WOMD                  ▯10000

<-TOTAL-WEAPONRY-STOCK-11->
<-AMMO-STOCK-11->
<-GUNS-STOCK-0->
<-MILITARY-VEHICLES-STOCK-0->
<-BOMBS-STOCK-0->
<-WEAPONS-OF-MASS-DESTRUCTION-STOCK-0->

<-CURRENT-WEAPONRY-MARKET-PRICES->
<-AMMO-PRICE-$1->
<-GUNS-PRICE-$100->
<-MILITARY-VEHICLES-PRICE-$100000->
<-BOMBS-PRICE-$1000000->
<-WEAPONS-OF-MASS-DESTRUCTION-PRICE-$1E+09->

<-CURRENT-MARKET-PRICES-FOR-SHOP-ITEMS>
<-AMMO-MAKER-COST-$10->
<-GUNS-MAKER-COST-$1000->
<-MIL.-VEHICLES-MAKER-COST-$1000000->
<-BOMBS-MAKER-COST-$1E+07->
<-W.O.M.D.-MAKER-COST-$1E+10->
<-AMMO-FACTORY-COST-$100->
<-GUNS-FACTORY-COST-$10000->
<-MIL.-VEHICLES-FACTORY-COST-$1E+07->
<-BOMBS-FACTORY-COST-$1E+08->
<-W.O.M.D.-FACTORY-COST-$1E+11->
<-TRAIN-AMMO-MAKER-COST-$10->
<-TRAIN-GUNS-MAKER-COST-$1000->
<-TRAIN-MIL.-VEHICLES-MAKER-COST-$1000000->
<-TRAIN-BOMBS-MAKER-COST-$1E+07->
<-TRAIN-W.O.M.D.-MAKER-COST-$1E+10->
<-IMPROVE-AMMO-FACTORY-COST-$100->
<-IMPROVE-GUNS-FACTORY-COST-$10000->
<-IMPROVE-MIL.-VEHICLES-FACTORY-COST-$1E+07->
<-IMPROVE-BOMBS-FACTORY-COST-$1E+08->
<-IMPROVE-W.O.M.D.-FACTORY-COST-$1E+11->
<-RESET-WARSCORE-COST-$50000->
```

Figure 1: Screenshot displaying "buy-ammo-maker" button reacting to mouse hovering over

- Something unexpected, but quite pleasing, happened during the final stages of development. The music loop I thew together and added into the game had a note playing each second, so it lines up perfectly with the timer (<-TIME-> as seen in figure 1) ticking on. While this is not intentional, it is helpful to the player. The player has the sound of their mouse clicking as feedback to the “PerClick” incrementation, but now they also have the music notes as an indictor of the “PerSec” incrementation.
- A more detailed description of the value chains within my game is in my game design tool spreadsheet, but in as few calculations as possible:
  (Note: weaponry refers to either ammo, guns, military vehicles, bombs or weapons of mass destruction)
  (Note: Square brackets denote that all within the brackets make up one variable, but the calculations are shown here with the smallest possible variables, i.e. varables that are filled in by hand in excel, or incremented in the game due to player action.)

  Weaponry per click = WeaponryMakerCount *
  [InitialMakerMultiplier * [MultiplyByValue * WeaponryMakerLevel] **]**

  Weaponry per second = WeaponryFactoryCount *
  [InitialFactoryMultiplier * [MultiplyByValue * WeaponryFactoryLevel] **]**

  WeaponryCurrentPrice = WeaponryInitialPrice *
  [CEILING(Abs(WarScore) / PriceModifierDivider)]

- The five core economic steps in my game is separated buy the types of weaponry you can generate and sell. The player starts of only making *ammo*, and the game stagnates until you make enough money (from your ammo-clicking-generation or ammo-per-second-generation and selling that ammo) to invest in the next weaponry-type: *guns*. This process repeats with all the weaponry-types until you go through all five (i.e., ammo, guns, military vehicles, bombs and finally weapons of mass destruction). The stagnation occurs since each weaponry-type’s generators is significantly more expensive than the last and their prices increase with each purchase to limit the player form buying too much.
- The time between the vehicle-step and the bomb-step was way to short at first, so I increased the bombs makers’ and factories’ initial prices.
- The primary currency is the money you make from selling the weaponry you generate.
- Thus the various exchange currencies are ammo, guns, military vehicles, bombs and weapons of mass destruction (in order of value in terms of the primary currency).
- There are two types of weaponry-generation in my game:
  -Per click generation: the amount of “makers” the player has purchased increases their weaponry-production-per-click.
  -Per second generation: the amount of “factories” the player has purchased increases their weaponry-production-per-second.
- The multipliers in the game are simple upgrades that increase the effect of each owned maker or factory. (For example: the player owns 5 bomb makers, at upgrade level 2 the player will then generate 10 bombs per click)

4. Reflection

I feel that I achieved my goal (stipulated in the Hypothesis section of this document) of creating a simple incremental game centred around the theme of war profiteering. I feel that my goals in terms of the economic progression within my game was met. The five core-economic steps were facilitated by the different exchange currencies (resources) the player has to work towards being able to generate in order to sell for money (the primary currency). I feel that my goals in terms of data management and player-feedback were also met. The player sees all necessary values they need (on screen and at any time) to make informed decisions while playing. I also feel that I achieved my goal for the core-user-experience by facilitating the balancing-act players must perform when selling weaponry to both sides of the war. All of these goals helped ground the game in the theme of war-profiteering where the player assumes the role of a greedy person taking advantage of war and the economy, without considering the ethical ramifications or loss of life (no numbers or variables on screen denoting factors such as “lives-lost” or “homes-destroyed”), all with the goal to increase the “life-time-revenue” value on the screen.

It was an interesting experience continuing working with tools such as Excel to plan out the data-relationships and value-chains beforehand and simply translating them into the game. As mentioned above in the Design Notes section, this greatly sped up my workflow, but I am still surprised by how much. Things that normal would take me a few hours to code, was done in less than one. I had a near-feature-complete prototype working much faster than I usually would have and this gave me more time to playtest and fine-tune some of the values or data-relationships.

All in all, the making of an incremental game was a great exercise for improving my understanding of games’ internal economies and it was great practice for data-handling, setting up value-chains, and core-calculations. This assignment also gave me a greater appreciation for planning variables and data-relationships ahead of time (with tools such as Excel or other game design tools) rather than my previous approach of “figure it out while coding”. Due to this, I feel that I’ve grown as a game designer in terms of planning, using tools other than the game engine, documentation and understanding the importance of internal economies.

References

[1] J. Thiennot, "Cookie Clicker," DashNet & Playsaurus (Steam release), 8 Aug 2013. [Online]. Available: https://orteil.dashnet.org/cookieclicker/.

[2] Frank Lantz & Bennett Foddy, "Universal Paperclips|," 9 Oct 2017. [Online]. Available: https://www.decisionproblem.com/paperclips/index2.html.