Alicia Cyphers
Bill O'Toole Final Raid: Design Document 12-2-2016
Tim Meehan

Name: Byte Fights

Vision

An educational card game designed to teach young programming students the basics of object oriented programming, including understanding the concepts of common methods, loops, statements, naming conventions, and the basic syntax of object oriented languages like Java.

High Level

2-4 player card game
deck of cards with numbers on one side and code snippets on the other
a crap ton of chips representing resources (RAM)
20 sided die
5 cards dealt for first turn
piece together and play code snippets from cards in hand
collect RAM chips from the resource pile
game/turns end when resource pile is fully exhausted
most points at the end wins

Baseline Story

Players are malware competing to steal the most RAM from the host in order to incapacitate the machine.

Gameplay

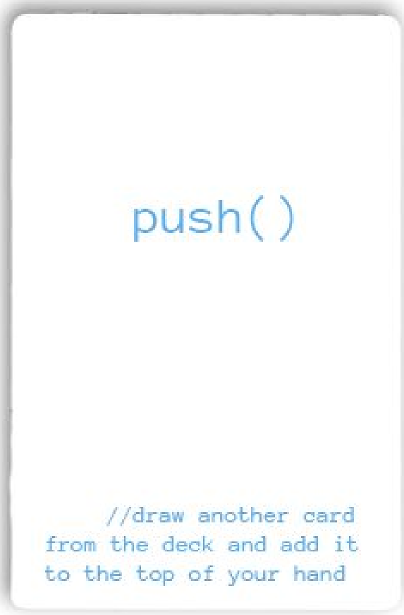
Players are dealt 5 cards each to start with. The person who deals the cards goes first. Players play as many cards/card combinations from their card hand as they can during their turn to either manipulate other players' hands, steal RAM from the computer or from other players which is represented with chips. Players each have their own player mat to keep track of their card deck size (since it can decrease or increase throughout the game) and to place/keep track of their accumulated RAM. Players cannot pick up during their turn unless they have cards that specify that they can.
For cards that require numbers to complete the action such as for loops, stealing RAM, changing the size of a player's hand, ect, the number can be found on the opposite side of the card but it cannot be looked at until after the player plays the card (so the card has some randomized component. There is also a rand.nextInt(20) which allows the player to roll a 20 sided die and use that number in place of the card value for any card except the hand size card (Arrays.copyOf(playerStack, length)) because it'd be game breaking. The game mostly uses simple randomization and methods to get players familiar with common methods and get them to think bigger about strategies to stop players from gathering RAM and to gather RAM for themself.

Visuals

Cards are color coded. For instance, methods are blue on the code side (to prevent other players from knowing its type since they can see the back) and lines of premade code are green (less common). This allows beginners to be able to tell the difference between the two (one is a method and the other is a complete line of code consisting of the object and method) before they become familiar with naming conventions and syntax. Each card also has a comment telling what each card code does.

Front of card

Back of card



push()

//draw another card from the deck and add it to the top of your hand

Touchstones

Screeps is another programming game, however, it’s complicated and can be overwhelming for beginners. We were initially considering doing something similar but thought it may be too much to ask younger kids to sit down and learn how to program (even if it was something simple) without having much experience with it.

Littlecodr is a card game that teaches even younger kids the basics of coding that gets kids to use direction cards to get to an end goal (getting them to an end point). However, this is too simple for kids 8 or older who can better grasp what programming is and already have some level of interest in it. It also teaches programming sequences in general, and doesn’t focus on helping kids with methods or concepts that are harder to grasp.

We wanted to make sure that our game would help students learn simple but commonly used methods, concepts like loops, and become familiar with object oriented programming’s syntax which many people have trouble with when they’re first introduced to it even if they already have programming experience. In this way our game teaches both general programming concepts (like loops) as well as more hard to grasp concepts like using methods on objects.