THE UNIVERSITY OF QUEENSLAND

Bachelor of Engineering (BE) Thesis

Technology Strategy as a Partially Adversarial Game

Student Name: Quan, TRAN

Course Code: MECH4500

Supervisor: Dr Alexander Klimenko

Submission date: 27 October 2017

A thesis submitted in partial fulfilment of the requirements of the Bachelor of Engineering (BE) degree in Mechanical and Aerospace Engineering

Faulty of Engineering, Architecture, and Information Technology

Abstract

A growing company needs a strong strategy to ensure success and allow for growth. The strategies used for technological development are vital to the infrastructure and growth of a company, but when other competitive companies become threats, there is wide debate on the recommended actions of the growing company. Competing and cooperating with competitive companies both have pros and cons, but one must be better than the other in specific scenarios. This thesis is aimed at investigating what scenarios would cooperation be more beneficial and in what scenarios would it be better to compete. To do this, the scenarios will be modelled as a partially adversarial game.

The game involves several players trying to accumulate money and Technology Points (TP). To do so, players will gain money each step and use that money to buy TP. To help the player achieve their goal, 8 strategies have been incorporated into the game: No Strategy, Higher Investments, Patents, Paid License, Free License, Joint Ventures, and Joint Venture Separation. Players will use a mix of these strategies to complete one of two motivations: economical or research. Conducting several tests on the game showed that using a higher investment strategy returns a larger money amount when only a single player is present. When multiple players were present, using a patent or a free license were the strategies that gave the highest winning rates. That patent strategy is very competitive and is commonly found in competing commercial technology companies. The free license strategy is very cooperative and is used commonly by non-profit research organisations.

A case study done on the PC industry was used to test the validity of the simulation. Unfortunately, the game was unable to accurately simulate the growth of the industry to its current position. Further investigation into artificial intelligence, game mechanics balance, and company statistic conversions will be undertaken to improve accuracy

Table of Contents

List of Figures

List of Tables

List of Acronyms

AME .... Average Money Amount

AQ .... Acquisition

AST .... Average Step Time

FL .... Free License

HI .... Higher Investments

HP .... Hewlett-Packard

IBM .... International Business Machines

IP .... Intellectual Property

JV .... Joint Venture

JVS .... Joint Venture Separation

MITS .... Micro Instrumentation Telemetry Systems

NS .... No Strategy

PA .... Patent

PC .... Personal Computer

PL .... Paid License

TP .... Tech Points

1.0 Project Outline

1.1 Background

For a company or an organisation to succeed, they require strategies that will support their current infrastructure and allow for future growth. Most companies have a strategy management plan that covers all their strategies for corporate, business, and technology sectors. The technology strategy is the objectives, tactics, and principles relating to the use and development of technology within the company (Meyer, 2006). The technology used within a company and the development of new technologies is important to the success of a company because it affects both the infrastructure of the company and potential growth.

Technological change is the first appearance of any novel product or process in a local production (Evenson & Westphal, 1994). To successfully implement and master new technology, time and money need to be invested. This can be done through cooperative research between interested parties, or individual research by each party separately. Both of these methods have their own benefits and consequences, which will be investigated during this project. Looking at cooperation, competition, and a combination of both, the best method for technological innovation will be speculated and researched.

From a global technological development standpoint, cooperation is the ideal method for growth as working together allows faster research and more thorough testing. Unfortunately, some (if not most) business companies are more interested in the economic benefits of technology development. Therefore, competition is the better method as individual research will allow them to solely reap the economic benefits of new technologies. This, and many other motivating factors that drive a certain strategy will be investigated throughout this thesis.

1.1 Objectives

Various studies have been conducted on the benefits and problems associated with competition, cooperation, or both. This thesis will investigate scenarios that provide the most potentially beneficial outcome for each strategy and more specifically, what scenarios would cooperation or competition be better as their technology strategy, with a company's economic or educational motivation being the scenarios to test. To test these scenarios, a simulation has been created that models a partially adversarial game. The game allows for changing conditions that can accurately recreate the strategies found in case studies and successfully recreating these strategies will indicate an accurate game.

1.2 Scope

To help achieve the goals mentioned above, the scope of this investigation has been defined as follows:

- Looking only common technology strategies such as investments, patents, acquisitions, paid licenses, free licenses, and joint ventures. Covert and illegal strategies such as insider trading or stealing ideas will not be considered.
- Only two main motivations drive a company in terms of technology strategy: economical and research based.
- Without artificial intelligence or a human opponent, the addition of a user controlled "main player" is not investigated in this project. The conscious thought of a human can't be recreated and so an advantage would be present in a human player.
- An accurate conversion between a company's statistics (patent history, acquisitions, sales growth, etc.) and the game strategy will not be thoroughly investigated. An accurate conversion would require a separate investigation. Thus, a rough conversion will be used during the case study simulation.

1.3 Thesis Overview

Chapter Two contains the background knowledge useful to helping understand the concepts presented in this thesis. The benefits and problems associated with competition and cooperation are presented with an in-depth analysis on game theory and the Nash Equilibrium.

Chapter Three provides an analysis on the motivations present in companies that will apply these technology strategies. Only economic and educational motivation is investigated in this project and the common strategies used by companies with these motivations are presented.

Chapter Four details the various types of technology strategy that a company could use. Looking at the benefits and drawbacks of each strategy, the strategies investigated are: No Strategy, Higher Investments, Patent, Acquisitions, Free Licensing, Paid Licensing, Joint Venture, and Alliance Separation.

Chapter Five summarises a case study conducted on the personal computer industry, looking at the life of six established computer companies: Apple, Hewlett-Packard, IBM, Acer, Lenovo, and Dell. A summary of strategy indicators is presented to show the types of strategies used by these companies.

Chapter Six presents the game that will be used to perform the tests. The game concept is outlined and the mechanics of the game is presented here. This chapter also outlines how the simulation presents the game statistics.

Chapter Seven details the investigation conducted with the simulation to test the variables within the game. The first test shows how each strategy affects the game individually, the second test shows how the strategies interact with each other and affect the game. The third test shows how each strategy compares with another in a two player game. From the data collected, a payoff matrix is created for the simulation in order to ascertain any Nash Equilibria.

Chapter Eight presents the results for the case study simulation. Hoping to recreate the current standings in the PC industry, the strategy analysis conducted on each company will be entered into the simulation. The results from the simulation will then be compared with the present-day standings of each company to test the validity of the simulation.

Chapter Nine outlines the conclusions drawn from this investigation. It will answer the questions posed in Chapter One, relay any important information discovered during this investigation, and recommendations will be made for future work

2.0 Literature Review

The basis of game theory involves solving problems in the presence of multiple people, each making their own decision (Gibbons, 1992). Within the game, each player has a motivation they wish to satisfy, this motivation influences the strategy undertaken by the player. Studying the strategies used by other players can help decide which strategy will provide the best outcome. The two main strategy types are cooperation and competition, with a third being a combination of the two.

2.1 Strategies

2.1.1 Cooperation

Cooperation involves working with the other players to achieve a shared goal. Complete cooperation between two players causes the two to act as one entity with a greater number of resources, such as money or knowledge. All players in complete cooperation lose their individuality and share the same goals. An example of a game requiring complete cooperation is escaping from an escape room. A group of people are locked inside a room or series of rooms, competing against the clock. The players are then faced, with puzzles and clues that, when solved, leads to an exit. Complete cooperation is necessary for this game as everyone has a shared goal: escaping the room before the time runs out. Players can't compete against each other in these rooms as there is only one of each puzzle and solving each puzzle helps the entire group. If the scenario were a group of people who each had their own individual room, then the game would become competitive, changing the dynamic of the escape room game.

One of the benefits of cooperation is the ability to pool local resources such as money, materials, and intellect. Cooperation allows for growth or progression through support from a second party. An example of this would be two companies merge/one takes on the other as a parent company, both companies would now have shared goals and it would be possible for resources to be allocated from either company to wherever necessary. If one of the companies needed money to research new technology for example, the other company could give them money and in exchange, make use of the new technology. In this instance, money is shared between the two companies and both receive benefits as a single entity. Rawls (1971) states that social cooperation makes a better life possible for all than if each were living solely on their own efforts.

2.1.2 Competition

Competition involves working against the other players to achieve an individual goal. Most multiplayer games are inherently competitive, especially games with only 2 players. One of the most renowned competitive two-person game is chess. Chess is purely competitive and impossible for either player to cooperate as the goal is to defeat the other player.

In *An Inquiry into the Nature and Causes of the Wealth of Nations*, Smith (1776) said “where competition is free, the rivalship of competitors, who are all endeavouring to justle one another out of employment, obliges every man to endeavour to execute his work with a certain degree of exactness”. This implies that a benefit of competition is that there is the constant necessity to do better than the opposition which consequently, improves the performance of the individual and the eventually, improves the industry standard.

2.1.3 Mixed Cooperation and Competition

Some games allow for a mix of competition and cooperation, for example, games played with teams or partial/limited cooperation. Games involving teams consists of multiple players cooperating in a competition against other groups. This type of game allows for benefits from both competitive and cooperative strategies. Teamwork in competitive situations allows for a group to collectively solve a problem whilst trying to perform better than an opposing group. An example of a game involving competitive teamwork is relay races: each member of a team runs a part of a track in succession to try and beat the other teams. It incorporates team work in the sense that each team member runs a leg of the race instead of an individual running the entire length and it is competitive where other teams are competing to get the fastest time.

Another more specific case of mixing cooperation and competition is limited cooperation that turns into competition. In this scenario, a player chooses to cooperate with another competitor for a certain amount of time or until a certain goal is achieved. Once this goal is achieved, cooperation is no longer necessary and the players may choose to return to individual strategies. An example of this strategy can be seen in some international sports. Most sports have a national league where teams compete against each other. When a team is chosen to represent their country, it is generally consisted of the best players in a specific role as opposed to the best team from that league. This means that the players that were once in competition, cooperate for the international game. They have a shared goal of winning for their nation and once they achieve this goal or the games finish, they go back to their team and go back to competing with each other.

2.2 Nash Equilibrium

In game theory, the Nash equilibrium is the stable solution of a non-cooperative scenario. The Nash equilibrium is the solution chosen by each participant when the choices of each other participant have been taken into consideration. The solution is at equilibrium because the solution is optimised in such a way that changing from that decision provides no reward (Gibbons, 1992). The most common problem that demonstrates a Nash equilibrium is *The Prisoners' Dilemma*.

2.2.1 The Prisoners' Dilemma

> "Two suspects are arrested and charged with a crime. The police lack sufficient evidence to convict the suspects, unless at least one confesses. The police hold the suspects in separate cells and explain the consequences that will follow from the actions they could take. If neither confesses then both will be convicted of a minor offense and sentenced to one month in jail. If both confess then both will be sentenced to jail for six months. Finally, if one confesses but the other does not, then the confessor will be released immediately but the other will be sentenced to nine months in jail—six for the crime and a further three for obstructing justice."

(Gibbons, 1992)

The actions in this game can be summarised into three actions and consequences:

1. Both players deny: both will receive 1 month in jail.
2. One confesses, one denies: the person who confessed is released and the person who denies receives 9 months in jail.
3. Both confess: both receive 6 months in jail.

This can also be represented by a matrix shown in Figure 1.

		Prisoner 2	
		Confess	Deny
Prisoner 1	Confess	6 , 6	0 , 9
	Deny	9 , 0	1 , 1

Figure 1: The Prisoners' Dilemma Matrix

The matrix shows the punishment for each prisoner depending on the response of the other prisoner. It can be seen that the globally optimal solution would be that both prisoners deny and both receive 1 month of jail time. This globally optimal solution does not consider the individual choices of each prisoner. For prisoner 1, their best course of action depends on the action of the other prisoner. If prisoner 2 confesses, prisoner 1 would be better off confessing as denying the claim would result in 9 months of jail. If prisoner 2 denies the claim, the best action for prisoner 1 would still be to confess as it would result in immediate release as opposed to 1 month in jail. In both hypothetical situations, prisoner 1 is better off confessing to the police. The same analysis can be conducted for prisoner 2 where confessing gives the optimal solution for themselves regardless of what prisoner 1 says. Therefore, the Nash Equilibrium is that both prisoners confess, and each receive 6 months of jail time. This is due to the fact that if either prisoner deviates from this solution, a worse punishment would result for that individual prisoner.

The equilibrium solution to the prisoners' dilemma is not the globally optimal solution due the personally beneficial solution chosen by each prisoner. The solution to this problem is the most basic demonstration of game theory where the consideration of choices made by other players affected the outcome. In this scenario, the globally optimal solution of both prisoners denying and receiving 1 month each is an unstable solution. This solution is unstable because there is a better solution for the other prisoner.

This problem has a few assumptions that help determine this equilibrium point. The problem firstly assumes that the prisoners didn't have any prior preparation to being caught. If the prisoners had organised a plan prior to getting caught, both prisoners could automatically default to denial without considerations of consequences. It also assumes the prisoners only consider their own punishment without considering the punishment of the other prisoner.

2.2.2 The Dollar Auction Game

The Dollar Auction Game is a game that is used in game theory to demonstrate a game scenario with no Nash Equilibrium. The game involves an auction to buy a one dollar bill and for simplicity, two bidders. The only rule different from a normal auction is that the second highest bidder is required to pay their bid (Shubik, 1971).

Letting the minimum bid and difference be 5 cents, Player A starts with 5 cents and Player B then bids 10 cents. Each player continues bidding as they wish to increase their gain and minimise their losses; if A bids 20 cents, they stand to gain 80 cents and B would lose 15 cents, thus B would bid higher than A to turn the loss into a gain. This would continue until the bid reaches 1 dollar, after this point, all players stand to make a loss but continue bidding to minimise the losses. If A bids 1 dollar and B then bids \$1.05, A stands to lose the dollar while B has a net loss of only 5 cents, thus if A bids \$1.10, they then only lose 10 cents whereas B loses \$1.05. This would continue until one player gives up, but the game has a potential of being infinitely long as both players could theoretically continue forever. Once bidding has started, there is no Nash equilibrium as both players would continue to bid to minimise losses.

The optimal solution to this situation would be that neither player participates in the game but that is assumed to be not a valid option. If this were possible, this solution would be unstable as changing from this strategy allows a player to receive a reward and once started, both players are trapped in the bidding war. Another assumption made during the game is the inability to communicate with the other bidders. Communication could lead to cooperative actions which defy the non-cooperative criteria of a Nash equilibrium.

2.3 Prior Literature

2.3.1 *Competition and Cooperation* by Saul Levmore (1998)

Levmore investigated the relationship between competition and cooperation and looked at their dependence on firm size. Levmore also looked into the factors that contribute to a company's decision to "make-or-buy"; to internally produce a part and consequently grow the company or source the part from a third-party producer or even the competitor. In this instance, he refers to cooperation as sharing or buying the same products. If two firms had prearranged a deal for the part production, they were considered to be engaging in explicit cooperation whereas one firm simply buying from a competitor is considered implicit cooperation. Throughout the article, Levmore uses Ford and General Motors as examples in the automotive industry.

Levmore first starts off by investigating the relationship between the size of a firm and the willingness to cooperate with competitors. He expected that for a given firm, if growth was present in the absence of increasing agency costs, the firm would internally produce necessary parts and thus, expanding the firm. This growth stems from the ability to control factors such as production rate, cost, time, and sale price as opposed to when the firm needed to buy parts externally. If Ford and General Motors were the only two companies producing a part, one company choosing to internally produce would be considered competitive as opposed to buying from the other and engaging in implicit cooperation.

Levmore next looks at the influence of the market and how it affects the inclination or disinclination to cooperate. He anticipated that even if a part produced by a competitor was the most economically viable across the market, a company would not purchase that part, that is, not engage in implicit cooperation. An example used was that Ford would not purchase brakes produced by General Motors and use them in their cars. This is not always the case though as some cross-competition supplying has happened before: an example used by Levmore is Microsoft selling to Apple.

2.3.2 *International Strategic Management* by Franklin Root and Kanoknart Visudtibhan (1992)

Looking at Part 5: Forming International Strategic Alliances, Root and Visudtibhan look at the benefits of cooperation and competition. They look at the factors that decide whether to have a cooperative arrangement of a 'wholly-owned operation'. They assume that projections have been made by a company looking into a fully-owned operation or a cooperative alternative. The projections would include the profitability, costs, and revenue. They claim that there is a preference for a Cooperative Venture (CV) opposed to a fully-owned subsidiary when the difference between the incremental benefit and cost are greater than the partner's profit share in that venture. Meaning cooperation is more favourable when the overall benefit or profit is greater than the partner's.

Root and Visudtibhan go on to list the direct and indirect benefits of a cooperative venture shown in Figure 2 and the costs that decrease with a cooperative venture shown in Figure 3.

Direct (R1)	Indirect (R2)
• Other partner's knowledge of market • Other partner's intangible assets such as technology, patents, trademark • Other partner's ties to government or important buyers • One less competitor; hence larger market share • Faster entry improves NPV • Access to market otherwise foreclosed	• More complete product line helps overall sales • Technical or new product ideas learned from other partner, diffused to other parts of company • Markups on components, or product

Figure 2: Direct and indirect benefits from a CV

Direct (C_1)	Indirect (C_2)
• Economies of scale from larger market shape • Rationalization based on each partner nation's corporative advantage • Government incentives/subsides given to CV's only • Using slack/underutilized equipment or design capabilities in each partner lowers capital cost and overheads • Fewer headquarter personnel deputed • Access through partner to cheaper inputs • More productive technology or administrative methods contributed by one partner	• Productivity and technical improvements diffused to other parts of company

Figure 3: Direct and indirect cost decreases due to a CV

They then investigate the 'costs' of cooperative ventures, looking at the possible detrimental effects of cooperation as opposed to a fully-owned investment. They speculate that a firm can be potentially constrained by its CV as it may not have the freedom to undertake investments in a new line of business. There is also a possibility that prices for end-products that were set during the collaboration are lower than they would prefer. Figure 4 lists the possible detrimental effects of a cooperative venture and Figure 5 lists the cost increases due to a cooperative venture.

Direct (R_3)	Indirect (R_4)
• CV Associated does not allow firm to expand into certain lines of business in the future • Partner reaps the benefit of future business expansion, not proportional to their future contribution • Setting lower price at behest of partner	• Partner's desire to export decreases sales made by other affiliates in international markets • Partner becomes more formidable competitor in the future

Figure 4: Possible detrimental effects from a CV

Direct (C_3)	Indirect (C_4)
• Cost of transferring technology/expertise to partner • Increased coordination/governance costs • Pressures from partner to buy from designated sources. Or sell through their distribution channel. • Global optimization of MNC partner may not be possible for — Sourcing — Financial flows — Tax — Transfer pricing — Rationalization of production	• Slight increase in headquarters administrative, legal and other overheads • Opportunity costs of executives or technicians deputed to CV

Figure 5: Cost increase due to a CV

Lastly, Root and Visudtibhan talk about the role of technology in the formation of cooperative arrangements. They conclude that firms with similar interests in research and development are more likely to cooperate. This willingness to cooperate is due to the escalating speed of technological innovation, as well as increasing numbers of technologically advanced competitors.

3.0 Strategy Motivation

Before implementing any strategy into a business, the goals and motivations of a company need to be assessed to determine what strategies will be most effective for the specific company. Economic and research are examples of motivation or goals that a company can undertake and will be investigated in this project.

3.1 Economic Motivation

Economic motivation is one that drives almost all multi-person party or company. Though it may not be an important factor for some companies, such as ones that don't sell any products, there would still be an underlying economic motivation.

Everything in the modern world requires money, including food, transport costs, and electricity in the electronic devices used by almost every person in a first world city. Companies cannot be managed and operated by a single person, so workers are required. Since workers need money to live, the company needs money to pay for workers. Most companies sell products to gain money, of which, includes the wages of its workers and profit for the owners. This is the most conventional business model and largely supports the economic motivation.

Some research companies such as ECRI Institute are non-profit organisations that are dedicated to research (ECRI Institute, 2017). These types of companies don't have a goal of earning money, but the research conducted by these companies and the wage of the workers require money, generally obtained through donations and research grants. Without money, the company would not be able to survive and hence, money is generally an underlying motivation in all companies.

A company with economic motivation would be more likely to compete with other companies as competition within a common market results in reduced sales and consequently, reduced profit. Since money is an important goal, cooperation would mean profits are shared with a different company and so, it would be more beneficial to compete alone and not share the profits.

3.2 Research Goals

A company with a research goal is one aimed at conducting experiments or inventing new methods, processes, or devices in hopes of improving the quality of life on Earth. SpaceX is an example of a company dedicated to research without economic motivation. SpaceX has designed a reusable rocket in the hopes of reducing the cost of space access by a hundredfold (SpaceX, 2017). Comparing SpaceX to Virgin Galactic, a company with the goal of "creating something new and lasting: the world's first commercial spaceline" (Virgin Galactic, 2017), it can be seen that Virgin Galactic are conducting research in order to produce a commercial service, aimed at earning money.

A company with a research orientated goal is more likely to cooperate with other companies as the end goal is something they will share with others and the world. There is no point in them competing and initiating conflict when cooperation will help them reach their goal.

4.0 Technology Strategies

To help improve the sales, growth, and infrastructure of a company, various strategies can be implemented, each with their own unique benefits and consequences.

4.1 No Investment/Strategy

This strategy, or lack thereof, is one that is found inherently in all companies and has both positive and negative effects. Using no strategy (NS) involves basing the technology change/upgrade on industry technological growth. This is analogous to a company buying the latest machine or software to use but makes no investment into expanding the technology internally. The technology of the company grows with the industry but if other companies are relying on this strategy than everyone advances at the same pace. This strategy has no extra benefit to the company but it also doesn’t cause conflict between competitors.

4.2 Higher Investments

A higher investment (HI) strategy is generally used when a company has encountered a setback or is struggling to complete a project. Investing more time and money into a project does not have a linear relationship with innovation (Hottenrott & Peters, 2009). An exact relationship has not been determined between investment and innovation but as an example: if we rate a product’s innovation out of 10, an investment of $1 million produced an product with an innovation level of 5, an investment of $2 million in the same project would only have an innovation level of 8. Putting all of a company’s money into a project is only something they would do as a backup plan for a quick but expensive investment.

4.3 Patent

A patent (PA) is a protection system for new methods, processes, substances, or inventions, giving legally enforced rights to the patented object and its owner (Australian Government, 2016). A patent prevents others from using, selling, and/or manufacturing the intellectual property (IP) without the owner's permission. When a patent has been successfully granted, the development of rival inventors/companies is affected significantly and if the patent is effectively protected, it will provide a competitive advantage to the owner (Ernst, 2003).

In *Managing Intellectual Capital*, Grindley and Teece (1997) stated that within the semiconductor industry, industry participants found it increasingly necessary to seek licensing due to patent protection. This means that the participant must either pay royalties to the IP owner or develop their own invention, both of which negatively affect the growth of the participant. As mentioned in Section 2.1.2, the competitive nature of the patent and the negative effects on the competition stimulates innovation as the competition must find new approaches to the problem, in fear of money or job loss.

The patent is beneficial to a company with a very competitive goal such as an economic motivation. It allows for control of the market but at a high cost, required for obtaining and maintain a patent.

4.4 Acquisition

An acquisition (AQ) is when a company (acquirer) wishes to gain control of another company (target) in either a merger or hostile takeover (Investopedia, 2017). A merger is a cooperative acquisition where both parties expect positive outcomes from the merger. For example, a more financially secure company can acquire a small company in debt to take on the debt while the small company provides the acquirer with profits over time. A hostile takeover is more aggressive and undesired by the target company. To perform a hostile takeover, the acquiring company will try and purchase the target company by placing an offer to the shareholders to gain control of the target. Once a takeover has been successfully completed, the target becomes a subsidiary of the acquirer, to either increase sales/profit of the acquirer or to just remove a competitor in the industry.

An example of a hostile acquisition is the Vodafone – Mannesmann takeover in 1999 (Deutsche Welle, 2010) between two of the largest telecommunications companies in Europe. In November 1999, U.K based Vodafone presented an offer of approximately 100 billion euros to the Mannesmann CEO, Klaus Esser. Esser refused the offer which led to Vodafone CEO Chris Ghent to repeated increase his offer, which was declined by Esser each time, eventually leading Ghent to approaching Mannesman shareholders directly. After Vodafone announced mergers with various other companies, investors and shareholders in Mannesmann began urging Esser to accept the deal by Ghent. In February 2000, the takeover was approved by the Mannesmann supervisory board in a 190 billion euro deal, the largest takeover bid at the time (Deutsche Welle, 2010). Today, Mannesmann is non-existent in the telecommunications industry whereas Vodafone is ranked 19th in the global telecommunications industry by Forbes (2017).

An acquisition would more likely be done by a competitive company looking to dominate a market, by removing the competition and gaining all the potential profit/clients and resources owned by the target company. A takeover is beneficial to a company by removing the competition but at a very high cost, one that may not be returned in sales.

4.5 Licensing

In technology strategy, licensing is the agreement or contract regarding the conditions permitted to the user for an IP made by the owner. In this investigation, two forms of licensing will be investigated, paid licensing and free licensing.

4.5.1 Free Licensing

By giving free license of IP, a company offers unrestricted use of the product to anyone for free. Although the company does not gain any direct profit, they do grow in terms of technological development due to feedback from the licensee (Kolk et al., 2015). Due to the product being free, the number of licenses distributed is quite high and consequently, allows for fast growth. It is beneficial to cooperative companies with a research oriented goal due to the wide range of feedback given by product users. Such strategy is not commonly used by competitive companies as they receive little to no return from investing in a strategy.

4.5.2 Paid Licensing

Similarly, paid licensing provides technological development through licensee feedback, but it also provides an income for the IP owner. As licensees must pay for the product, the paid license will generally have less sales than a free license. Irrespective of the price, paid licenses have shown to be very effective in the growth of a company. Using the example of Microsoft: Bill Gates and Paul Allen developed MS-DOS, an operating system for the IBM-PC (Swaine & Freiberger, 2014). The operating system that they built required a paid license, paid by the computer hardware manufacturer and given to the consumer preinstalled in the computer, meaning the consumer essentially purchased a license from the manufacturer. Microsoft convinced IBM of two things, to allow the operating system to have an "open architecture" (allowing programmers to develop programs freely for the operating system) and allow Microsoft to distribute the operating system to other PC companies. The open architecture was one of the main success factors in the IBM-PC and consequently, higher sales in IBM-PC's meant more royalties to Microsoft. Being able to distribute their product to other companies as well, Microsoft now have the largest user base for the Windows Operating System.

Paid licensing is a mixture of both cooperative and competitive strategies. It allows anyone to use the product but also provides income for the company. This is beneficial to all types of companies as it allows for both economic and technological growth but there is a risk of imitation when allowing potential competitors to have access to licensed products.

4.6 Joint Venture Formation

Similar to a merger, a joint venture (JV) involves two parties that form an alliance as separate entities. The joint venture allows both parties to cooperate, sharing resources, knowledge, and risks (Investopedia, 2017). A joint venture is purely cooperative as both parties must be willing to participate and share resources. An example of a joint venture is the Sony and Ericsson joint venture. Forming Sony Ericsson, the joint venture climaxed in 2012 as the 9th highest worldwide mobile phone vendor.

As mentioned in Section 2.3.2, for competing parties, a joint venture is beneficial when there is greater overall benefits than the partner's. another case in which a joint venture would be beneficial would be when more than two companies are competing in the same market, if one is significantly greater than the others, the smaller companies would benefit from a joint venture against the larger company. In certain cases, a joint venture may not be beneficial, this could be when the logistical costs of a joint venture are large, and the market sales may not produce an adequate return.

A joint venture is also a good way for a company to get a foothold in a foreign country. Instead of spending exorbitant amounts of time and resources on marketing in a foreign country, a company will generally have a better reception with a partner already in that market (Agarwal & Ramaswami, 1991).

4.7 Joint Venture Separation

The timing for a joint venture separation is also something to be considered when applying a technology strategy. A joint venture is based on either a mutual or contractual agreement, both of which can provide some benefit to a company if they decide to end the agreement.

If the joint venture is based on a mutual agreement, for example, two store owners agree to only sell on separate street. If one owner decides to separate from the joint venture and sell on the other's street, the second owner has lost some profit while the first person gains an advantage. Similarly, if a competitive company decides to leave a merger, it will do so in a way that is most beneficial for itself: make preparations without informing the partner, allowing the competing company to be ready for the separation and the partner in shock.

Contractually based joint ventures are more complex and both companies are legally required to fulfil the contract before leaving. Once separation has been agreed, the contract should define how the profits and resources are divided between each company. This allows both companies to have reassurance about their profits and resources.

5.0 Case Study – PC Industry

To validate the feasibility of using a game to simulate the growth of a company, a case study on the PC industry has been conducted. The findings from this case study will be implemented into the simulation to compare the game results with the present-day standings of each company. Six companies; Apple, Hewlett-Packard (HP), International Business Machines (IBM), Acer, Lenovo, and Dell, will be investigated and their strategy analysis will be implemented into the simulation.

5.1 History

The 1900's saw the start on an electronic revolution. Companies designing and producing electrical components such as semiconductors and transistors were becoming increasingly common and the prices of these components were decreasing drastically. Before the PC, computers were machines about the size of a room, costed hundreds of thousands of dollars (Swaine & Freiberger, 2014) and was generally only purchased by research laboratories and governments. The first "PC" is said to be made by a company called Micro Instrumentation Telemetry Systems (MITS). MITS developed the Altair 8800 in 1974, shown in Figure 6.

Figure 6: MITS Altair 8800 Computer (Swaine & Freiberger, 2014)

The Altair 8800 was a personal computer but very different to the contemporary computers. It was a personal computer in the sense that the price was low enough that a working-class citizen could afford one, but comprised of only an integrated circuit board, lights, and switches. It came as a kit that hobbyists could build themselves and included minimal features. The more conventional computer would soon to emerge.

5.2 Apple

5.2.1 History

After the release of the Altair 8800, many companies and entrepreneurs began building and selling similar computer kits, The Apple-I being one of them. Steve Jobs and Steve Wozniak founded Apple from their garage with an investment from Ron Wayne. Similarly to the Altair 8800, the Apple-I did not include a monitor, keyboard, or mouse but it did come fully assembled which was quite appealing to potential customers that did not have the technical knowledge to build their own computer from parts. It was also cheaper than the Altair whilst only slightly reducing the capabilities. The Apple-I computer was still more aimed at hobbyist that had the technical knowledge to build and use a computer rather than an average consumer. The Apple-II changed that by introducing an "all-in-one" design that incorporated a color screen and keyboard (Swaine & Freiberger, 2014). This computer could be seen as the first real commercial personal computer for all consumers.

5.2.2 Strategy Indicators

The Apple-II ran a version of Beginner's All-purpose Symbolic Instruction Code or BASIC (Dartmouth College, 1964), an easily accessible, free licensed code that allowed hobbyists to write their own software. In 1979, VisiCalc (Visible Calculations) was created on an Apple-II and was an instant success (Swaine & Freiberger, 2014). Personal computers had previously been for hobbyists and software designers and didn't have any practical use for normal consumers. VisiCalc changed that by bringing a business standard software to an affordable personal computer. As it was only available on Apple-II in its first year, the high demand for this program increased sales in the Apple-II.

Apple felt the need to build a new and improved machine as the hype of the Apple-II began to decline. Apple went public in 1980 to support the research and marketing costs (Swaine & Freiberger, 2014). In 1991, Apple CEO John Sculley presented an operating system that was compatible with an IBM-PC to IBM. Impressed with the display, Apple, IBM, and Motorola (Forming AIM) begin "collaborating" (Swaine & Freiberger, 2014) to design a new generation of computers. After several months, the joint venture with IBM was failing and had caused a loss of $300 million which led Apple to separate from the AIM alliance (Swaine & Freiberger, 2014).

Apple acquired NeXT Inc., a computer company owned by the Apple founder Steve Jobs (Swaine & Freiberger, 2014). Apple used the technology developed by NeXT in its next line of computers. To date, apple have acquired at least 90 technologies or companies with possibly more undocumented acquisitions (Crunchbase, 2017). In 2016, Apple Inc, had the 11th highest number of patents granted, a total of 2102 (IFI Claims Patent Servies, 2016)

5.2.3 Current Standings

From the first Apple computer, Apple have released various other computer models and is still producing new models today. Apple is currently the most valuable company in all of history with a market cap of $752 Billion (Forbes, 2017). Although Apple has the highest market cap, in the PC industry, Apple's market share is approximately 6.9% (currently 4th) as of the second quarter of 2017 (Gartner, 2017).

5.3 Hewlett-Packard

5.3.1 History

Hewlett-Packard was founded in 1939 by two friends, Bill Hewlett and Dave Packard. Both had been working together to make audio oscillators for Walt Disney (Hewlett-Packard, 2017). Oscillators were HP's main product until 1972 when they introduced the HP-35 Calculator. This calculator and following calculator models became the main product of the company until

1980 when HP released their first computer: the HP-85. Given the established foundation of the company, the HP-85 had reliable sales but not nearly as much as the Apple-II (Swaine & Freiberger, 2014).

5.3.2 Strategy Indicators

In 1963, HP forms its first joint venture with Yokogawa, forming Yokogawa-Hewlett-Packard (YHP) (Hewlett-Packard, 2017). In 2001, HP announces an agreement with Compaq to merge into a “gloabal technology leader” (Hewlett-Packard, 2017). To date, HP have made 157 acquisitions (Hewlett-Packard Alumni Association, 2017). In 2016, HP were granted 594 patents (IFI Claims Patent Servies, 2016).

5.3.3 Current Standings

HP is currently the top seller of PC’s worldwide (Gartner, 2017) with a market cap of $29.4 Bn (Forbes, 2017).

5.4 International Business Machines

5.4.1 History

International Business Machines is a company that began in 1911 as the “Computing-Tabulating-Recording Company” and sold machinery such as industrial time recorders, tabulators, and punched cards (International Business Machines, 2017). After entering the calculator market in 1944, IBM designed its first large computer: the IBM 701, a mainframe computer used mainly by governments and research laboratories due to its cost and size. Being such a large company with solid financial and technological foundation, IBM became one of the most recognised companies to produce mainframe computers and the slightly smaller, “fridge-sized” minicomputers. The personal computer industry was still niche until IBM released their first PC, the “IBM-PC”. When IBM introduced their PC, they legitimized the industry as they were seen as an established technological leader (Swaine & Freiberger, 2014). Something that gave IBM an edge was the use of the open operating system initially developed by Microsoft. This allowed software developers to create programs compatible with the IBM computers, something IBM hadn't done before and Apple weren't doing at the time. The open system provided both an advantage and disadvantage; it meant that software was readily being developed for the IBM-PC but left it vulnerable to imitations. This is what happened when Compaq Computer built a 100 percent IBM-compatible computer. Being compatible and cheaper, IBM sales were adversely affected. Soon many other companies would produce an IBM-compatible machine, leaving only the IBM brand as the standout between the machines. As cost started to become a more important selling point than specifications, IBM sales begin to dwindle. In 1992, IBM introduce the industry's first notebook, the IBM ThinkPad. After successful sales and innovative upgrades, the ThinkPad became the first notebook to be certified by the Trusted Computing Platform Alliance (Lenovo, 2017). In 2004, IBM announce an agreement to sell the IBM Personal Computing Division to Lenovo.

5.4.2 Strategy Indicators

The inception of IBM was through an acquisition, the computer, tabulating, and recording company was the merger of 3 companies: International Time Recording Company, Tabulating Machine Company, and Computing Scale Company (International Business Machines, 2017). To date, IBM have acquired at least 176 companies or technologies (Crunchbase, 2017).

Originally, IBM had closed source software, meaning if people were to buy an IBM computer, they had to purchase IBM built software and couldn't develop their own. When IBM released the IBM-PC, they used a Microsoft operating system called PC-DOS (Swaine & Freiberger, 2014). Microsoft founder, Bill Gates, convinced IBM to license the operating system and allow software developers to build software for their PC's. This is one of the major contributors to the success of the IBM-PC. IBM later formed a joint venture with Apple and Motorola to form the AIM alliance. This objective of this collaboration was to produce the next generation of computers.

In 2016, IBM were granted 8088 patents, making them the highest patent assignee of the year (IFI Claims Patent Servies, 2016).

5.4.3 Current Standings

IBM is non-existent in the PC industry today, with a market cap of $162.4 Bn (Forbes, 2017), it is not in the top 6 vendors of PC's by market share (Gartner, 2017).

5.5 Acer

5.5.1 History

Acer was founded in 1976 in Taiwan as Multitech. Initially the company dealt mainly in processors but quickly joined the IBM-PC compatible market in 1981 (Acer Inc., 2017).

5.5.2 Strategy Indicators

In 2007, Acer announced the acquisition of Gateway Inc, a strong U.S. based computer company, for $710 million (IDG News Service, 2007). Acer have made 13 other major acquisitions in its lifetime. In 2016, Acer was granted 109 patents (IFI Claims Patent Servies, 2016). In 2017, Acer and Starbreeze have announced a joint venture in the virtual reality market (Starbreeze, 2017).

5.5.3 Current Standings

Acer, with a market cap of $49.53Bn (Google, 2017), is now a household name when it comes to computers and laptops and is currently sitting sixth in the global market share (Gartner, 2017).

5.6 Lenovo

5.6.1 History

Founded in 1984 in China as Legend, Lenovo didn't begin to produce computers until 1990 when it released the Legend PC (Lenovo, 2017). In 2004, Lenovo acquired IBM's Personal

Computing Division, gaining the rights to the ThinkPad notebook and ThinkCentre Desktop PC.

5.6.2 Strategy Indicators

Lenovo's acquisition of the IBM ThinkPad and ThinkCentre was the crux in the company's sales. IBM were losing money and increasing debt fast, unable to keep a strong foothold in foreign markets, IBM PC sales were dwindling. Lenovo purchased the IBM PC division for $1.25Bn (Forbes, 2017). Lenovo (ranked 9th in the industry at the time) utilised their foothold in the Chinese market and the established branding of the ThinkPad and ThinkCentre series by IBM contributed to the global industry leader it is today. Purchasing the ThinkPad and ThinkCentre were not enough to account for the success of the company. Once acquired, Lenovo invested money and time to bring new innovations to the already advanced computer range (Lenovo, 2017).

In 2011, Lenovo and NEC Corporation announce a joint venture to form the largest PC vendor in the Japanese market (Lenovo, 2011). In 2016, Lenovo were granted 200 patents (IFI Claims Patent Servies, 2016).

5.6.3 Current Standings

Continuing to push the sales of the ThinkPad and ThinkCentre, Lenovo would become the #1 PC vendor worldwide in 2013 until 2017 (Gartner, 2017) where HP would overtake Lenovo in sales, leaving it in 2nd place. As of 2017, Lenovo has a market cap of $51.10bn (Forbes, 2017).

5.7 Dell

5.7.1 History

Like Apple, Dell was founded by Michael Dell in his garage in 1984. Dell focused on personal computers until 2003 when it began expanding into printers and home entertainment (Fell, 2017).

5.7.2 Strategy Indicators

In 2006, Dell acquired Alienware, a top of the line gaming PC manufacturer (Hachman, 2006). At the time, Alienware were a small company that built high-end gaming computers. The combination of innovation from Alienware and Dell's global connections brought prosperity to both brands. From its inception, Dell have acquired about 32 other companies or technologies (Crunchbase, 2017). In 2015, Dell announced a joint venture with Chinese vendor and service provider, Kingsoft (Judge, 2015). In 2016, Dell were granted 486 patents (IFI Claims Patent Servies, 2016).

5.7.3 Current Standing

Dell is currently the 3rd highest selling PC vendor (Gartner, 2017) with a market cap of $13.5 Bn (Forbes, 2017).

6.0 Game Simulation

To simulate the technology, a partially adversarial game has been constructed in Python.

6.1 Game Concept

The game involves between 2 to 6 players competing to earn a certain number of Tech Points (TP). In each step of the game, each player is given some money depending on how much TP they have. This is representative of the regular profit a company would receive from some product, where a higher level of technology allows for higher profits. After a certain time interval, each player has the potential upgrade(increase their TP) at the cost of money. The cost of each upgrade increases with respect to the amount of TP they have accumulated. This is to represent the increasingly difficult task of improving or inventing new technology, both of which require more resources than previously (Rogers, 1983).

To end the game, a player will need to reach a certain number of TP, the tech cap. Once this is achieved, a post-game analysis will summarise the results to see if the player achieved their goal. The Python code for the simulation can be found in Appendix 11.1

6.2 Game Mechanics

The mechanics behind each component of the game is described below, a detailed user guide can be found in Appendix 11.2.

6.2.1 Game Start-up

To begin the simulation, the user chooses the number of players present in the game, the Tech Cap, and whether to run the default settings or custom settings. Figure 7.

```
GAME START

Welcome to the game...
Use Default settings?
or play Custom game?
(0) Default Game
(1) Custom Game

0
Number of Players: 6
Tech Cap: 100
Max Steps: 410
Player 1 Strategy: [12, 20, 5, 30, 10, 5]
Player 1 Starting Time: 0

Player 2 Strategy: [20, 15, 2, 10, 10, 1]
Player 2 Starting Time: 40

Player 3 Strategy: [30, 25, 0, 15, 15, 5]
Player 3 Starting Time: 50

Player 4 Strategy: [15, 10, 1, 15, 15, 5]
Player 4 Starting Time: 70

Player 5 Strategy: [10, 15, 10, 15, 20, 1]
Player 5 Starting Time: 140

Player 6 Strategy: [10, 10, 7, 15, 10, 5]
Player 6 Starting Time: 80
```

Figure 7: Opening text for the simulation using default settings

The default settings are pre-set strategies and starting times for each player. The starting time is the point at which a company enters the market. If the user wishes to enter custom setting, they are able to set the starting time and strategy for each player. The strategies that a player can implement are: No Strategy (NS), Higher Investments (HI), Patent (PA), Acquisition (AQ), Free Licensing (FL), Paid Licensing (PL), Joint Venture (JV), and Joint Venture Separation (JVS).

6.2.2 Choosing a Strategy

During each turn, if the player is not currently implementing a strategy, they are able to choose a strategy. Since a "choice" made by the simulation would require artificial intelligence, the choosing mechanism is based on probability. The strategy mechanism is set in the form: [HI, PA, AQ, PL, FL, JV], where each strategy is given a number value. Since the choice replies on probability, the sum of these strategies must not exceed 100. NS and JVS were not included in the strategy choice as NS is the default strategy if none of these are chosen, and JVS requires the player to be in an alliance first. Since JVS is the probability that the player will want to leave the alliance, it can be set as the inverse of the JV attribute, where a player that has a high preference of forming an alliance would have a lower chance of breaking the alliance.

Using the form above, the choices made by a player of certain motivation can be simulated with high degree of accuracy. For example, if a competitive player with strong economical motivation was to be modelled, it could have a potential strategy of [10, 20, 20, 5, 10, 0]. This shows a strong affinity to creating a patent and wanting an acquisition whereas it would have a low probability of giving out a free license or forming a joint venture. A strategy is chosen at random and the player will generally continue to use this strategy for a certain number of turns depending on the situation.

During each turn, the order in which the players decide their strategy is randomised as to not allow for any player to have an advantage of acting first each turn.

6.2.3 No Strategy

The default strategy for each player is to not do anything significant other than continue gaining profit until they can increase their TP. While the player is implementing this strategy, they are able to choose a new one unless if it too soon after a previous strategy. If a player does not have enough money to implement a certain strategy, they will default back to NS

6.2.4 Higher Investments

If a player chooses to implement the HI strategy, they are choosing to pay significantly more money for an instantaneous increase in TP. This is analogous to a company spending more funding to complete more research faster. Within the game, the player would gain 1 TP at a higher price instantaneously.

6.2.5 Patent Application

Assuming all players are a part of the same industry, a patent would negatively affect all other players, causing them to spend resources on modifying their product to satisfy the patent law. In the game, for a very high cost, a patent will cause the other players to lose half of their current TP. Once a patent has been used by a player, that player cannot file for another patent for 30 steps.

6.2.6 Acquisitions

An acquisition can be a very resource consuming strategy, but can provide good benefits if completed successfully. To buy a company, the acquirer must be able to pay an offer of:

$$Cost_{AQ} = M_T + 2 * T_T \tag{1}$$

Where $Cost_{AQ}$ is the cost of the hostile takeover to the acquirer and, M_T and T_T is the money and TP owned by the target respectively. This means that a company with more resources and technology would be move expensive to takeover. If the acquirer is successful, the target gains the $Cost_{AQ}$ directly and loses all of its TP, the acquirer loses the takeover cost but gains half of the TP owned by the target. The acquirer doesn't get the full amount of TP from the target to account for factors such as technology cross-over (same tech in a acquiring company), tech incompatibility, and logistic losses in data. Once acquired, the target is left with money and no TP, it can stay like that for the rest of the game or choose to try and attempt a takeover if it has adequate funding. The acquirer will revert back to NS for 30 steps before being able to use another strategy.

6.2.7 Free Licensing

When a player implements a free license, they are giving open access to their product in hopes of useful feedback to increase their TP. For 10 steps, the player will lose some portion of money each turn but during the tech upgrade interval, they will gain 4 TP.

6.2.8 Paid Licensing

For the paid licensing strategy, the player gains some portion of money each turn but due to lower amount of sales than the FL, less feedback is given and they only gain 2 TP.

6.2.9 Joint Venture

When a player chooses to initiate a joint venture, the initiating player must first choose who to form an alliance with. To choose a partner, the initiating player takes into account the TP and money of all the players. A higher TP and more money give the player a higher probability of being chosen as a partner. The choice is once again randomised with ideal candidates having a higher probability of being chosen.

Once the partner has been chosen, the alliance is still not guaranteed. The partnership must be a mutual agreement, so the JV attribute of the potential partner affects the probability of the joint venture. Using probability principles, if both player were to have a 50% chance of wanting an alliance, there is a 25% chance that both agree to an alliance. This principle will determine the probability of a successful alliance formation, where two high JV probable player are significantly more likely to accomplish an alliance as opposed to two competitive (low JV) players.

Once the joint venture has been approved, the two companies merge and become a new entity, acting as one company and using shared resources. The alliance becomes a new player entirely, leaving the constituent players without TP or money. The resources available to the alliance is 80% of the combined TP from both companies and 70% of the combined money from both companies. The loss of TP is due to similar reasons as the loss during a hostile takeover. There is higher TP retention rate though as both players are willing to share TP. The loss of money is due to the cost of merging into one company, such as legal fees or logistical costs. The alliance uses the average of the constituent players' strategy, creating its own unique strategy, representative of the input from two companies where one player with an inclination towards HI can persuade the other to do so as well.

If one of the players is already in an alliance, then the same process is applied. There is still a chance of not successfully joining the alliance but once it is approved, the player adds all of their TP and money to the alliance, and incorporates their strategy.

6.2.10 Alliance Separation

The players that are in an alliance have no TP or money. They do not implement any strategy but while in an alliance, each player has the potential to leave at any point. The probability of leaving is dependent on their initial JV attribute. A player with a high JV attribute is less likely to leave an alliance opposed to a player with a low JV attribute. The JV attribute alone is not enough to simulate the probability of leaving an alliance. Once in an alliance, it is very unlikely that a player will want to leave, due to more losses in money and TP. To account for this, the probability of wanting to leave an alliance is reduced by a factor of 10.

If a player has chosen to separate from the alliance, they will return to being a single player with their unique strategy. If the JV only consisted of two players, the alliance will be dissolved completely, and the resources and TP will be split amongst the players. The share of the resources that each player gets is dependent on their initial contribution to the alliance. If the alliance has more than two players, the share of the leaving player is deducted from the TP and money and the strategy is adjusted for the two remaining players.

6.3 Post Game Results

The game will run until a player has reached the Tech Cap, once the Tech Cap has been reached, the game will report the end game analysis. The analysis for each game shows the final TP and money for each player. The post-simulation analysis will show the average final TP, the highest money amount (HMA) earnt during the simulation, and the average total money earnt (AME) during the game for each player. This allows analysis of which player consistently had the highest TP, money, and total money earnt during the simulation. The average steps taken (AST) for the simulation is also recorded to assess the research motivation.

7.0 Simulation Experiments

To test the simulation, the effects of each strategy will be investigated in various scenarios.

7.1 Test 1 – Isolation Test

The isolation test involves assessing each strategy alone, seeing how each one will affect various parameters. The game was run with only 1 player and a single strategy with a goal of reaching a TP of 15. The test was repeated 100 times and the AST, HMA, and AME were found. The average money earnt per step was also found to compare between strategies. Table 1 shows the results from the test.

Table 1: Isolation Test Results

	Probability (%)	Average Steps Taken (steps)	Highest Money Amount ($)	Average Total Money Earnt ($)	Average Money Earnt per Step($/step)
No Strategy	100	190	90.2	314.2	1.653
Higher Investment	25	203.4	74.6	356.1	1.7507
	50	207.81	69.2	366.256	1.7624
	75	208.98	65.6	370.284	1.7718
Free Licensing	10	87	77.2	135.48	1.5572
	30	76.6	79.82	112.48	1.4684
	50	73.6	79.82	108.16	1.4695
Paid Licensing	10	132.7	78.28	216.52	1.6316
	30	124.4	79.24	202.84	1.6305
	50	119.5	79.68	191.56	1.6030

Using NS as the control strategy, clear trends can be seen in the data that replicate the strategies.

7.1.1 Higher Investment Analysis

It can be seen for HI, as the probability of using the HI strategy increases, the average number of steps taken to complete a simulation increases. This is expected as the cost of an instant investment is higher than normal, if a company uses that strategy, they will inevitably increase the total amount of money needed to fulfil the objective. This is demonstrated in the increasing total money earnt for the HI test. It can also be seen that on average, the HI strategy provided more money to the player each step.

7.1.2 Free Licensing Analysis

For the free licensing, a clear relationship between the strategy preference to use a free license. The increasing probability of FL causes a significant reduction in the average steps and money earnt. This is expected as a Free License allows everyone to develop software and products and offer feedback, allowing for faster tech upgrades. The disadvantage of using the FL strategy is that it costs money to do so.

7.1.3 Paid License Analysis

For the PL simulations, once again there is a decrease in AST and AME as expected but the average values for both of these are higher than the average for the FL attribute. This is due to the lower amount of TP received when using a paid license which increases the number of steps required to complete a simulation. The higher AME is due to the strategy including an income source for the player which reduces the overall money losses.

7.1.4 Analysis Summary

This test shows that the simulation can represent the benefits and setbacks of each strategy accurately. For an economic motivation, a higher investment strategy provides the most money for a player in total and per step. For a research based motivation, the free license reaches Tech Cap the fastest.

7.2 Test 2 – Combination Test

The combination test had only one player but aims at investigating how the results are affected with a second strategy implemented (not including the No Strategy tactic). The results for this test are presented in Table 2 with data in the form of Average Steps Taken|Highest Money Amount|Average Money Earnt.

Table 2: Effects of Strategy Pairs

		HI Probability						PA Probability						POS Probability					
		25			50			5			15			10			50		
		AST (steps)	HMA ($)	AME ($)	AST (steps)	HMA ($)	AME ($)	AST (steps)	HMA ($)	AME ($)	AST (steps	HMA ($)	AME ($)	AST (steps)	HMA ($)	AME ($)	AST (steps)	HMA ($)	AME ($)
PA Probability	5	216	79.9	376.244	216.94	71.95	375.48												
	15	225.52	79.45	397.51	224.24	76.85	392.19												
POS Probability	10	170.22	93.49	282.27	189.32	92.79	317.59	145.4	80.32	242.52	164.6	79.2	283.52						
	50	136.83	83.7	222.66	147.8	91.29	242.6	124.7	79.72	203.54	131.8	78.88	218.52						
FOS Probability	10	138.96	79.8	226.12	138.99	85.2	224.46	97.8	75.94	158.88	122.7	77.88	216.52	98.1	79.56	156.98	109.6	79.24	174.82
	50	100.63	84.16	157.31	112.56	84.32	176.29	76.2	79.58	111.46	86.4	79.7	134.56	82.3	79.82	129.08	83.7	78.5	130.96

This test does not include NS, AQ, or JV because NS is a default strategy that was present in Test 1, and AQ and JV are both multiplayer attributes that only affect other players. Several important deductions can be made from the data recorded in this test.

7.2.1 Higher Investment Analysis

Firstly, looking at the HI strategy, when coupled with a patent, the average amount of money earnt each game and the average steps taken were among the highest in the test. This outcome was expected due to the results from test 1, where a higher patent probability increased the average steps taken and the average money earnt.

When coupled with a paid license, the higher PL attribute reduced the AST significantly, but also reduced the AME. When the free licence was used with the HI strategy, the average steps taken were minimal for the HI tests and so was the amount of money earnt.

7.2.2 Patent Application Analysis

Looking at the combination of patents and a paid licence, a higher PL attribute reduces the AST and a AME but increasing the PA attribute increases both factors. With the free license, the same effect is observed but with a higher reduction of AST and AME.

7.2.3 Paid License Analysis

A combination of PL and FL produced the lowest AST and AME. This is expected as they are both cooperative strategies and both aim to reduce the AST with sacrifice to AME.

7.2.4 Analysis Summary

Looking at all the analyses, it can be seen that each strategy in combination has a cumulative effect. Each simulation showed that when the probability of either strategy is increased, the AST and AME increase or decrease according to the how the strategy is affected alone. The most noticeable is the combination of patents and free licenses, where patents are highly competitive and free licenses are highly cooperative. Increase the FL reduced the AST and AME but increasing the patents increased the AST and AME.

7.3 Test 3 – Multiple Players

This test aims to compare each strategy against another strategy. Looking at 2 players, each with a single strategy, the simulation will run 100 games and have a Tech Cap of 15. It will investigate the number of wins by each player, AST, and AME for each player and strategy. Table 3 summarises the number of wins for each strategy in the form of: Player 1 wins | Player 2 wins. Refer to Appendix 11.3 for the AST and AME results from Test 3.

Table 3: Multiple Players Test: Number of Games Won

			Player 2 Strategy											
			HI		PA		AQ		PL		FL		JV	
			25	50	5	15	10	30	10	50	10	50	25	50
Player 1 Strategy	HI	25	58|60	85|30										
		50	76|36	54|60										
	PA	5	100|0	100|0	51|49	5|95								
		15	100|0	100|0	84|16	42|58								
	AQ	10	99|1	100|1	31|69	36|64	69|61	69|48						
		30	96|8	99|1	22|78	30|70	52|68	61|58						
	PL	10	100|0	100|0	100|0	88|13	97|3	97|3	70|71	38|94				
		50	100|0	100|0	100|0	100|0	100|0	100|0	95|45	65|62				
	FL	10	100|0	100|0	59|44	27|73	100|0	100|0	99|2	99|8	57|65	38|85		
		50	100|0	100|0	71|31	31|70	100|0	100|0	100|0	100|0	90|22	72|76		
	JV	25	97|12	94|9	1|100	0|100	59|86	58|62	0|100	0|100	0|100	0|100	31|30|56	6|3|92
		50	90|16	89|14	0|100	0|100	41|90	53|66	0|100	0|100	0|100	0|100	4|10|90	0|0|100

7.3.1 Higher Investment Analysis

When Player 2 had a HI strategy, the most effective strategies to win the game (reach Tech Cap) were PA, PL, and FL. These strategies allowed Player 1 to reach Tech Cap before Player 2 in all 100 repetitions. Notably, acquisition and joint venture both have above a 90% win rate. Using the same strategy (HI vs HI) produced near equal win with the same HI probability. It should be noted that using a patent strategy increased the AST compared to PL and FL but produced significantly higher money for the Player 1 on average, as predicted from the isolation test.

7.3.2 Patent Analysis

Analysis of the tests against the patent strategy show that reciprocating a patent strategy gives roughly a 50% chance of winning, a free license has a slightly higher chance and a paid license has the highest chance of winning the game. An acquisition has about a 30% chance of winning while appealing for a JV has almost no chance of winning. Against a stronger patent attribute, only a paid license has higher than a 50% chance of winning.

Against a PA strategy, a paid license had the least AST while using a HI strategy had the highest. Usually a free license would take the least amount of time but due to the competitive nature of a patent, the free license was unable to finish the game quickly. Due to the high AST when using a patent (over double the other attributes), the AME of the patent is also quite high, whereas PL had the lowest.

7.3.3 Acquisition Analysis

The simulations involving an acquisition strategy for Player 2 showed a high win rate (above 50%) for each strategy except JV. The high win rate from the other strategies is likely due to Player 2 being unable to afford to buy Player 1 using an AQ strategy alone, and the low win rate of a JV is due to the time and money invested into attempting a JV. Once again, FL has the lowest AST and AME and JV has the highest.

7.3.3 Paid License Analysis

Looking at the paid license simulations, there is a very high win rate for the player with a strong PL attribute except against a free license. Since a PL and FL don't affect each other, they will race until one player hits the Tech Cap. FL loses money when using this strategy but gains more TP while FL loses less money but gains less TP. With the current game balance, the rate of TP gain by FL is higher than that of PL.

7.3.4 Free License Analysis

The free license strategy is almost unbeatable unless Player 1 has a high patent probability or a higher FL attribute. Two players with a FL attribute greatly reduce the AST and both have very low AME.

7.3.5 Joint Venture Analysis

The joint venture attribute hasn't had much impact until now, two players with the joint venture attribute are able to form and alliance. When both players have 25% chance of forming an alliance, there is a 50% chance that the alliance will win the game. When both players have a probability of 50, the alliance chance is 100%. Notable results from the JV-JV test show that a lower JV probability in both players gives each individual player a higher AME when they won and when they both had a higher JV probability, the alliance had a higher AME than the if the players won individually.

7.3.6 Analysis Summary

Summary of the test analyses show that when each strategy is used against each other, two strategies stand out with strong winning rates: patents and free licenses. Both of these strategies are strongly supportive of a certain motivation: patents are very competitive and support the economic motivation, and free license are very cooperative, supporting the research motivation.

7.4 Simulation Nash Equilibrium

A payoff matrix has been constructed for the strategical options available within the simulation. Each Payoff matrix has been created for each potential strategy used by an opponent. In each matrix, T represents the current TP of the player and M represents the money.

7.4.1 General Payoff Matrix

Table 4 shows the payoff matrix for an opponent using NS.

Table 4: General Payoff Matrix for NS

			Player 2(No Strategy)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	T_1+1	T_2+1	0
		Money	$M_1-T_1^{1.5}$	$M_2-T_2^{1.5}$	0
	Higher Investment	TP	T_1+1	T_2+1	0
		Money	$M_1 - 1.5*T_1**^{1.5}$	$M_2-T_2^{1.5}$	0
	Patent Application	TP	T_1+1	$\frac{T_2}{2}+1$	0
		Money	M_1-25	$M_2-T_2^{1.5}$	0
	Acquisition	TP	$T_1+\frac{T_2}{2}$	0	0
		Money	$M_1 - M_2 – 2*T_2$	$2*M_2 + 2*T_2$	0
	Paid License	TP	T_1+2	T_2+1	0
		Money	$M_1 - 0.5*T_1^{1.5}$	$M_2-T_2^{1.5}$	0
	Free License	TP	T_1+4	T_2+1	0
		Money	$M_1 - 1.5*T_1^{1.5}$	$M_2-T_2^{1.5}$	0
	Joint Venture	TP	0	0	$(T_1+T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Joint Venture Separation	TP	$T_3*\frac{T_1}{T_1+T_2}$	$T_3*\frac{T_2}{T_1+T_2}$	0
		Money	$M_3*\frac{M_1}{M_1+M_2}$	$M_3*\frac{M_2}{M_1+M_2}$	0

The matrix for the remaining strategies can be found in Appendix 11.4. Like in the Prisoners' Dilemma, this matrix shows how a player's resources is affected given the choice of another player. If an example initial condition was incorporated, the matrix is easier to comprehend.

7.4.2 Example Payoff Matrix

For an example matrix, the following values will be used in a two player game:

- $T_1 = 4$
- $T_2 = 6$
- $M_1 = 30$
- $M_2 = 19$

Given these condition, the matrix turns into Table 5.

Table 5: Example Payoff Matrix

			Player 2 (No Strategy)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	5	7	0
		Money	22	4.3	0
	Higher Investment	TP	5	7	0
		Money	12	4.3	0
	Patent Application	TP	5	4.5	0
		Money	5	4.3	0
	Acquisition	TP	7	0	0
		Money	-1	50	0
	Paid License	TP	6	7	0
		Money	26	4.3	0
	Free License	TP	8	7	0
		Money	18	4.3	0
	Joint Venture	TP	0	0	7.2
		Money	0	0	34.4
	Joint Venture Separation	TP	3.2	4.8	0
		Money	24	15.2	0

From Table 5 it can be seen that if Player 1 had an economic motivation, they would prefer to use a paid license to both increase the TP and reduce the amount of money lost. If Player 1 had a motivation for research, a free license would provide the highest TP. For player 2, if they had an economic motivation, being acquired by Player 1 would provide the largest amount of money. If Player 2 had a research motivation, they would hope that Player 1 would not use a patent or acquisition.

Unlike the Prisoners' Dilemma where each player is assumed to have a similar motivation (less jail time) this matrix has 2 possible motivations. A completed matrix with the example initial conditions can be found in Appendix 11.5. it should be noted that in this matrix, all possible choices are investigated, and if the money amount goes below zero, the option is not viable.

For an economic motivation, the action that provides the most money for each player is if they are acquired by the opposing player. Firstly, this is not possible and would lead to an infinitely long game as each player would keep buying the other. Secondly, being acquired by another player may provide the highest money amount for that moment in the game but once bought out, there is very little chance of returning to the game and so, the amount of money they receive is likely their final money total. Economically, it would be better to hope the other player doesn't attempt an acquisition as continuing to play would allow continual profits, increasing the AME. The strategy that provides the most money and is dependent on the player's decision (not hoping the other will attempt an acquisition) is if both players use a paid license.

For a research orientated motivation, the ideal strategy for both players is if Player 1 uses a free license and Player 2 uses a paid license. For Player 2, a free license would actually give them more TP but as the player cannot afford to do so, their next best option is a paid license.

7.4.1 Analysis Summary

From this analysis, it can be seen that the paid license strategy is the Nash Equilibrium for Player 2 regardless of their motivation. For Player 1, their option is dependent on their motivation. All of the results from the example matrix are dependent on the TP and money of each player at a given time, meaning with each time step, the matrix changes, as does the ideal strategy. The matrix also only considers the local time optimal solution, meaning the ideal strategy for a given moment in time. If time were considered when choosing a strategy, both players would choose an acquisition as soon as possible. This is because when a player has been acquired, they are no longer competition and cannot negatively affect the player's resources and the player can grow freely. Such considerations are essential in Game Theory.

8.0 Case Study Simulation

The validity of the simulation will be tested by attempting to replicate the PC industry case study.

8.1 Case Study Analysis

8.1.1 Starting Time

The year of each company's first computer will serve as the starting time and also the number of turns to use in the simulation. Starting from 1976 when apple first released the Apple-I, each year will be represented by 10 steps in the program. Table 6 details the starting time and end of the simulation.

Table 6: Starting time for each company and end time for simulation

Company	Year of First PC Release	Years from 1976	Step Time Start
Apple	1976	0	0
HP	1980	4	40
IBM	1981	5	50
Acer	1981	5	50
Lenovo	1990	14	140
Dell	1984	8	80
Present Day (End Simulation)	2017	41	410

7.1.2 Company Strategy

From the information found in the case study, several indicators were found that will help determine the strategy attributes for each company. An example is that Dell and Lenovo both acquired other companies within the PC industry, giving them a higher AQ score. Another is the number of patents granted to each company in 2016. Table 7 shows the strategy to be used by each player/company in the form of: [HI, PA, AQ, PL, FL, JV].

Table 7: Player/Company Strategies

Player Number	Company	Strategy
1	Apple	[12, 20, 5, 30, 10, 5]
2	HP	[20, 15, 2, 10, 20, 1]
3	IBM	[30, 25, 0, 15, 15, 5]
4	Acer	[15, 10, 1, 15, 15, 5]
5	Lenovo	[10, 15, 10, 15, 20, 1]
6	Dell	[10, 10, 7, 15, 10, 5]

The values used for each strategy were tested and balanced according to attribute weightings used in the simulation. Further investigation between a company's strengths and the game attribute will need to be conducted.

8.2 Simulation Results

The simulation produced results that has some similarities to the current standings of each company. Appendix 11.1 has the Python code required to run the case study simulation. Figure 9 shows (a) the result from a single game and (b) the game statistics after 100 repetitions. It should be noted that each simulation ran for 410 steps, and since no companies reached the Tech Cap within that time, each company has no wins.

```
### ----- Game 0 Results ----- ###

Player: 1
Final Player Tech Score: 5
Final Player Money Count: 50.9

Player: 2
Final Player Tech Score: 10
Final Player Money Count: 16.35

Player: 3
Final Player Tech Score: 0
Final Player Money Count: 18.66

Player: 4
Final Player Tech Score: 5
Final Player Money Count: 34.15

Player: 5
Final Player Tech Score: 7
Final Player Money Count: 47.3

Player: 6
Final Player Tech Score: 6
Final Player Money Count: 47.46
```

(a)

```
### ----- GAME STATISTICS ----- ###

Player 1
Player Strategy: [12, 20, 5, 30, 10, 5]
Average Tech Points: 9.13
Highest Money Amount: 195.51
Total Money Earnt: 772.7
Games Won: 0

Player 2
Player Strategy: [20, 15, 2, 10, 10, 1]
Average Tech Points: 6.04
Highest Money Amount: 116.7
Total Money Earnt: 474.054
Games Won: 0

Player 3
Player Strategy: [30, 25, 0, 15, 15, 5]
Average Tech Points: 5.96
Highest Money Amount: 153.99
Total Money Earnt: 428.448
Games Won: 0

Player 4
Player Strategy: [15, 10, 1, 15, 15, 5]
Average Tech Points: 6.5
Highest Money Amount: 112.94
Total Money Earnt: 408.67
Games Won: 0

Player 5
Player Strategy: [10, 15, 10, 15, 20, 1]
Average Tech Points: 6.96
Highest Money Amount: 191.32
Total Money Earnt: 283.004
Games Won: 0

Player 6
Player Strategy: [10, 10, 7, 15, 10, 5]
Average Tech Points: 6.92
Highest Money Amount: 162.86
Total Money Earnt: 397.716
Games Won: 0
```

(b)

Figure 8: Case study simulation results (a) shows the results for an individual game and (b) shows the results for 100 repetitions

8.3 Results Analysis

From the results of a single game, it can be seen that the distribution for the player TP has Player 2 (HP) having the highest Tech Score and Player 3 (IBM) having the lowest. If the computer sales for each company is used as an analogy for the tech score, we see that the simulation has almost replicated the industry. Table 8 shows a comparison between the Tech Score for a single game and the company sales ranking according to Gartner (2017).

Table 8: Comparison between the simulation and global sales for a single game

Player Number	Company	Simulation (a) Tech Score	Simulation (a) Ranking	Current Global Sales Ranking
1	Apple	5	4th/5th	4th
2	HP	10	1st	1st
3	IBM	0	6th	6th
4	Acer	5	4th/5th	5th
5	Lenovo	7	2nd	2nd
6	Dell	6	3rd	3rd

Comparing these results to the results from the 100 repetition test shows conflicting conclusions. Table 9 compares the results from the repeated to the global sales for each company

Table 9: Comparison of data from repeated tests and global sales for each company

Player Number	Company	Simulation (b) Tech Score	Simulation (b) Ranking	Current Global Sales Ranking
1	Apple	9.13	1st	4th
2	HP	6.04	5th	1st
3	IBM	5.96	6th	6th
4	Acer	6.5	4th	5th
5	Lenovo	6.96	2nd	2nd
6	Dell	6.92	3rd	3rd

The analogy between the Tech Score in the simulation and the sales of a company is a simplified model of the business that assumes that better technology increases sales. Using this analogy, the Tech Scores from the single test corresponds to the global sales, but the results from the repeated tests are less convincing.

In the repeated tests, most simulation rankings are within 1 rank of their global sales ranking. Player 1 (Apple) and Player 2 (HP) have a considerable difference from their tech score to their sales ranking. The discrepancy between the single test and repeated test is in the nature of the strategy choice. Since the choice is based on probability, the conditions in each test can differ vastly. The repeated test allows for an average value to provide an estimation of relative rankings. It is more likely that the repeated test is a more accurate representation of the simulation results as opposed to the single test. Using the results for average total money earnt in the repeated tests, the average sum of money obtain in each test is provided. Table 10 shows the company's average total money earnt compared with the market cap.

Table 10: Comparison between the company market cap and average total money earnt in simulation

Player Number	Company	Simulation (b) Average Money Earnt	Simulation (b) Ranking	Current Market Cap ($bn)	Market Cap Ranking
1	Apple	772.7	1st	752	1st
2	HP	474.054	2nd	29.4	5th
3	IBM	428.448	3rd	162.4	2nd
4	Acer	408.67	4th	49.53	4th
5	Lenovo	283.004	6th	51.10	3rd
6	Dell	397.716	5th	13.5	6th

The majority of the comparisons from this data is incorrect except for the representation of the money owned by Apple. The discrepancies between the simulation and current standing of each company is most likely due to imbalance in the factors governing each strategy. Balancing these factors is not within the scope of this project but once completed in future investigations, the simulation will be able to recreate the scenarios more accurately.

9.0 Project Summary

9.1 Conclusions

In this investigation, a simulation has been made to determine what strategies are the most potentially beneficial for a company in various scenarios. Looking at 2 scenarios; economical and research based motivations for a company. A game was created to test these scenarios and tests were conducted to balance the game and test each scenario. Once tested, the PC industry was used as a case study and the simulation was used to attempt to recreate the current PC industry.

An isolation test was performed to see how a single strategy affects the game. NS, HI, PL, and FL were investigated as they did not require an opponent. The best strategy for an economically motivated company is to increase investments into research. Although a higher investment required more time to complete a game, the money earnings per turn was the highest with a HI strategy. For a research motivation, using a free license strategy allowed the player to finish the game the fastest, satisfying the research motivation.

A combination test was conducted to examine the effects of combining each strategy. Once again using only 1 player and this time investigating the HI, PA, PL, and FL strategies, the effects on AST, and AME are investigated. From this test, it was found that the strategies were independent and when used in conjunction, their respective effects were present. For example, when a patent and free license strategy were present, a higher FL reduced the AST and AME, increasing the patent attribute then increased the AST and AME. When both were increased, both the AST and AME decreased but only slightly, due to the higher FL attribute.

A test involving multiple players was conducted to see how the strategies would compete against each other. From this test, it was found that the patent and free license had the highest win rates and both strategies were supportive of one motivation type, economical and research respectively. Along with this test, a payoff matrix was constructed to analyse the ideal choices for each player. Using an example scenario with given TP and money for each player, the payoff matrix was calculated. In the example scenario, the best option for Player 2 was to use a paid license regardless of their motivation. For Player 1, the best option was to use a paid license for an economic motivation and a free license for a research based motivation. The payoff matrix only considered the local time optimal solution and didn't consider the potentials of acquiring a player or using a patent.

The final simulation conducted was to attempt to recreate the case study. The details regarding the PC industry history and 6 popular PC companies were investigated and implemented into the strategy. From the results of the simulation, it was found there was slight similarities between the case study and the simulation results, but the differences were too overwhelming to produce an accurate recreation. The simulation may never be able to perfectly recreate the development of an industry due to unforeseeable event that accelerate or destroy companies, such as a global financial crisis.

This simulation will need to be repeated with balanced mechanics in the game and an investigation into how a company's history can be converted into attributes for the simulation, will need to be conducted. Both of these are not included in the scope of this investigation.

9.2 Recommendations

In this investigation, the objectives were not sucessfully completed. Ideal strategies for companies with an economical and research based motivation were found using the simulation but upon testing the simulation in the case study, the simulation was found to be inaccurate.

The game concept and the actions for each strategy are believed to be accurate but the strengths of each strategy need to be balanced. To do so, it is recommended that more tests be conducted to find the best balance for the game. Once this is completed, an investigation will need to be performed on the case study to find a process of converting the statistics of a company into profile for each company to use in the game. The balance of the game mechanics will be a factor in the conversion process.

To improve the reliability of the decision making, a low-level AI is recommended to be implemented as a player. The AI would only need to assess the decisions made by other players and act according to a predetermined motivation. The payoff matrix can be used as a decision guide for the AI as well as a set strategy input. For this AI, to simulate a realistic decision, it will need to consider time and the consequences of certain actions.

10.0 References

Acer Inc. (2017, October). *Milestones & Innovations*. Retrieved from Acer Group: https://www.acer-group.com/ag/en/TW/content/history

Agarwal, S., & Ramaswami, S. N. (1991). Choice of Foreign Market Entry Mode: Impact of Ownership, Location and Internalization Factors. *JSTOR*.

Australian Government. (2016, May). *Patent Basics*. Retrieved from Australian Government: https://www.ipaustralia.gov.au/patents/understanding-patents/patent-basics

Bleeke, J., & Ernst, D. (1995). Is Your Strategic Alliance Really a Sale? *Harvard Business Review*, 97-105.

Crunchbase. (2017, October). *Acquisitions*. Retrieved from Crunchbase: https://www.crunchbase.com/search/acquisitions/field/organizations/num_acquisitions

Dartmouth College. (1964). *BASIC*. Dartmouth College.

Deutsche Welle. (2010, February). *Mannesmann: the mother of all takeovers*. Retrieved from Deutsche Welle: http://www.dw.com/en/mannesmann-the-mother-of-all-takeovers/a-5206028

ECRI Institute. (2017, October). *About ECRI Institute*. Retrieved from ECRI Institute: https://www.ecri.org/about/Pages/default.aspx

Ernst, H. (2003). Patent Information for Strategic Technology Management. *Worl Patent Information 25*, 1.

Evenson, R. E., & Westphal, L. E. (1994). *Technological Chance and Technological Strategy*.

Fell, J. (2017). *3 Decades of Dell*. Retrieved from Entrepreneur: https://www.entrepreneur.com/article/225670

Forbes. (2017, May). *#7 Apple*. Retrieved from Forbes Global 2000: Top Regarded Companies: https://www.forbes.com/companies/apple/

Forbes. (2017). *Global 2000*. Retrieved from Forbes: https://www.forbes.com/global2000/#2634036c335d

Gartner. (2017, July). *Global PC Shipments Decrease for 11th Straight Quarter*. Retrieved from Gartner: https://www.gartner.com/newsroom/id/3759964

Gibbons, R. (1992). *A Primer in Game Theory.* Essex, England: Pearson Education Limited.

Google. (2017, October). *Acer Inc*. Retrieved from Google Finance: https://finance.google.com/finance?q=TPE:2353

Grindley, P. C., & Teece, D. J. (1997). Managing Intellectual Captial: Licensing and Cross-Licensing in Semiconductors and Electronics. *California Management Reveiw*, 8-41.

Hachman, M. (2006, March). *IT's Official: Dell beams up Alienware*. Retrieved from PC Mag: https://www.pcmag.com/article2/0,2817,1941376,00.asp

Hewlett-Packard. (2017, October). *Timeline of our History*. Retrieved from HP: http://www8.hp.com/us/en/hp-information/about-hp/history/hp-timeline/timeline.html

Hewlett-Packard Alumni Association. (2017, September). *Acquisitions, Divestitures, and Spinoffs*. Retrieved from HP Alimni: http://www.hpalumni.org/acquisitions

Hottenrott, H., & Peters, B. (2009). Innocative Capability and Financing Constraints for Innovation. More Money, More Innovation? *ZEW: Centre for European Economic Research*, 9-81.

IDG News Service. (2007, August). *Acer to Acquire Gateway in $710 Million Deal*. Retrieved from PC World: https://www.pcworld.com/article/136415/article.html

IFI Claims Patent Servies. (2016). *IFI Claims Top 1000 Assignees for 2016*. Retrieved from IFI Claims: https://www.ificlaims.com/rankings-misc-top-1000-2016.htm

International Business Machines. (2017, October). *Chronological History of IBM*. Retrieved from IBM: http://www-03.ibm.com/ibm/history/history/decade_1910.html

Investopedia. (2017, October). *Joint Venture - JV*. Retrieved from Investopedia: http://www.investopedia.com/terms/j/jointventure.asp

Investopedia. (2017, October). *Takeover*. Retrieved from Investopedia: http://www.investopedia.com/terms/t/takeover.asp

Judge, P. (2015, September). *Dell announces $125bn China Plan*. Retrieved from Data Center Dynamics: http://www.datacenterdynamics.com/content-tracks/colo-cloud/dell-announces-125bn-china-plan/94779.fullarticle

Kolk, M., Webster, P., Uchida, K., & Jansz, J. (2015). Capturing sustainable value from technologyL a case for strategic licensing. *Prism*, 78-89.

Lenovo. (2011, January). *Lenovo and NEC form joint venture to create Japan's largest PC group*. Retrieved from Lenovo Newsroom: http://news.lenovo.com/news-releases/lenovo-and-nec-form-joint-venture-to-create-japans-largest-pc-group-1.htm

Lenovo. (2017). *Company History*. Retrieved from Lenovo: https://www3.lenovo.com/in/en/lenovo/company-history/

Levman, S. (1998). Competition and Cooperation. *Michigan Law Review*, 216-244.

Meyer, A. D. (2006). Technology Strategy. *Working Paper Series*.

Policonomics. (2017). *Game Theory II: Prisoner's Dilemma*. Retrieved from Policonomics: http://policonomics.com/lp-game-theory2-prisoners-dilemma/

Rawls, J. (1971). *A Theory of Justice.* Harvard University Press.

Rogers, E. M. (1983). *Diffusion of Innovation.* New York: The Free Press.

Root, F., & Visudtibhan, K. (1992). *International Strategic Management.* New York: Hemisphere Publishing Corporation.

Shubik, M. (1971). The Dollar Auction Game: A Paradox in Noncooperative Behavior and Escalation. *The Journal of Conflict Resolution, Volume 15*, 109-111.

Smith, A. (1776). *An Inquiry Into the Nature and Causes of the Wealth of Nations, Volume 1.*

SpaceX. (2017, March). *Reusability: The Key to Making Human Life Multi-Planetary*. Retrieved from SpaceX: http://www.spacex.com/news/2013/03/31/reusability-key-making-human-life-multi-planetary

Starbreeze. (2017, October 24). *Starbreeze partner Acer steps up financing of the StarVR joint venture*. Retrieved from Starbreeze: https://www.starbreeze.com/2017/10/starbreeze-partner-acer-steps-up-financing-of-the-starvr-joint-venture/

Swaine, M., & Freiberger, P. (2014). *Fire in the Valley: Birth and Death of the Personal Computer*. Rayleigh: Pragmatic Bookshelf.

Virgin Galactic. (2017, October). *Who We Are*. Retrieved from Virgin Galactic: http://www.virgingalactic.com/who-we-are/

11.0 Appendices

11.1 Game Simulation Code

Please read User Guide before running simulation.

11.1.1 Game Base Code

```
# -*- coding: utf-8 -*-
"""
Created on Tue Oct 17 23:47:05 2017

@author: Quan
"""
import numpy as np
import random as rdm
from Strategy_Choice import Strategy_Choice
from Buyout import Buyout
from Joint_Venture import Joint_Venture
from Alliance_Split import Alliance_Split
np.set_printoptions(threshold=np.nan)

#===============================================================================
# Game Introduction
#===============================================================================
print("GAME START")
print(" ")
print("Welcome to the game...")
print("Use Default settings?")
print("or play Custom game?")
print("(0) Default Game")
print("(1) Custom Game")
print("(2) Case Study")
def_strat = int(input())  #input default or custom settings

### Case Study ###
if def_strat == 2:

    """
    Using PC industry as default settings
    Player 1 = Apple
    Player 2 = HP
    Player 3 = IBM
    Player 4 = Acer
    Player 5 = Lenovo
    Player 6 = Dell
    """

    NPlayers = 6
    Tech_Cap = 100
    Max_Steps = 410
    Strategy1 = [[12, 20, 5, 30, 10, 5], \
                 [20, 15, 2, 10, 20, 1], \
                 [30, 25, 0, 15, 15, 5], \
                 [15, 10, 1, 15, 15, 5], \
                 [10, 15, 10, 15, 20, 1], \
                 [10, 10, 7, 15, 10, 5]]
    Starting_Time = [0, 40, 50, 70, 140, 80]

    print("Number of Players: 6")
    print("Tech Cap: 100")
    print("Max Steps:", Max_Steps)
    for i in range(len(Strategy1)):
        print("Player", i+1, "Strategy:", Strategy1[i])
        print("Player", i+1, "Starting Time:", Starting_Time[i])
        print(" ")

### Default Settings
elif def_strat == 0:


    NPlayers = 6
```

```
    Tech_Cap = 20
    Max_Steps = 99999
    Strategy1 = [[20, 10, 5, 10, 10, 5], \
                 [10, 20, 5, 10, 10, 5], \
                 [10, 10, 10, 10, 10, 5], \
                 [10, 10, 5, 20, 10, 5], \
                 [10, 10, 5, 10, 20, 5], \
                 [10, 10, 5, 10, 10, 50]]
    Starting_Time = [0, 0, 0, 0, 0, 0]

    print("Number of Players: 6")
    print("Tech Cap: 20")
    print("Max Steps:", Max_Steps)
    for i in range(len(Strategy1)):
        print("Player", i+1, "Strategy:", Strategy1[i])
        print("Player", i+1, "Starting Time:", Starting_Time[i])
        print(" ")

### Custom Settings ###
else:

    NPlayers = int(input("Number of Players:"))
    Tech_Cap = int(input("Tech Cap:"))
    Max_Steps = int(input("Max Steps:"))
    Strategy1 = []
    Starting_Time = []
    print("Strategy Layout")
    print("[High Investment, Patent Application, Acquisition,\
    Paid License, Free License, Joint-Venture]")
    print("Sum of all attributes must be less than 100")
    for i in range(NPlayers):
        HI = int(input("Higher Investment Level:"))
        print("Remaining:" , 100-HI)
        Pat = int(input("Patent Application Level:"))
        print("Remaining:" , 100 - HI+Pat)
        buy = int(input("Buyout Level:"))
        print("Remaining:" , 100 - HI+Pat+buy)
        POS = int(input("Partially Free Open-Source Level:"))
        print("Remaining:" , 100 - HI+Pat+buy+POS)
        FOS = int(input("Free Open-Source Level:"))
        print("Remaining:" , 100 - HI+Pat+buy+POS+FOS)
        JV = int(input("Joint-Venture Level:"))
        print("Remaining:" , 100 - HI+Pat+buy+POS+FOS+JV)
        Strategy1.append([HI, Pat, buy, POS, FOS, JV])
        start_time = int(input("Player Starting Time:"))
        Starting_Time.append(start_time)



"""
Player strategy layout
[[invest more, patent, Buyout, partial open-source,  open source, JV]]
"""

#===============================================================================
# Game Factors
#===============================================================================

Strat_Steps = 10 #how long a player must keep their strategy
Tech_Cost = 20
Tech_Factor = 0.1
Step_Money = 2
HI = 1.25 #Higher investment factor

alliance_list = []
alliance_resources = []

Tech_Counter_Step = list(np.linspace(0, Max_Steps, (Max_Steps/10)+1))
del Tech_Counter_Step[0]
Tech_Counter_Step = np.array(Tech_Counter_Step)


repetitions = int(input("Number of Repetitions:"))

tech_rep = np.zeros(8)
money_rep =  np.zeros(8)
Money_Sum = np.zeros(8)
games_won = []
step_rep = []
```

```
print("Auto Run?")
print("(0) Yes")
print("(1) No")

Auto_Run = int(input())


for reps in range(repetitions):

    Strategy = Strategy1
    Strat_Choice = np.zeros(NPlayers)
    Revert_Strategy = np.zeros(NPlayers) #List of when a player can change strategy
    Player_List = list(np.linspace(1, NPlayers, NPlayers))
    Player_List = [int(x) for x in Player_List]
    Player_List2 = np.copy(Player_List)
    Tech_Counter = np.zeros(NPlayers)
    Tech_Counter = [x+1 for x in Tech_Counter]
    Money_List = np.zeros(NPlayers)
    alliance_list = []
    alliance_resources = []

    for Steps in range(Max_Steps): #Step Iteration
        if Auto_Run == 0:
            pass
        else:
            print("Step:", Steps)
            print(" ")
        alliance_del = [] #if alliance is deleted, it wont act that turn
        Player_List2 = np.copy(Player_List)
        rdm.shuffle(Player_List2)
        for Player2 in Player_List2: #Player Iteration
            if (Player2 not in alliance_del) is False:
                pass
            else:
                PI = Player_List.index(Player2) #original player index
            if Auto_Run == 1:
                print(" ")
                print("Player", Player_List[PI], "turn")

            if (Player2 not in alliance_del) is False:
                pass
            elif Starting_Time[PI] <= Steps: #Testing if the player has started
                #Player income added
                Money_List[PI] = round(Money_List[PI] + Step_Money*\
                                      Tech_Counter[PI]*Tech_Factor,2)
                Money_Sum[PI] += round(Step_Money*Tech_Counter[PI]*Tech_Factor,2)
                if Money_List[PI] >= money_rep[PI]:
                    money_rep[PI] = Money_List[PI]
                ### SRATEGY CHOICE ###
                if Steps >= Revert_Strategy[PI]:
                    Strat_Choice[PI] = int(Strategy_Choice(PI, Player_List, \
                                        Strategy, Tech_Counter, alliance_list))
                    if Strat_Choice[PI] >= 1:
                        Revert_Strategy[PI] = Steps + Strat_Steps
    #            print(Strat_Choice)

                ### NO STRATEGY ###
                if Strat_Choice[PI] == 0:
                    if Tech_Counter[PI] >= 0.0001:
                        #tech development every ith iteration
                        if (Steps not in Tech_Counter_Step) is False:
                            if Auto_Run == 1:
                                print("Player", Player_List[PI], "No Strategy")
                            #player cant tech up if doesnt have enough money
                            if (Money_List[PI] - int((Tech_Counter[PI])**1.5)) >= 0:
                                #deduct tech cost
                                Money_List[PI] -= int((Tech_Counter[PI])**1.5)
                                if Tech_Counter[PI] == 0:
                                    pass
                                else:
                                    Tech_Counter[PI] += 1 #increase tech
                            else:
                                if Auto_Run == 1:
                                    print("Player", Player_List[PI],\
                                          "cant afford tech upgrade")
                                pass

                ### HIGHER INVESTMENT ###
```

```
elif Strat_Choice[PI] == 1:
    if Tech_Counter[PI] >= 0.0001:
        if Auto_Run == 1:
            print("Player", Player_List[PI], "is investing more money")
        #player cant tech up if doesnt have enough money
        if (Money_List[PI] - HI*int((Tech_Counter[PI])**1.5)) >= 0:
            #deduct tech cost
            Money_List[PI] = Money_List[PI] - HI*int((Tech_Counter[PI])**1.5)
            if Tech_Counter[PI] ==0:
                pass
            else:
                Tech_Counter[PI] += 1 #increase tech
                Strat_Choice[PI] = 0
                Revert_Strategy[PI] = Steps+20
        else:
            if Auto_Run == 1:
                print("Player", Player_List[PI], "cant afford tech upgrade")
                Strat_Choice[PI] = 0
            pass

### COPYRIGHT/PATENT ###
elif Strat_Choice[PI] == 2:
    if Tech_Counter[PI] >= 0.0001:
        if Steps <= 25:
            if Auto_Run == 1:
                print("Too early to obtain patent")
            Strat_Choice[PI] = 0
            Revert_Strategy[PI] = 0

        else:
            Patent_Cost = 25
            if Auto_Run == 1:
                print("Player", Player_List[PI], "attempts copright/patent")
            if Money_List[PI] - Patent_Cost >= 0:
                if Auto_Run == 1:
                    print("Player", Player_List[PI],\
                          "acquires copright/patent")
                Money_List[PI] = Money_List[PI] - Patent_Cost
                for i in range(len(Player_List)):
                    if i == PI:
                        pass
                    else:
                        if Tech_Counter[i] >= 5:
                            Tech_Counter[i] = int(Tech_Counter[i]/2)
                Strat_Choice[PI] = 0
                Revert_Strategy[PI] = Steps + 30
            else:
                if Auto_Run == 1:
                    print("Player", Player_List[PI],\
                          "cant afford copright/patent")
                Strat_Choice[PI] = 0
                Revert_Strategy[PI] = Steps + 10

        #tech development every ith iteration
        if (Steps not in Tech_Counter_Step) is False:
            #player cant tech up if doesnt have enough money
            if (Money_List[PI] - int((Tech_Counter[PI])**1.5)) >= 0:
                #deduct tech cost
                Money_List[PI] -= int((Tech_Counter[PI])**1.5)
                if Tech_Counter[PI] == 0:
                    pass
                else:
                    Tech_Counter[PI] += 1 #increase tech
            else:
                if Auto_Run == 1:
                    print("Player", Player_List[PI], \
                    "cant afford tech upgrade")
                pass
### BUYOUT ###
elif Strat_Choice[PI] == 3:
    if Tech_Counter[PI] >= 0.0001:
        if Steps <= 30:
            if Auto_Run == 1:
                print("Too early to buyout other company")
            Strat_Choice[PI] = 0
            Revert_Strategy[PI] = 0
        else:
            if Auto_Run == 1:
                print("Player", Player_List[PI], \
```

```
                  "attempting to buyout another company")
            Money_List, Tech_Counter = Buyout(PI, Player_List, \
                                    Tech_Counter, Money_List, Auto_Run)
            Strat_Choice[PI] = 0
            Revert_Strategy[PI] = Steps+20

    ### Partial open source ###
    elif Strat_Choice[PI] == 4:
        if Tech_Counter[PI] >= 0.0001:
            Money_List[PI] += 0.2*Step_Money*Tech_Counter[PI]*Tech_Factor
            if Auto_Run == 1:
                print("Player", Player_List[PI],\
                      "using partial open source strategy")
            #tech development every ith iteration
            if (Steps not in Tech_Counter_Step) is False:
                 #player cant tech up if doesnt have enough money
                if (Money_List[PI] - int((Tech_Counter[PI])**1.4)) >= 0:
                    #deduct tech cost
                    Money_List[PI] -= int((Tech_Counter[PI])**1.4)
                    if Tech_Counter[PI] == 0:
                        pass
                    else:
                        Tech_Counter[PI] += 2 #increase tech
                        Strat_Choice[PI] = 0
                        Revert_Strategy[PI] = Steps + 20
                else:
                    if Auto_Run == 1:
                        print("Player", Player_List[PI], \
                        "cant afford tech upgrade")
                        pass
        else:
            pass
    ### Full Open Source ###
    elif Strat_Choice[PI] == 5:
        if Tech_Counter[PI] >= 0.0001:
            Money_List[PI] -= 0.1*Step_Money*Tech_Counter[PI]*Tech_Factor
            if Auto_Run == 1:
                print("Player", Player_List[PI], \
                "using full open source strategy")

             #tech development every ith iteration
            if (Steps not in Tech_Counter_Step) is False:
                #player cant tech up if doesnt have enough money
                if (Money_List[PI] - int((Tech_Counter[PI])**1.4)) >= 0:
                    #deduct tech cost
                    Money_List[PI] -= int((Tech_Counter[PI])**1.4)
                    if Tech_Counter[PI] == 0:
                        pass
                    else:
                        Tech_Counter[PI] += 4 #increase tech
                        Strat_Choice[PI] = 0
                        Revert_Strategy[PI] = Steps + 20
                else:
                    if Auto_Run == 1:
                        print("Player", Player_List[PI], \
                        "cant afford tech upgrade")
                    pass
        else:
            pass
    ### Joint Venture ###
    elif Strat_Choice[PI] == 6:
        if Tech_Counter[PI] >= 0.0001:
            if Steps <= 5:
                if Auto_Run == 1:
                    print("Too early to form an alliance")
                Strat_Choice[PI] = 0
                Revert_Strategy[PI] = Steps+10
            else:
                if Auto_Run == 1:
                    print("Player", Player_List[PI], \
                    "attempting joint venture")
                Player_List, Tech_Counter, \
                Money_List, Strategy, alliance_list, \
                alliance_resources, Starting_Time, Revert_Strategy,\
                Strat_Choice = Joint_Venture\
                (PI, Player_List, Tech_Counter, \
                 Money_List, Strategy, alliance_list, \
                 alliance_resources, Starting_Time, Steps,\
                 Revert_Strategy, Strat_Choice, Auto_Run)
```

```
                elif Strat_Choice[PI] == 7:
                    if Auto_Run == 1:
                        print("Player", Player_List[PI], "wants to leave the alliance")
                    Tech_Counter, Money_List, Player_List, Strategy, Revert_Strategy, \
                           Strat_Choice, Starting_Time, alliance_resources, \
                           alliance_list , alliance_del = Alliance_Split\
                           (PI, Player_List, alliance_list, alliance_resources,\
                            Strategy, Revert_Strategy, Strat_Choice, Tech_Counter, \
                            Money_List, Tech_Cap, Starting_Time, Steps, alliance_del,\
                            Auto_Run)

            else:
                pass
        if np.amax(Tech_Counter) >= Tech_Cap:
            break
        if Auto_Run == 1:
            print("Players:", Player_List)
            print("Tech:", [float("%.2f"% m) for m in Tech_Counter])
            print("Money:", [float("%.2f"% m) for m in Money_List])
            print("Alliance List", alliance_list)

        if Auto_Run == 0:
            end = 0
        else:
            end = input()
            if end == '1':
                break


    print(" ")
    print(" ### ----- Game", reps, "Results ----- ###")
    for i in range(len(Player_List)):
        print(" ")
        print("Player:", Player_List[i])
        temp = 1
        for j in range(len(alliance_list)): #check if player is the alliance
            if Player_List[i] == alliance_list[j][0]:
                print("Player", Player_List[i], "is an alliance made from:")
                for k in range(len(alliance_list[j])):
                    if k == 0:
                        pass
                    else:
                        print("Player", alliance_list[j][k])
                print("Final Alliance Tech Score:", round(Tech_Counter[i],2))
                print("Final Alliance Money Count:", round(Money_List[i],2))
                temp = 0
            else:
                pass

        if int(Tech_Counter[i]) and (Money_List[i]) == 0: #Player is in an alliance
            print("Player is in an alliance")
        elif temp == 0:
            pass
        else:
            print("Final Player Tech Score:", round(Tech_Counter[i], 2))
            print("Final Player Money Count:", round(Money_List[i],2))
    for i in range(len(Player_List)):
        tech_rep[i] += Tech_Counter[i]

    step_rep.append(Steps)

    for i in range(len(Tech_Counter)):
        if Tech_Counter[i] >= Tech_Cap:
            games_won.append(Player_List[i])

if repetitions >= 2:
    print(" ")
    print(" ### ----- GAME STATISTICS ----- ###")
    print(" ")
    for i in range(len(tech_rep)):
        print("Player", i+1)
        if i <= len(Strategy)-1:
            print("Player Strategy:", Strategy[i])
        print("Average Tech Points:", tech_rep[i]/repetitions)
        print("Highest Money Amount:", money_rep[i])
        print("Total Money Earnt:", Money_Sum[i]/repetitions)
        win_sum = 0
        for j in range(len(games_won)):
```

```
            if i+1 == games_won[j]:
                win_sum += 1
        print("Games Won:", win_sum)
        print(" ")
    print("Average Number of Steps:", sum(step_rep)/len(step_rep))

#print(money_rep)
#print([round(i/repetitions,2) for i in Money_Sum])
```

11.1.2 Strategy Choice Code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 11:18:53 2017

@author: Quan
"""

import random as rdm
import numpy as np

#==============================================================================
# Strategy Choice
#==============================================================================

def Strategy_Choice(PI, Player_List, Strategy, Tech_Counter, alliance_list):
    Strat = Strategy[PI]
    Strat_Chance = []
    for i in alliance_list:
        for j in i:
            if j == Player_List[PI] and Tech_Counter[PI] == 0:
                for i in range(int((100-Strategy[PI][5])/20)):
                    Strat_Chance.append(7)
#                 print(Strat_Chance)
                for k in range((100 - len(Strat_Chance))):
                    Strat_Chance.append(0)
                return rdm.choice(Strat_Chance)
            else:
                pass
    for strat, chance in enumerate(Strat, 1):
#        print(Player, strat, chance)
        if strat == 0:
            pass
        else:
            for j in range(int(chance)):
                 Strat_Chance.append(strat)
    for k in range((100 - len(Strat_Chance))):
        Strat_Chance.append(0)
    choice = rdm.choice(Strat_Chance)
#    print(choice)
    return choice
```

11.1.3 Acquisition Code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 17:18:20 2017

@author: Quant
"""

import random as rdm

#==============================================================================
# Acquisition
#==============================================================================

def Buyout(PI, Player_List, Tech_Counter, Money_List, Auto_Run):
    buy_chance = []
    for i in range(len(Player_List)):

        if i == PI:
            pass
        elif Tech_Counter[i] == 0:
            pass
        elif Money_List[i] == 0:
            pass
        else:
            potential = int(30/(Money_List[i]))
            for j in range(potential):
                buy_chance.append(Player_List[i])
    for i in range(len(Player_List)):
        if i == PI:
            pass
        elif Tech_Counter[i] == 0:
            pass
        elif Tech_Counter[i] == 1:
            pass
        else:
            potential = int((Tech_Counter[i])/2)
            for j in range(potential):
                buy_chance.append(Player_List[i])
    if buy_chance == []:
        return Money_List, Tech_Counter

    buyout_client = rdm.choice(buy_chance)
    client_index = Player_List.index(buyout_client)
    if Auto_Run == 1:
        print("Player", Player_List[PI], "attempting to buy player", buyout_client)

    if Money_List[PI] <= Money_List[client_index] + Tech_Counter[client_index]*2:
        if Auto_Run == 1:
            print("Player", Player_List[PI], "cant afford buyout")
        pass
    else:
        Money_List[PI] -=  round(Money_List[client_index] + Tech_Counter[client_index]*2,2)
        Money_List[client_index] += round(Money_List[client_index] + \
                                          Tech_Counter[client_index]*2,2)

        Tech_Counter[PI] += round(int(Tech_Counter[client_index]/2),2)
        Tech_Counter[client_index] = 0
        if Auto_Run == 1:
            print("Player", Player_List[PI], "successfully bought out player", buyout_client)
    return Money_List, Tech_Counter
```

11.1.4 Joint Venture Code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 22:56:17 2017

@author: Quant
"""

import numpy as np
import random as rdm

#===============================================================================
# Joint Venture
#===============================================================================

def Joint_Venture(PI, Player_List, Tech_Counter, Money_List, \
                  Strategy, alliance_list, alliance_resources, \
                  Starting_Time, Steps, Revert_Strategy, Strat_Choice, Auto_Run):

    ### CHOOSING POTENTIAL COPANY TO BUY ###
    potential_alliance = []

    for i in range(len(Player_List)):
        if i == PI:
            pass
        elif Tech_Counter[i] == 0:
            pass
        else:
            potential = int((Money_List[i])/2)
            for j in range(potential+1):
                potential_alliance.append(Player_List[i])
    for i in range(len(Player_List)):
        if i == PI:
            pass
        elif Tech_Counter[i] == 0:
            pass
        else:
            potential = int((Tech_Counter[i])/2)
            for j in range(potential+1):
                potential_alliance.append(Player_List[i])

    if potential_alliance == []: #Returns if no potential alliance partner
        Revert_Strategy[PI] = Steps+1000
        Strat_Choice[PI] = 0
        return Player_List, Tech_Counter, Money_List, Strategy, \
        alliance_list, alliance_resources, Starting_Time, Revert_Strategy, Strat_Choice

    alliance_client = rdm.choice(potential_alliance)
    client_index = Player_List.index(alliance_client)
    if Auto_Run == 1:
        print("Player ", alliance_client, "chosen for alliance")
    if Player_List[PI] >= 6 and Player_List[client_index] >= 6:
        return Player_List, Tech_Counter, Money_List, Strategy, \
        alliance_list, alliance_resources, Starting_Time, Revert_Strategy, Strat_Choice
    ### TEST IF ALLIANCE IS MUTUAL ###
    all_prob = int((Strategy[PI][5])*(Strategy[client_index][5])/100)
#    print(all_prob)
    all_chance = np.zeros(100)
    for i in range(all_prob):
        all_chance[i] = 1
    alliance = rdm.choice(all_chance)

    if alliance == 0:
        if Auto_Run == 1:
            print("Alliance Failed")
        Revert_Strategy[PI] = Steps+10
        Strat_Choice[PI] = 0
        return Player_List, Tech_Counter, Money_List, Strategy, \
        alliance_list, alliance_resources, Starting_Time, Revert_Strategy, Strat_Choice


    else:
        for i in range(len(alliance_list)):
            ### PLAYER ALREADY IN ALLIANCE ###
            if (Player_List[PI] not in alliance_list[i]) is False:
                if Auto_Run == 1:
                    print("Initiating player already in alliance")
```

```
        alliance_player = alliance_list[i][0]
        alliance_index = Player_List.index(alliance_player)
        alliance_list[i].append(alliance_client)

        Tech_Counter[alliance_index] += round(Tech_Counter[client_index]*0.8,2)
        Money_List[alliance_index] += round(Money_List[client_index]*0.8,2)

        for j in range(len(Strategy[PI])):
            strat_sum = 0
            for k in range(len(alliance_list[i])):
                if alliance_list[i][k] == alliance_player:
                    pass
                else:
                    temp_ind = Player_List.index(alliance_list[i][k])
                    strat_sum += Strategy[temp_ind][j]
            Strategy[alliance_index][j] = strat_sum/(len(alliance_list[i])-1)


        ### Documenting client alliance contributions
        for j in range(len(alliance_resources)):
            for k in range(len(alliance_resources[j])):
                if (alliance_resources[j][k][0] not in\
                    alliance_list[i]) is False:
                    alliance_resources[j].append([alliance_client, \
                    Tech_Counter[client_index], Money_List[client_index]])
                    break
        ### Removing client resources ###
        Tech_Counter[client_index] = 0
        Money_List[client_index] = 0
        Revert_Strategy[client_index] = Steps+20
        Revert_Strategy[client_index] = 0
        return Player_List, Tech_Counter, Money_List, \
        Strategy, alliance_list, alliance_resources, Starting_Time, \
        Revert_Strategy, Strat_Choice

        #""" CLIENT ALREADY IN ALLIANCE """
    elif (Player_List[client_index] not in alliance_list[i]) is False:
        if Auto_Run == 1:
            print("Player", Player_List[client_index], "already in an alliance")
        alliance_player = alliance_list[i][0]
        alliance_index = Player_List.index(alliance_player)
        alliance_list[i].append(Player_List[PI])

        Tech_Counter[alliance_index] += round(Tech_Counter[PI]*0.6,2)
        Money_List[alliance_index] += round(Money_List[PI]*0.8,2)

        for j in range(len(Strategy[PI])):
            strat_sum = 0
            for k in range(len(alliance_list[i])):
                if alliance_list[i][k] == alliance_player:
                    pass
                else:
                    temp_ind = Player_List.index(alliance_list[i][k])
                    strat_sum += Strategy[temp_ind][j]
            Strategy[alliance_index][j] = strat_sum/(len(alliance_list[i])-1)


        ### Documenting player alliance contributions
        for j in range(len(alliance_resources)):
            for k in range(len(alliance_resources[j])):
                if (alliance_resources[j][k][0] not in\
                    alliance_list[i]) is False:
                    alliance_resources[j].append([Player_List[PI], \
                                Tech_Counter[PI], Money_List[PI]])
                    break
        ### Removing player resources ###
        Tech_Counter[PI] = 0
        Money_List[PI] = 0
        Revert_Strategy[PI] = Steps+20
        Revert_Strategy[PI] = 0
        return Player_List, Tech_Counter, Money_List, \
        Strategy, alliance_list, alliance_resources, Starting_Time, \
        Revert_Strategy, Strat_Choice
    else:
        pass
### NEW ALLIANCE ###
# Neither player nor client were in any alliance
if Auto_Run == 1:
    print("NEW ALLIANCE FORMED")
```

```
alliance_player = Player_List[-1]+1 #New alliance

alliance_list.append([alliance_player, Player_List[PI], alliance_client])
all_res_temp = []
for i in range(len(alliance_list)):
    if (alliance_player not in alliance_list[i]) is False:
        for j in range(len(alliance_list[i])):
            if alliance_list[i][j] == alliance_player:
                pass
            else:
                temp_ind = Player_List.index(alliance_list[i][j])
                all_res_temp.append([alliance_list[i][j], \
                        Tech_Counter[temp_ind], Money_List[temp_ind]])
    else:
        pass
alliance_resources.append(all_res_temp)

### ADDING ALLIANCE + RESOURCES TO LISTS ###
Player_List.append(alliance_player)
Tech_Counter.append(round((Tech_Counter[PI] + Tech_Counter[client_index])*0.8,2))
#60% of tech is transferrable

Money_List = list(Money_List)
Money_List.append(round((Money_List[PI] + Money_List[client_index])*0.8,2))
#20% fee for merger
Money_List = np.array(Money_List)

Starting_Time.append(Steps)
Revert_Strategy = list(Revert_Strategy)
Revert_Strategy.append(Steps+20)
Revert_Strategy = np.array(Revert_Strategy)
Revert_Strategy[PI] = Steps+30
Revert_Strategy[client_index] = Steps + 30
Strat_Choice = list(Strat_Choice)
Strat_Choice.append(0)
Strat_Choicey = np.array(Strat_Choice)

### ADDING NEW ALLIANCE STRATEGY ###
all_strat = np.zeros(len(Strategy[PI]))
for j in range(len(Strategy[PI])):
    all_strat[j] = (Strategy[PI][j] + Strategy[client_index][j])/2

Strategy.append(all_strat)

### REMOVE RESOURCES FROM INDIVIDUALS ###
Tech_Counter[PI] = 0
Tech_Counter[client_index] = 0
Money_List[PI] = 0
Money_List[client_index] = 0

return Player_List, Tech_Counter, Money_List, Strategy, \
alliance_list, alliance_resources, Starting_Time, Revert_Strategy, Strat_Choice
```

11.1.5 Joint Venture Separation

```
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 21 18:32:43 2017

@author: Quan
"""
from numpy import *

#==============================================================================
# Joint Venture Separation
#==============================================================================


def Alliance_Split(PI, Player_List, alliance_list, alliance_resources, Strategy, \
                   Revert_Strategy,Strat_Choice, Tech_Counter, \
                   Money_List, Tech_Cap, Starting_Time, Steps, alliance_del, Auto_Run):

    for i in range(len(alliance_list)):
        if (Player_List[PI] not in alliance_list[i]) is False:
            alliance_player = alliance_list[i][0]
            alliance_index = Player_List.index(alliance_player)

    #        print(alliance_player)
            if len(alliance_list[i]) == 3:
                if alliance_list[i][1] == Player_List[PI]:
                    Player1 = copy(alliance_list[i][1])
                    Player2 = copy(alliance_list[i][2])
                else:
                    Player1 = copy(alliance_list[i][2])
                    Player2 = copy(alliance_list[i][1])
                p1_ind = Player_List.index(Player1)
                p2_ind = Player_List.index(Player2)


                for j in range(len(alliance_resources)):
                    for k in range(len(alliance_resources[j])):
                        if alliance_resources[j][k][0] == Player1:
                            tech1 = alliance_resources[j][k][1]
                            money1 = alliance_resources[j][k][2]
                            temp1 = j
                        elif alliance_resources[j][k][0] == Player2:
                            tech2 = alliance_resources[j][k][1]
                            money2 = alliance_resources[j][k][2]

                Tech_Counter[p1_ind] = round(Tech_Counter[alliance_index]*\
                            (tech1/(tech1+tech2)),2)
                Money_List[p1_ind] = round(Money_List[alliance_index]*\
                          (money1/(money1+money2)),2)
                Tech_Counter[p2_ind] = round(Tech_Counter[alliance_index]*\
                            (tech2/(tech1+tech2)),2)
                Money_List[p2_ind] = round(Money_List[alliance_index]*\
                          (money2/(money1+money2)),2)

                Revert_Strategy[PI] = Steps+20
                Strat_Choice[PI] = 0

                del alliance_resources[temp1]

                Revert_Strategy = list(Revert_Strategy)
                del Revert_Strategy[alliance_index]
                Revert_Strategy = array(Revert_Strategy)

                Strat_Choice = list(Strat_Choice)
                del Strat_Choice[alliance_index]
                Strat_Choice = array(Strat_Choice)

                Money_List = list(Money_List)
                del Money_List[alliance_index]
                Revert_Strategy = array(Money_List)


                del Player_List[alliance_index]
                del Tech_Counter[alliance_index]
                del Starting_Time[alliance_index]
                alliance_del.append(alliance_player)
                del Strategy[alliance_index]
```

```
            del alliance_list[i]
            return Tech_Counter, Money_List, Player_List, Strategy, \
        Revert_Strategy, Strat_Choice, Starting_Time, alliance_resources,\
        alliance_list , alliance_del
        else:
            for j in range(len(alliance_resources)):
#                print(j)
                for k in range(len(alliance_resources[j])):
#                    print(k)
                    if alliance_resources[j][k][0] == Player_List[PI]:
                        res_index1 = j
                        res_index2 = k
                        break
                    else:
                        pass
            tech_sum = 0
            money_sum = 0
            for j in range(len(alliance_resources[res_index1])):
#                print(j)
                tech_sum += alliance_resources[res_index1][j][1]
                money_sum += alliance_resources[res_index1][j][2]
                if alliance_resources[res_index1][j][0] == Player_List[PI]:
                    Player_tech = alliance_resources[res_index1][j][1]
                    Player_money = alliance_resources[res_index1][j][2]
#                    del alliance_resources[all_index1][j]
                else:
                    pass

            Tech_Counter[PI] = round(Player_tech*Tech_Counter[alliance_index]/tech_sum,2)
            Money_List[PI] = round(Player_money*Money_List[alliance_index]/money_sum,2)

            Tech_Counter[alliance_index] -= Tech_Counter[PI]
            Money_List[alliance_index] -= Money_List[PI]

            Revert_Strategy[PI] = Steps+20
            Strat_Choice[PI] = 0

            for j in range(len(Strategy[PI])):
                Strategy[alliance_index][j] = ((Strategy[alliance_index][j]*\
                        (len(alliance_list[i])-1)) - Strategy[PI][j])/\
                        ((len(alliance_list[i])) - 2)

            for j in range(len(alliance_list[i])):
                if alliance_list[i][j] == Player_List[PI]:
                    Del = j

            del alliance_resources[res_index1][res_index2]
            del alliance_list[i][Del]
            return Tech_Counter, Money_List, Player_List, Strategy, \
        Revert_Strategy, Strat_Choice, Starting_Time, alliance_resources,\
        alliance_list, alliance_del
    else:
        pass
```

11.2 Game Simulation User Guide

Technology Strategy as a Partially Adversarial Game

Table of Contents

----

(1) INTRODUCTION

A growing company needs a strong strategy to ensure success and allow for growth.
The strategies used for technological development are vital to the infrastructure
and growth of a company, but when other competitive companies become threats, there
is wide debate on the recommended actions of the growing company. Competing and
cooperating with competitive companies both have pros and cons, but one must be
better than the other in specific scenarios.

----

(2) SYSTEM REQUIREMENTS

-Spyder Interface running Python 3.6 or later
-Keyboard
-Mouse

----

(3) INSTALLATION GUIDE

Unzip game file
Run the Game.py file in Spyder

----

(4) HOW TO PLAY

A) Choose to play default or custom settings

Custom Settings:
Choose Number of Players
Choose Tech Cap
Choose Player Strategy (HI, PA, AQ, PL, FL, JV)
Choose Starting Time

B) Choose Auto Run

C) Press Enter to Run through program if Auto Run is disabled

D) Press 1 then Enter to stop the simulation

----

(5) STRATEGY BREAKDOWN

NS - No Strategy
Player doesnt act. if it is a tech upgrade turn, player will buy a TP

HI - Higher Investment
Player will spend more money to instantly buy a TP.
Recharge after 10 turns

PA - Patent
Player will spend money to buy a patent. patent will reduce the TP of every other player.
Recharge after 20 turns

AQ - Acquisition
Player will attempt to buyout another player. if successful, the acquirer will gain some of the target's TP. Target will gain 100% of the purchase cost.

PL - Paid License
Player will release a paid license, allowing others to buy the product and provide feedback. Increases TP and reduce money cost.

FL - Free License
Player will release a free license, allowing others to buy the product for free and provide feedback.
Increases TP greatly at higher cost.

JV - Joint Venture
Enter a joint venture with another player. Choice of player is based on TP and money, and chosen at random. The chosen player has an option to decline the invitation. If accepted, both players join together and create a new player. TP and money will be combined with small losses

JVS _ Joint Venture Separation
Once in the joint venture, either player is able to leave the JV when desired. If a player leaves, they will take their fraction of resources from the alliance. The alliance will hold if there are at least 2 remaining players. Otherwise the alliance will be disbanded and the resources distributed accordingly

----

(6) TROUBLESHOOTING

if problems occur in the code, try closing and re-running the program. if problems persists, check player strategies and make sure the total is less than 100.

----

(7) CONTACT DETAILS

Contact Quan Tran at:
quan.tran2@uqconnect.edu.au

11.3 Multiple Player Test Results

11.3.1 Number of Games Won

Table 11: Multiple Player Test: Number of Games Won

			Player 2 Strategy											
			HI		PA		AQ		PL		FL		JV	
			25	50	5	15	10	30	10	50	10	50	25	50
Player 1 Strategy	HI	25	58|60	85|30										
		50	76|36	54|60										
	PA	5	100|0	100|0	51|49	5|95								
		15	100|0	100|0	84|16	42|58								
	AQ	10	99|1	100|1	31|69	36|64	69|61	69|48						
		30	96|8	99|1	22|78	30|70	52|68	61|58						
	PL	10	100|0	100|0	100|0	88|13	97|3	97|3	70|71	38|94				
		50	100|0	100|0	100|0	100|0	100|0	100|0	95|45	65|62				
	FL	10	100|0	100|0	59|44	27|73	100|0	100|0	99|2	99|8	57|65	38|85		
		50	100|0	100|0	71|31	31|70	100|0	100|0	100|0	100|0	90|22	72|76		
	JV	25	97|12	94|9	1|100	0|100	59|86	58|62	0|100	0|100	0|100	0|100	31|30|56	6|3|92
		50	90|16	89|14	0|100	0|100	41|90	53|66	0|100	0|100	0|100	0|100	4|10|90	0|0|100

11.3.2 Average Steps Taken

Table 12: Multiple Player Test: Average Steps Taken

			Player 2 Strategy											
			HI		PA		AQ		PL		FL		JV	
			25	50	5	15	10	30	10	50	10	50	25	50
Player 1 Strategy	HI	25	209.38	210.31										
		50	210.39	212.29										
	PA	5	220.5	218.2	486.7	569.4								
		15	237	236.8	614.8	624.4								
	AQ	10	174.3	173.98	212.27	227.6	176.85	175.46						
		30	179.49	172.48	214.94	232.83	176.22	178.71						
	PL	10	132.2	131.3	95.2	119.7	133.5	131.8	128	119.9				
		50	119.9	121.7	76.1	79.7	119.2	119.7	121.1	115.6				
	FL	10	87.2	91.7	183.6	212.4	87	87	87.3	87.5	80	72.8		
		50	73.6	74.4	166.5	209.4	74.9	74.7	73.9	74.6	73.8	71.8		
	JV	25	197.91	198.93	220.5	235.9	187.76	185.09	133	118.7	90.5	75	218.97	249.79
		50	197.86	198.85	238	236.8	185.69	185.59	131	120.4	88.3	73.8	246.46	483.81

11.3.3 Average Money Earnt

Table 13: Multiple Player Test: Average Money Earnt (rounded to the nearest dollar)

<table>
<tr><th></th><th></th><th></th><th colspan="12">Player 2 Strategy</th></tr>
<tr><th></th><th></th><th></th><th colspan="2">HI</th><th colspan="2">PA</th><th colspan="2">AQ</th><th colspan="2">PL</th><th colspan="2">FL</th><th colspan="2">JV</th></tr>
<tr><th></th><th></th><th></th><th>25</th><th>50</th><th>5</th><th>15</th><th>10</th><th>30</th><th>10</th><th>50</th><th>10</th><th>50</th><th>25</th><th>50</th></tr>
<tr><td rowspan="12">Player 1 Strategy</td><td rowspan="2">HI</td><td>25</td><td>359|359</td><td>362|358</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>50</td><td>358|361</td><td>363|363</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">PA</td><td>5</td><td>385|285</td><td>380|283</td><td>755|751</td><td>721|957</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>15</td><td>420|271</td><td>419|271</td><td>1013|807</td><td>878|917</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">AQ</td><td>10</td><td>271|246</td><td>270|252</td><td>300|301</td><td>289|319</td><td>269|265</td><td>268|248</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>30</td><td>277|249</td><td>262|228</td><td>281|330</td><td>281|341</td><td>251|267</td><td>268|264</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">PL</td><td>10</td><td>214|166</td><td>213|162</td><td>150|99</td><td>195|152</td><td>213|167</td><td>216|154</td><td>203|203</td><td>183|192</td><td></td><td></td><td></td><td></td></tr>
<tr><td>50</td><td>193|140</td><td>197|143</td><td>112|65</td><td>119|72</td><td>191|136</td><td>193|130</td><td>194|186</td><td>180|181</td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">FL</td><td>10</td><td>133|84</td><td>144|90</td><td>294|292</td><td>312|358</td><td>136|79</td><td>135|71</td><td>135|105</td><td>134|112</td><td>115|116</td><td>102|108</td><td></td><td></td></tr>
<tr><td>50</td><td>107|61</td><td>110|62</td><td>272|250</td><td>324|352</td><td>108|60</td><td>109|54</td><td>107|77</td><td>110|81</td><td>104|99</td><td>102|102</td><td></td><td></td></tr>
<tr><td rowspan="2">JV</td><td>25</td><td>326|325</td><td>328|327</td><td>276|385</td><td>260|418</td><td>297|303</td><td>289|284</td><td>165|218</td><td>134|190</td><td>86|144</td><td>60|111</td><td>256|257|50</td><td>181|174|108</td></tr>
<tr><td>50</td><td>322|327</td><td>327|326</td><td>255|422</td><td>250|419</td><td>274|300</td><td>277|286</td><td>155|213</td><td>135|195</td><td>78|138</td><td>56|109</td><td>171|178|114</td><td>118|120|346</td></tr>
</table>

11.4 Simulation Payoff Matrix

11.4.1 No Strategy

Table 14: General Payoff Matrix for NS

			Player 2(No Strategy)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	T_1+1	T_2+1	0
		Money	$M_1-T_1^{1.5}$	$M_2-T_2^{1.5}$	0
	Higher Investment	TP	T_1+1	T_2+1	0
		Money	$M_1 - 1.5*T_1**^{1.5}$	$M_2-T_2^{1.5}$	0
	Patent Application	TP	T_1+1	$\frac{T_2}{2}+1$	0
		Money	M_1-25	$M_2-T_2^{1.5}$	0
	Acquisition	TP	$T_1+\frac{T_2}{2}$	0	0
		Money	$M_1 - M_2 - 2*T_2$	$2*M_2 + 2*T_2$	0
	Paid License	TP	T_1+2	T_2+1	0
		Money	$M_1 - 0.5*T_1^{1.5}$	$M_2-T_2^{1.5}$	0
	Free License	TP	T_1+4	T_2+1	0
		Money	$M_1 - 1.5*T_1^{1.5}$	$M_2-T_2^{1.5}$	0
	Joint Venture	TP	0	0	$(T_1+T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Joint Venture Separation	TP	$T_3*\frac{T_1}{T_1+T_2}$	$T_3*\frac{T_2}{T_1+T_2}$	0
		Money	$M_3*\frac{M_1}{M_1+M_2}$	$M_3*\frac{M_2}{M_1+M_2}$	0

11.4.2 Higher Investments

Table 15: General Payoff Matrix for HI

			Player 2(Higher Investment)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	$T_1 +1$	T_2+2	0
		Money	$M_1 - T_1^{1.5}$	$M_2-2.5 * T_2^{1.5}$	0
	Higher Investment	TP	$T_1 +1$	T_2+2	0
		Money	$M_1 - 1.5*T_1^{1.5}$	$M_2-2.5 * T_2^{1.5}$	0
	Patent Application	TP	T_1+1	$\frac{T_2}{2} +2$	0
		Money	M_1-25	$M_2-2.5 * T_2^{1.5}$	0
	Acquisition	TP	$T_1+ \frac{T_2}{2}$	0	0
		Money	$M_1 – M_2 – 2*T_2$	$2*M_2 + 2*T_2$	0
	Paid License	TP	T_1+2	T_2+2	0
		Money	$M_1 - 0.5*T_1^{1.5}$	$M_2-2.5 * T_2^{1.5}$	0
	Free License	TP	T1+4	T_2+2	0
		Money	$M_1 - 1.5*T_1^{1.5}$	$M_2-2.5 * T_2^{1.5}$	0
	Joint Venture	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Joint Venture Separation	TP	$T_3*\frac{T_1}{T_1+T_2}$	$T_3*\frac{T_2}{T_1+T_2}$	0
		Money	$M_3*\frac{M_1}{M_1+M_2}$	$M_3*\frac{M_2}{M_1+M_2}$	0

11.4.3 Patents

Table 16: General payoff Matrix for PA

			Player 2(Patents)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	$\frac{T_1}{2}$+1	T_2+1	0
		Money	M_1 - $T_1^{1.5}$	M_2-25	0
	Higher Investment	TP	$\frac{T_1}{2}$ +1	T_2+1	0
		Money	M_1 - 1.5*$T_1^{1.5}$	M_2-25	0
	Patent Application	TP	$\frac{T_1}{2}$+1	$\frac{T_2}{2}$ +1	0
		Money	M_1-25	M_2-25	0
	Acquisition	TP	$\frac{T_1}{2}+\frac{T_2}{2}$	0	0
		Money	M_1 – M_2 – 2*T_2	2*M_2 + 2*T_2	0
	Paid License	TP	$\frac{T_1}{2}$+2	T_2+1	0
		Money	M_1 - 0.5*$T_1^{1.5}$	M_2-25	0
	Free License	TP	$\frac{T_1}{2}$+4	T_2+1	0
		Money	M_1 - 1.5*$T_1^{1.5}$	M_2-25	0
	Joint Venture	TP	0	0	(T_1+ T_2)*0.8
		Money	0	0	(M_1+M_2)*0.8
	Joint Venture Separation	TP	T_3*$\frac{T_1}{T_1+T_2}$	T_3*$\frac{T_2}{T_1+T_2}$	0
		Money	M_3*$\frac{M_1}{M_1+M_2}$	M_3*$\frac{M_2}{M_1+M_2}$	0

11.4.4 Acquisition

Table 17: General Payoff Matrix for AQ

			Player 2(Acquisition)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	0	$T_2 + \frac{T_1}{2}$	0
		Money	$2*M_1 + 2*T_1$	M_2-25	0
	Higher Investment	TP	0	$T_2 + \frac{T_1}{2}$	0
		Money	$2*M_1 + 2*T_1$	M_2-25	0
	Patent Application	TP	0	$T_2 + \frac{T_1}{2}$	0
		Money	$2*M_1 + 2*T_1$	M_2-25	0
	Acquisition	TP	-	-	0
		Money	-	-	0
	Paid License	TP	0	$T_2 + \frac{T_1}{2}$	0
		Money	$2*M_1 + 2*T_1$	M_2-25	0
	Free License	TP	0	$T_2 + \frac{T_1}{2}$	0
		Money	$2*M_1 + 2*T_1$	M_2-25	0
	Joint Venture	TP	0	T_3	0
		Money	0	M_3	0
	Joint Venture Separation	TP	0	T_3	0
		Money	0	M_3	0

11.4.5 Paid License

Table 18: General Payoff Matrix for PL

			Player 2(Paid License)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	$T_1 + 1$	$T_2 + 2$	0
		Money	$M_1 - T_1^{1.5}$	$M_2 - 0.5*T_2^{1.5}$	0
	Higher Investment	TP	$T_1 + 1$	$T_2 + 2$	0
		Money	$M_1 - 1.5*T_1^{1.5}$	$M_2 - 0.5*T_2^{1.5}$	0
	Patent Application	TP	$T_1 + 1$	$\frac{T_2}{2} + 2$	0
		Money	$M_1 - 25$	$M_2 - 0.5*T_2^{1.5}$	0
	Acquisition	TP	$T_1 + \frac{T_2}{2}$	0	0
		Money	$M_1 - M_2 - 2*T_1$	$2*M_2 + 2*T_1$	0
	Paid License	TP	$T_1 + 2$	$T_2 + 2$	0
		Money	$M_1 - 0.5*T_1^{1.5}$	$M_2 - 0.5*T_2^{1.5}$	0
	Free License	TP	$T_1 + 4$	T_2+2	0
		Money	$2*M_1 + 2*T_1$	$M_2 - 0.5*T_2^{1.5}$	0
	Joint Venture	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Joint Venture Separation	TP	$T_3*\frac{T_1}{T_1+T_2}$	$T_3*\frac{T_2}{T_1+T_2}$	0
		Money	$M_3*\frac{M_1}{M_1+M_2}$	$M_3*\frac{M_2}{M_1+M_2}$	0

11.4.6 Free License

Table 19: General Payoff Matrix for FL

			Player 2(Free License)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	$T_1 + 1$	$T_2 + 4$	0
		Money	$M_1 - T_1^{1.5}$	$M_2 - 1.5*T_2^{1.5}$	0
	Higher Investment	TP	$T_1 + 1$	$T_2 + 4$	0
		Money	$M_1 - 1.5*T_1^{1.5}$	$M_2 - 1.5*T_2^{1.5}$	0
	Patent Application	TP	$T_1 + 1$	$\frac{T_2}{2} + 4$	0
		Money	$M_1 - 25$	$M_2 - 1.5*T_2^{1.5}$	0
	Acquisition	TP	$T_1 + \frac{T_2}{2}$	0	0
		Money	$M_1 - M_2 - 2*T_1$	$2*M_2 + 2*T_1$	0
	Paid License	TP	$T_1 + 2$	$T_2 + 4$	0
		Money	$M_1 - 0.5*T_1^{1.5}$	$M_2 - 1.5*T_2^{1.5}$	0
	Free License	TP	$T_1 + 4$	T_2+4	0
		Money	$M_1 - 1.5*T_1^{1.5}$	$M_2 - 1.5*T_2^{1.5}$	0
	Joint Venture	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Joint Venture Separation	TP	$T_3*\frac{T_1}{T_1+T_2}$	$T_3*\frac{T_2}{T_1+T_2}$	0
		Money	$M_3*\frac{M_1}{M_1+M_2}$	$M_3*\frac{M_2}{M_1+M_2}$	0

11.4.7 Joint Venture

Table 20: General Payoff Matric for JV

			Player 2(Joint Venture)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Higher Investment	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Patent Application	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Acquisition	TP	T_3	0	0
		Money	M_3	0	0
	Paid License	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Free License	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Joint Venture	TP	0	0	$(T_1+ T_2)*0.8$
		Money	0	0	$(M_1+M_2)*0.8$
	Joint Venture Separation	TP	$T_3*\frac{T_1}{T_1+T_2}$	$T_3*\frac{T_2}{T_1+T_2}$	0
		Money	$M_3*\frac{M_1}{M_1+M_2}$	$M_3*\frac{M_2}{M_1+M_2}$	0

11.5 Example Payoff Matrix

11.5.1 No Strategy

Table 21: Example Payoff Matrix for NS

			Player 2 (No Strategy)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	5	7	0
		Money	22	4.3	0
	Higher Investment	TP	5	7	0
		Money	12	4.3	0
	Patent Application	TP	5	4.5	0
		Money	5	4.3	0
	Acquisition	TP	7	0	0
		Money	-1	50	0
	Paid License	TP	6	7	0
		Money	26	4.3	0
	Free License	TP	8	7	0
		Money	18	4.3	0
	Joint Venture	TP	0	0	7.2
		Money	0	0	34.4
	Joint Venture Separation	TP	3.2	4.8	0
		Money	24	15.2	0

11.5.2 Higher Investments

Table 22: Example Payoff Matrix for HI

			Player 2 (Higher Investment)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	5	7	0
		Money	22	-3.045	0
	Higher Investment	TP	5	7	0
		Money	12	-3.045	0
	Patent Application	TP	5	4	0
		Money	5	-3.045	0
	Acquisition	TP	7	0	0
		Money	-1	50	0
	Paid License	TP	6	7	0
		Money	26	-3.045	0
	Free License	TP	8	7	0
		Money	18	-3.045	0
	Joint Venture	TP	0	0	8
		Money	0	0	39.2
	Joint Venture Separation	TP	3.2	4.8	0
		Money	24	15.2	0

11.5.3 Patents

Table 23: Example Payoff Matrix for Patents

			Player 2 (Patent)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	3	7	0
		Money	22	-6	0
	Higher Investment	TP	3	7	0
		Money	12	-6	0
	Patent Application	TP	3	4	0
		Money	5	-6	0
	Acquisition	TP	7	0	0
		Money	-1	50	0
	Paid License	TP	4	7	0
		Money	26	-6	0
	Free License	TP	6	7	0
		Money	18	-6	0
	Joint Venture	TP	0	0	8
		Money	0	0	39.2
	Joint Venture Separation	TP	3.2	4.8	0
		Money	24	15.2	0

11.5.4 Acquisition

Table 24: Example Payoff Matrix for AQ

			Player 2 (Acquisition)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	0	8	0
		Money	68	-19	0
	Higher Investment	TP	0	8	0
		Money	68	-19	0
	Patent Application	TP	0	8	0
		Money	68	-19	0
	Acquisition	TP	-	-	0
		Money	-	-	0
	Paid License	TP	0	8	0
		Money	68	-19	0
	Free License	TP	0	8	0
		Money	68	-19	0
	Joint Venture	TP	0	0	8
		Money	0	0	39.2
	Joint Venture Separation	TP	3.2	4.8	0
		Money	24	15.2	0

11.5.5 Paid License

Table 25: Example Payoff Matrix for PL

			Player 2 (Paid License)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	5	8	0
		Money	22	11.652	0
	Higher Investment	TP	5	8	0
		Money	12	11.652	0
	Patent Application	TP	5	5	0
		Money	5	11.652	0
	Acquisition	TP	7	0	0
		Money	-1	50	0
	Paid License	TP	6	8	0
		Money	26	11.652	0
	Free License	TP	8	8	0
		Money	18	11.652	0
	Joint Venture	TP	0	0	8
		Money	0	0	39.2
	Joint Venture Separation	TP	3.2	4.8	0
		Money	24	15.2	0

11.5.6 Free License

Table 26: Example Payoff Matrix for FL

			Player 2 (Free License)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	5	10	0
		Money	22	-3.045	0
	Higher Investment	TP	5	10	0
		Money	12	-3.045	0
	Patent Application	TP	5	7	0
		Money	5	-3.045	0
	Acquisition	TP	7	0	0
		Money	-1	50	0
	Paid License	TP	6	10	0
		Money	26	-3.045	0
	Free License	TP	8	10	0
		Money	18	-3.045	0
	Joint Venture	TP	0	0	8
		Money	0	0	39.2
	Joint Venture Separation	TP	3.2	4.8	0
		Money	24	15.2	0

11.5.7 Joint Venture

Table 27: Example Payoff Matrix for JV

			Player 2 (Joint Venture)		
			Player 1	Player 2	Alliance
Player 1	No Strategy	TP	0	0	8
		Money	0	0	39.2
	Higher Investment	TP	0	0	8
		Money	0	0	39.2
	Patent Application	TP	0	0	8
		Money	0	0	39.2
	Acquisition	TP	8	0	0
		Money	39.2	0	0
	Paid License	TP	0	0	8
		Money	0	0	39.2
	Free License	TP	0	0	8
		Money	0	0	39.2
	Joint Venture	TP	0	0	8
		Money	0	0	39.2
	Joint Venture Separation	TP	3.2	4.8	0
		Money	24	15.2	0