



Fast Food Wars

Final Executive Summary

Group 10

Abhishek Rathore
Jeff Grandt
William Montgomery
Zichen Wang

About the Game

Fast Food Wars is a turn-based, strategy game for 2-6 players. Players start out on a six-sided board made up of smaller hexagons. The entire board is an abstraction of a city, with the downtown in the center of the board and the suburbs around the edge of the board. This means that placing a franchise in the middle of the board will be more expensive, but there is a higher potential payoff.

Each player plays as the head of a fast food company. In the game, the player can play as a burger, pizza, fried chicken, taco, fried rice, or sandwich chain. Each player starts the game with a single franchise at the beginning of the game and a player piece. The player piece is placed on the board atop the player's franchise.

Each turn, the player can move one tile in any direction. If the player lands on an empty space, he or she can choose to build a franchise on the tile. If the player lands on a space owned by an opposing player, then the player can choose to try to put the opponent's franchise out of business and build a new franchise on the spot. The player puts up a multiplier times the original purchase price for that space. That multiplier can be from 1 to 6. A die is rolled and if the roll is less than or equal to the multiplier, then the opposing player is put out of business and the player can purchase a franchise on the space. Thus, by putting down 2 times the original purchase price, the player can expect a $\frac{1}{3}$ chance to put the opposing player out of business. Likewise, if the player puts 6 times the purchase price, the player automatically puts the other player's franchise out of business. Otherwise, the opposing player stays in business with no harm. In either case, the money is forfeit and no money goes to the opposing player, simulating money spent on advertising or other marketing activities.

At the end of each turn, the player receives a profit from the franchises he or she owns. The profit is higher on downtown locations, although the spaces are more expensive and contested. With a two player game, the strategy will likely involve buying the most efficient property in the most efficient manner. In a game with multiple players, there are several strategies that the player could take. The first would be to slowly purchase franchises in the suburbs, quietly letting the other players battle it out to swoop in and take the spots when

the time is right. Players might try to collude and attack a single opposing player together. Finally, a player might try to play very aggressively, hoping for big rewards in the downtown spaces early.

System Design

When designing the system, we tried to build a big system but with the idea that part of the system had to actually be implemented in code by following students. We envisioned the game as being played on a mobile Android device, with the a web back-end allowing for multiplayer games. We recognize that building a web service on top of a client would be out of scope for students to implement in one semester, so we put more emphasis on the design for the Android client.

Our basic architecture is a 4-tier, with the presentation layer on the Android client and the other three tiers in a hosted environment. The client will communicate with the server via http. In the case that the server needs to notify the client, the server will send a request to an existing in-house push notifications server, which will forward the request through a push notifications server in the cloud. When the client receives the message, it will send a request to the web server to sync the data.

System Implementation

We have proposed that the initial design work be completed in three phases, two development and one testing. In the first development phase, the developers will implement the basic board and allow for basic interaction including placing and moving a piece on the board. In the second phase, the developers will implement the game logic and a rudimentary AI. We feel this is a reasonable goal given the short development time. Additionally, we have included a plan for testing. Ideally, the testing phase should start before the development with some prototyping of the app. During the development phase, it is encouraged that the team use code review to create cleaner code and to ease the testing phase.

Summary

We had fun developing this game, and hope that it is apparent. We tried hard to think about a large project, while keeping a focus on a small piece that would be conducive to a software engineering course.