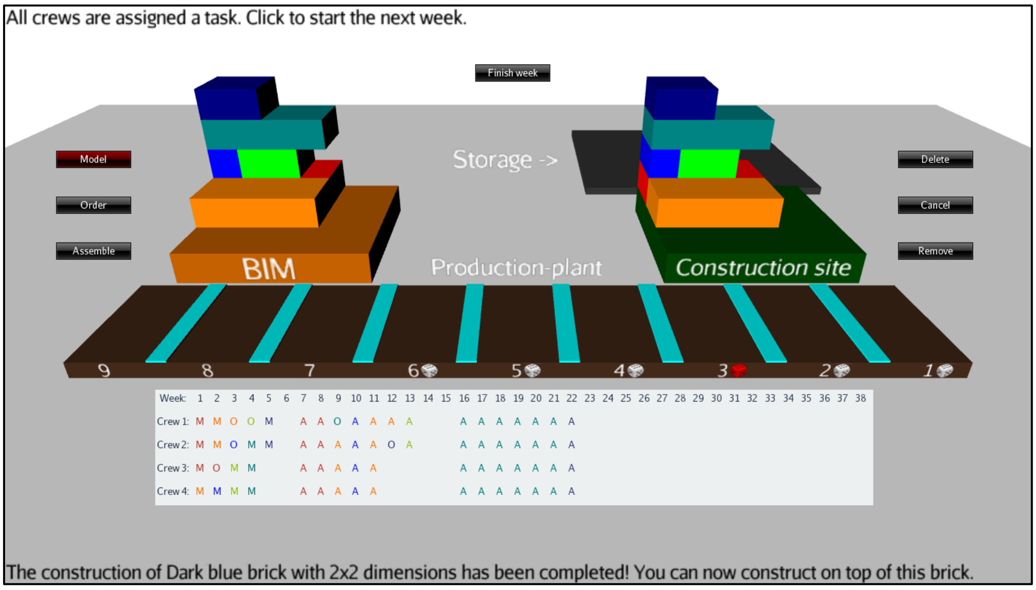
University of Twente

Developing and testing a high-tech serious game "Tower of Infinity" for supply chain management

Bachelor Thesis



Author: B.H.G. (Bram) Schrooten
Student number: s1598392
E-Mail: b.h.g.schrooten@student.utwente.nl
1st supervisor: M. C. van den Berg
2nd supervisor: J. T. Voordijk

Date: *07-07-2017*

Abstract
The field of supply chain management in construction is lagging behind in regard to other fields, due to the unique characteristics that this field owns. As an attempt to create insight in this field, a serious board game called 'Tower of Infinity' has been developed using Triadic Game Design. However, a digital game might offer more possibilities than a traditional board game. A high tech version of 'Tower of Infinity' will be developed and tested, where the strong aspects of digital games will be optimized. Triadic Game Design and Agile Software Development have been used to develop the game, jMonkey has been used for programming the game, and the testing will be done in a workshop with multiple experts.

Out of the game- and literature-analysis, it became clear which characteristics were already present in the low-tech version of "Tower of Infinity", where the interface between design, production and construction has a significant priority. These aspects have been implemented into the first version of the high-tech game, where only one aspect was not possible to translate into the high-tech version due to its single-player restriction. Out of feedback-sessions, it became clear that the usability was deficient. In the final version of the game, this has been improved. The first high-tech version of "Tower of Infinity" has been used in a presentation during a knowledge-table, after which a questionnaire was handed out that questioned the attendees about the functionality of the game. The response from this questionnaire was mostly positive. From this, it can be said that the high-tech game has more possibilities to gain insight from: through playing the game, but also through a presentation as a discussion starter.

This research hopes to add to the field of Civil Engineering by creating a high-tech game that possesses more possibilities than the low-tech game. Additionally, this research can also be used as a framework for translating a low-tech game into a high-tech version.

Table of Contents

1. Introduction

In this thesis, a high-tech version of the serious game "Tower of Infinity" will be developed and tested. In this introductory part, the problem is briefly explained. For a clear description of different definitions that are used in this report, please refer to Appendix A.

Instead of dividing the supply chain, supply chain management (SCM) views the entire supply chain, and aims to increase transparency and alignment of the supply chain's co-ordination and configuration (Vrijhoef & Koskela, 2000). However, in the field of construction supply chain management, productivity did not increase significantly around 2000, while the costs of construction objects had (Vrijhoef & Koskela, 2000). Because every supply chain in construction management is unique (Segerstedt & Olofsson, 2010), there is not one optimal supply chain approach for every construction. Due to its unique nature and the risks of costs and time of real-life testing, easy and cheap ways to gain insight into SCM in construction must be created to add to the productivity of the construction industry.

Serious gaming is one of the solutions for this problem. A serious game about SCM in construction, dubbed "Tower of Infinity", has already been created (van den Berg M. C., 2015). However, at the moment this game is exclusively available as a board-game (low-tech version). The fact is however, that low-tech games have certain disadvantages in contrast to high-tech games (computer games). High-tech games have high plasticity, possibility to function as a referee and high speed of calculating administrative matters (Crawford, 1984). These factors show that a high-tech game has more possibilities than a low-tech game. However, no knowledge is available on high-tech serious games about SCM in construction, or how to translate such a low-tech game into a high-tech version. Because of this, a high-tech version of "Tower of Infinity" should be developed and tested, so users of this game are able to gain more insight into construction supply chain management. Furthermore, it will allow game-developers to gain insight in translating low-tech games into high-tech games.

2. Research methodology

In this part, the base of the research is given. The aim of the research, the main question, sub-questions and methods are all given in this chapter.

2.1. Research aim

The aim of this research is to analyse how a high-tech version of "Tower of Infinity" can create insight in supply chain management for practitioners that work in the construction industry. Advantages of high-tech games as opposed to low-tech games will be analysed and these characteristics will be implemented in the game. At first, only the current base of the board game will be implemented. Then, further analysis will be done on what the current version is still missing.

2.2. Research questions

In short, the following research question has to be answered:

> **How do users of a high-tech game "Tower of Infinity" become more aware of different problems of the field of supply chain management in construction?**

This can be divided into several different sub-questions:

1. **What are the characteristics of construction supply chain management?**
2. **How are problem areas already implemented in the low-tech version of "Tower of Infinity"?**
3. **What advantages and disadvantages do high-tech games have in regard to low-tech games?**
4. **How can the high-tech version of "Tower of Infinity" be developed?**
5. **How can "Tower of Infinity" create awareness for its users about the problems of SCM in construction?**

2.3. Research methods

To design the high-tech version of "Tower of Infinity", the game engine jMonkey will be used. Game engines are toolkits aimed to ease the development of videogames, acting as a superstructure of several development efforts (Navarro, Pradilla, & Rios, 2012). Because jMonkey is written in Java, it has many possibilities in the desktop world, allowing development for different platforms. The only complaint of Navarro was that jMonkey did not support Android devices yet. However, they now do support Android and iOS devices (jMonkeyEngine, 2017). There are also no plans to create a mobile version of "Tower of Infinity". Additionally, jMonkey has an active online community, where answers are answered fairly quickly.

Sub-question 1 will be answered by use of literature research and sub-question 2 will be answered using the current low-tech version of "Tower of Infinity". The different characteristics of SCM in construction will be analysed using the literature, and so these characteristics can be analysed in the game. Sub-question 3 will also be answered using primarily available literature, that is, if information about this comparison is available. If this is not available, Participants of the workshops – from sub-question 5 – will be asked what they found advantageous about the high-tech version. Sub-question 4 will be answered using game-design literature, combined with brainstorming. While literature does give a great indication in which direction one should go with designing an aspect into a game, there should still be a degree of freedom for creativity in the way that the game is designed. Sub-question 5 will be answered using surveys during the workshops. In these surveys, the participants are asked what they found useful about the serious game and in what aspects they have gained more insight. Of course, there is also room for feedback and other comments. This question is to check whether the high-tech version of "Tower of Infinity" actually has an added value to SCM in construction.

3. Theoretical background

In this research, two main fields are analysed: Supply chain management in construction and serious gaming. The theoretical background contains summaries of all researches that are used for this thesis.

3.1. Supply chain management in construction

According to Christopher (as stated in Vrijhoef & Koskela, 2000), the supply chain has been defined as "the network of organisations that are involved, through upstream and downstream linkages, in the different processes and activities that produce value in the form of products and services in the hands of the ultimate customers". Instead of dividing this network, supply chain management (SCM) views the entire supply chain, and aims to increase transparency and alignment of the supply chain's co-ordination and configuration (Vrijhoef & Koskela, 2000). A generic visualisation of SCM is given in Figure 1. For example, if a customer demands a certain item, this demand will 'flow' backwards and show up at the retailer. This retailer knows how to create this item, but needs other supplies to create the demanded item for the customer. The demand changes to what the retailer needs, and flows further back in the supply chain. This repeats until the demand ends up at the end of the supply chain, and the process moves forward again, towards the earlier mentioned customer. Because these supply chains consist of so many different organizations, it is crucial that management between these different organizations goes as smooth as possible.

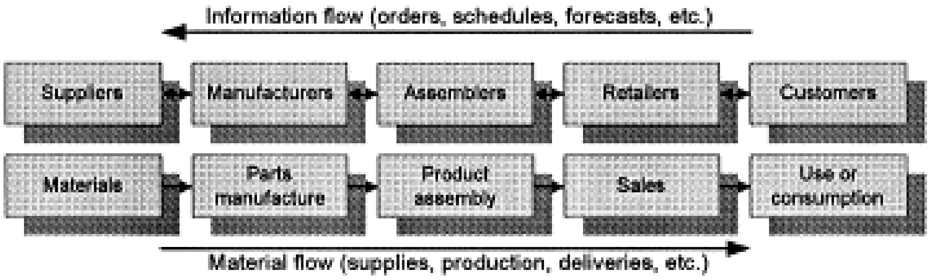


Figure 1: Generic configuration of a supply chain in manufacturing (Vrijhoef & Koskela, 2000)

SCM is a concept that originated and flourished in the manufacturing industry. The main goal of these systems was to drastically decrease inventories, and to effectively regulate the suppliers' interaction with the production line. Deming (as stated in Vrijhoef & Koskela, 2000) also suggested that working with the supplier as a partner in a long-term relationship of loyalty and trust would improve the quality and decrease the costs of production. Because of these emergences, the concept of SCM did not only become fundamental in the industrial management theory, but also became a distinct subject of scientific research (Vrijhoef & Koskela, 2000). Since then, SCM has become a concept that is used in many different fields.

When comparing SCM to the traditional approach, it can be seen that supply chain management focuses on a long-term horizon, with an on-going joint planning (Cooper & Ellram, 1993). Supply chains consist of three primary structural aspects: the distinction between members of the supply chain (primary and secondary members), the structural dimensions of the network (vertical and horizontal) and the type of business process links ((not-)managed, monitored, non-member process links). The management of these supply chains do not only consist of physical and technical components, but they also possess managerial and behavioural components (Lambert, Cooper, & Pagh, 1998). These characteristics are all elaborated on in Chapter 4.1.

In this research, the focus lies on supply chain management in construction. Eccles (as stated in Segerstedt & Olofsson, 2010) defined construction as "the erection, maintenance, and repair of immobile structures, the demolition of existing structures, and land development". Some papers claim that the annual productivity increase in building construction – only 3% in the Netherlands – is not enough to compensate for the average increase on labour costs (Vrijhoef & Koskela, 2000). However, another paper claims that these kind of investigations for SCM in construction may be debatable, due to uniqueness in output that is produced, firm size and use of technology in different sectors. These differences make it hard to characterize and measure the performance of construction on industry level (Segerstedt & Olofsson, 2010). Nevertheless, the papers do both agree that increase of productivity in the construction industry is still possible.

However, the construction industry differs significantly from industries like the manufacturing industry, because of their one-of-a-kind products, temporary organization and on-site production. Additionally, only 25% of the product's value is built solely by the principal construction company (Segerstedt & Olofsson, 2010). For these reasons, the construction industry must find other ways to improve performance and efficiency. Vrijhoef & Koskela (2010) characterized the construction supply chain in terms of structure and function as the following:

- A converging supply chain where all materials are directed at the construction site where the project is assembled from all incoming materials. In contrast to the manufacturing industry, the specific "construction factory" is set up around a single product instead of multiple products;
- It is in most cases only a temporary supply chain producing one-off construction projects. This means that every construction supply chain has a different structure, and so is characterized by instability, fragmentation and separation between the design and the actual construction of the object;
- It is a typical make-to-order supply chain: every project creates a different product or prototype. Apart from rare exceptions, there is little repetition in these supply chains. However, in particular cases the process can be very similar.

Because of these characteristics, combined with the risk of losing valuable time and money if wrong decisions are made, this field is currently inefficient and full of risks.

3.2. Serious Gaming

One of the possibilities to gain insight into this field are so-called "serious games". Serious games are games that are used for purposes other than mere entertainment. These games "allow learners to experience situations that are impossible in the real world for reasons of safety, cost, time, etc., but they are also claimed to have positive impacts on the players' development of a number of different skills" (Susi, Johannesson, & Backlund, 2007). The biggest difference between serious games and 'classic' games is that serious games have the addition of pedagogy: activities that educate or instruct the player, and so teach the player something they can use in real life. Before such a serious game can be developed, a design approach has to be chosen.

In every field of work where design is involved, it is important to consider certain methods, techniques procedures, theories, and approaches. Triadic Game Design (TGD) is a game design approach that is specifically oriented at serious games (Harteveld, 2011). At the same time, TGD must be seen as a design philosophy: it does not give a fixed roadmap on how to work on a design, but it does give a certain way of thinking: how one should look at their design, and what kind of aspects it needs. In TGD, the focus lies on the balance between the worlds of Meaning (one should learn something from it), Reality (it should have a link with reality) and Play (it should be fun): these three aspects are equally important, and they constitute the design space of developing a game. If one of these worlds is missing in the final design, the game collapses due to its unfulfilled purpose (Harteveld, 2011). A visualisation of this triadic game design approach in given in Figure 2.

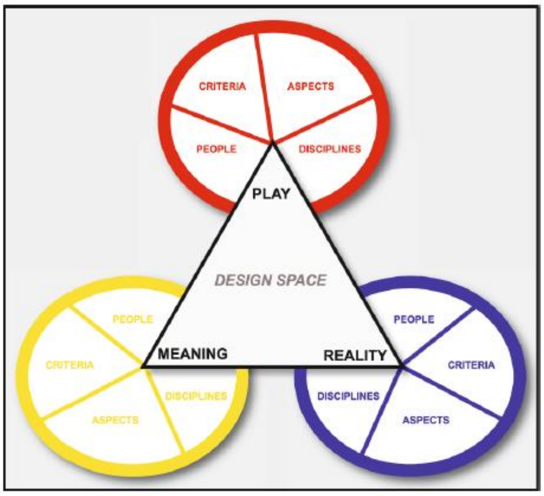


Figure 2: Triadic game design (Harteveld, 2011)

One of the fundamental serious games about SCM is called the 'beer distribution game' (Sterman, 1987). This game is a multi-player game, where participants take the role of managers in a simulated inventory management system. In this simulation, there are multiple actors, feedbacks, nonlinearities and time delays present. Analysis of this serious game shows that the participants fall victim to misperceptions of feedback. This results in a so-called 'bullwhip effect', where "variances of orders may be larger than that of sales, and the distortion tends to increase as one moves upstream" (Lee, Padmanabhan, & Whang, 1997). This game has been fundamental for showing the usefulness of serious games. However, several weaknesses exist in this game (Knolmayer, Schmidt, & Rihs, 2007):

- There is a inflexibility in the structure of the supply chain;
- The supply chain is introduced as a linear system, neglecting effects of decisions by companies that are not members of the supply chain;
- Capacity constraints are neglected;
- There are unrealistic assumptions, such as the delay in transferring information, which do not reflect the potential of today's powerful communication systems;
- There is a inflexibility in changing underlying parameters which hampers students' natural desire to play around with the model.

On the other hand, many serious games about construction management have been developed through the years. For example, Construct.it (Mayer, et al., 2013) is one of these serious games. Construct.it is a high-tech multiplayer planning game concerning the urban reconstruction of a seaport area in a real Dutch town.

Still, both of these serious games are not about supply chain management in construction specifically. The 'Beer Game' assumed the supply chain as a linear system, which was also fixed – meaning there are no differences in the supply chain between the different chains. However, due to one-of-a-kind products, the supply chain of every construction project is unique. While Construct.it does talk about (re)construction, it is not about one construction project but an entire area. Up until the latest years, there has not been a serious game about SCM in construction.

3.3. “Tower of Infinity”

In the latest years, a serious game regarding construction SCM was developed, dubbed “Tower of Infinity”. In this game, you play the role of a company that has gotten a contract for the design and construction of a big building project. It is a single-player board game where the player takes the role of the main contractor and needs to make sure that all processes in the design and construction processes run as smooth as possible. This game has already undergone testing by students of the course “Supply Chain Management & ICT” at the University of Twente and by a group of educational experts (van den Berg M. C., 2015).

In “Tower of Infinity”, you need to design a model, order the materials and assemble it on the construction site. When ordering bricks, there are multiple possibilities to order this brick, where the brick either costs more but is delivered quicker or the other way around. There is also the chance that delay occurs in a certain part of the production-sequence. A detailed analysis of “Tower of Infinity” can be found in Appendix B.

However, low-tech games – like board games – have their limits. Users of serious games might prefer high-tech (computer-based) versions over low-tech versions. This is because high-tech games can easily adapt to a needed situation, while still keeping the rules of the game in check and calculating the necessary parameters in only a second (Crawford, 1984). Nevertheless, knowledge in high-tech games about SCM in construction is not available yet. Additionally, not much research has been done on translating a low-tech game into a high-tech games. Knowledge about these two subjects is important when developing a high-tech serious game in this field. For this reason, a high-tech version of “Tower of Infinity” will be developed by researching the differences between high- and low-tech games. This way, the high-tech game will be translated as optimal as possible.

4. Results

Based on detailed analyses, the different sub-questions can be answered. The detailed analyses can be found in appendices, while the summarised conclusions are given here.

4.1. Characteristics of construction supply chain management

Before the goals of "Tower of Infinity" can be researched, it must first be analysed what kind of roles and characteristics SCM in construction has. After this is known, "Tower of Infinity" will be evaluated and different roles and characters of SCM in construction will be linked to different aspects of the game. Based on literature, the characteristics will be divided and explained in the following headings:

1. Roles
2. Comparison to the traditional approach
3. Framework
4. Problems

Roles

As mentioned in Chapter 3.1, supply chain management views the entire supply chain instead of only one group of manufacturers or suppliers. This way, SCM aims to boost the transparency and co-ordination of projects (Vrijhoef & Koskela, 2000). However, there are different ways that these aims are fulfilled. According to Vrijhoef & Koskela (2000), SCM in construction features four major roles, which are also visualised in Figure 3:

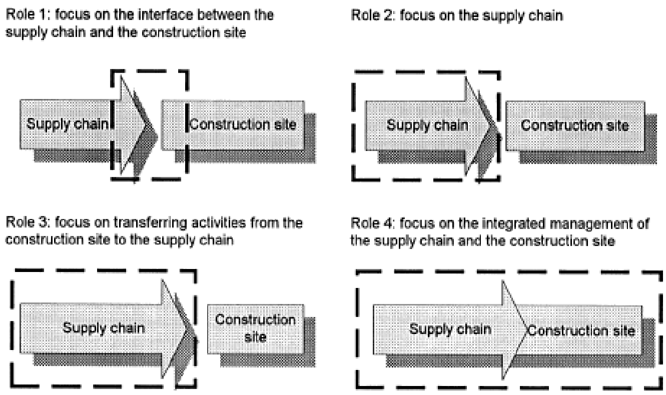


Figure 3: The four roles of supply chain management in construction

1. Firstly, the focus could lie on the impacts of the supply chain on site activities, as well as on the interface between these the supply chain and construction site. The aim of this role is to make sure that the translation from the supplier to the construction site goes as smoothly as possible, by means of dependable material and labour. These factors should result in a minimization of disruptions in the workflow, and so reducing costs and duration of on-site activities.
2. The focus might also be on the supply chain itself, where the goal is to reduce costs, especially those that are related to logistic, lead-time and inventory. In this role, the construction site is not taken into account for the most part.
3. Another role is to transfer activities from the construction site to the supply chain. This is useful, because the construction site has inferior conditions compared to the supply chain, next to many technical dependencies. The goal is again to reduce the total costs and duration of the project.
4. Lastly, the focus may be on the integrated management and improvement of the supply chain and the site production. This means that the construction site is also seen as part of the supply chain, and integrated into the supply chain management.

The difference between these roles is mostly how the construction site is taken into account. In role 1, the focus lies on the translation, which means that only the translation to the construction site is taken into account. In role 2 however, the construction site is mostly disregarded, because the focus is on getting through the supply chain as efficient as possible. Role 3 takes the construction site, and tries to transfer as much activities from the construction site to the supply chain. Role 4 merges the construction site completely into SCM.

It is mentioned that these roles should not be seen as mutually exclusive and are often used jointly (Vrijhoef & Koskela, 2000). This is important, because if only one role is taken into account, aspects from other roles might suffer significantly. For example, if only role 2 is taken into account, the supply chain will be the most cost-effective. However, the delays at the construction site will cause much bigger costs, because delays are not taken into account in role 2.

Apart from aforementioned roles, SCM in construction also has a number of characteristics that are very important to take into account. As mentioned in Chapter 3.1, Vrijhoef & Koskela (2010) characterized the construction supply chain as the following:

- A converging supply chain where all materials are directed at the construction site where the project is assembled from all incoming materials. In contrast to the manufacturing industry, the specific "construction factory" is set up around a single product instead of multiple products;
- It is in most cases only a temporary supply chain producing one-off construction projects. This means that every construction supply chain has a different structure, and so is characterized by instability, fragmentation and separation between the design and the actual construction of the object;
- It is a typical make-to-order supply chain: every project creates a different product or prototype. Apart from rare exceptions, there is little repetition in these supply chains. However, in particular cases the process can be very similar.

Comparison to traditional approach
The characteristics of SCM can also be defined by comparing it to the traditional approach. The significant differences between these two approaches are given in Table 1.

Table 1: Traditional and Supply Chain Management Approaches Compared (Cooper & Ellram, 1993)

Element	Traditional	Supply Chain
Inventory Management Approach	Independent efforts	Joint reduction in channel inventories
Total Cost Approach	Minimize firm costs	Channel-wide cost efficiencies
Time Horizon	Short term	Long term
Amount of Information Sharing and Monitoring	Limited to needs of current transaction	As required for planning and monitoring processes
Amount of Coordination of Multiple Levels in the Channel	Single contact for the transaction between channel pairs	Multiple contacts between levels in firms and levels of channel
Joint Planning	Transaction-base	On-going
Compatibility of Corporate Philosophies	Not relevant	Compatible at least for key relationships
Breadth of Supplier Base	Large to increase competition and spread risk	Small to increase coordination
Channel Leadership	Not needed	Needed for coordination focus
Amount of Sharing of Risks and Rewards	Each on its own	Risks and rewards shared over the long term
Speed of Operations, Information and Inventory Flows	"Warehouse" orientation (storage, safety stock) Interrupted by barriers to flows; Localized to channel pairs	"DC" orientation (inventory velocity) Interconnecting flows; JIT, Quick Response across the channel

Framework

Another way to explain the characteristics of supply chain management is by elaborating on the three primary structural aspects of a company's network structure (Lambert, Cooper, & Pagh, 1998):

- **The members of the supply chain**: The members of a supply chain include all companies and organizations with whom the focal company interacts directly or indirectly through its suppliers or customers. As a simplification measure, a distinction is made between primary members (companies that perform operational and/or managerial activities to produce a specific output) and supporting members (companies that only provide resources, knowledge, etc. for the primary units).
- **The structural dimensions of the network**: Three structural dimensions of the supply chain are essential when describing, analysing and managing the supply chain: the horizontal structure, the vertical structure and the horizontal position of the focal company.
  - The horizontal structure refers to how <u>long</u> the supply chain is. A supply chain can have only three tiers of suppliers, while another has five.
  - The vertical structure refers to how <u>broad</u> the supply chain is. The focal company may have only one supplier, but might as well have three different suppliers. While this only counts as one tier in the horizontal structure, the vertical structure is much more complicated.
  - The horizontal position of the focal company refers to <u>how much tiers</u> the supply chain still has to pass before and after the supply of the focal company. The focal company could be close to the initial source of supply, the ultimate customer or somewhere in-between.
- **Types of Business Process Links:** Because managing all process links is not efficient, priorities have to be made. Because of this, allocating scarce recourses in the supply chain is crucial. In supply chain management, four different types of business process links can be identified:
  - <u>Managed Process Links:</u> Links where the focal company integrates a process with one or more customers or suppliers;
  - <u>Monitored Process Links:</u> Links that are not as critical to the focal company as managed process links. Still, it is important to the focal company that these process links are integrated and managed appropriately between other member companies.
  - <u>Not-managed Process Links:</u> Links where the focal company is not actively involved, due to scarcity of recourses and/or little importance of the process link.
  - <u>Non-member Process Links:</u> Links between members of the focal company's supply chain and non-members of this supply chain. These non-members can be competitors, and are important to take into account because of possible shortage of a certain supply.

Problems

Management components are also an important element of the SCM framework. The fundamental components are given in Figure 4. While the physical and technical components are the most visible, tangible, measurable and easy-to-change, this should not be the only focus of managerial attention. Much literature on change management shows that this will cause the supply chain to most likely fail (Lambert, Cooper, & Pagh, 1998). The managerial and behavioural components are much harder to assess and alter, but define how the other components can be implemented.

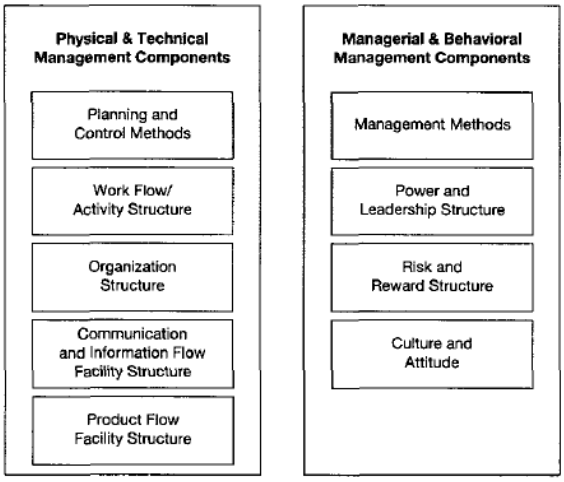


Figure 4: Fundamental components of supply chain management (Lambert, Cooper, & Pagh, 1998)

Many researches have mentioned the importance of business processes and relations in supply chain management. Working with the supplier as a partner in a long-term relationship would improve the quality and decrease the costs of production (Deming, 1982). Supply chains can create superior and unique performances by changing the structure of activities within and between companies. These activities must be integrated into key supply chain business processes, instead of managing only individual functions (Lambert, Cooper, & Pagh, 1998). The construction industry suffers from poor performance, and is lagging behind in terms of supply chain practices and efficiency (Vrijhoef & Koskela, 2000). The dominating focus on projects, fragmentation of the industry and separation of the design and production processes have been suggested as the root causes for this poor performance, next to lack of coordination and communication between participants, adversarial contractual relationships, lack of customer-supplier focus, price-based selection, and ineffective use of technology (Bankvall, Bygballe, Dubois, & Jahre, 2010). Most SCM models that are not developed for SCM in construction are problematic to use in the construction industry, because the pattern of interdependencies is very different in construction compared to other industries.

In conclusion, Table 2 is created that is branched into groups of characteristics of SCM in construction. This table is used later to easily analyse which characteristics have already been implemented in "Tower of Infinity".

4.2. Construction SCM problems that are present in "Tower of Infinity"

Before characteristics of construction supply chain management can be assigned to aspects of "Tower of Infinity", the game itself needs to be analysed. This analysis is given in Appendix B. After this, construction SCM can be compared to "Tower of Infinity". This will be done by taking the different SCM-headings from Chapter 4.1 and analysing which characteristics of these headings are and are not present in "Tower of Infinity".

Roles (Vrijhoef & Koskela, 2000)
The main role that "Tower of Infinity" focuses on is 'the interface between supply chain and construction site'. As the player, you need to make sure that the supply chain is integrated well with the construction site, so that materials don't have to be stored, and that there is always work to do on the construction site.

The second role – focusing on the supply chain – is present, but very simplified. The supply chain is visualised as a system, where every materials starts at a specific week (represents the preparation time of the material), and then flows linearly through this system, with a 1/6 chance of a delay when it's in its final 6 weeks of production. The chance for all materials are the same. All of these aspects are simplified, so while the supply chain is present, that is not the main focus of this game.

In the game, there is no possibility to transfer activities from the construction site to the supply chain. Every material has fixed supply rates and assembly rates. While there are different supply-rate choices for some materials, this is not the case for the assembly. Because of this, the third role is not present in "Tower of Infinity".

The last role of SCM in construction is to subsume the construction site into the supply chain. This role is another important aspect of "Tower of Infinity", since the game visualises the design, supply and construction as one entire chain of activities. The construction plant – which visualises the supply chain – should be adapted to the activities that happen on the construction site.

Comparison to traditional approach (Cooper & Ellram, 1993)
The characteristics that are the most important to "Tower of Infinity" are 'channel-wide costs efficiencies', 'long-term time horizon', 'on-going joint planning' and 'high level of coordination'. In the game, you try to spend as little money as possible through the entire project, and do not give any attention to where this is spent. The player also plans orders and assemblies so that these tasks are completed at the right time in the long-term. This is also an aspect of joint planning, where the supply and construction are integrated to acquire the best transition between these two chains.

The remaining aspects are also present in "Tower of Infinity" but are not influenced by the player, and so not as important as the abovementioned aspects. 'Joint reduction in channel inventories' and 'Small supplier base' are both framed by the product list. This list is a lot smaller and simpler than all the possibilities that are present in real-life. 'Sharing of information desired', 'High level of coordination required', 'Required channel leadership', 'Risks and rewards shared over the long term' and 'Interconnecting information flows' are all present due to the position that the player is placed in. The player overlooks the project from a top view, and so has all information of the project. The leadership is done singlehandedly by the player, and so coordination is relatively easy. The risks and rewards are observed globally at the end of the project, so that risks and rewards are shared.

Framework (Lambert, Cooper, & Pagh, 1998)
In "Tower of Infinity", there is no explicit division between primary and secondary members of the supply chain. Materials in the product list are not related to any supplier. The members that provide this product list could be seen as secondary members, since they only provide resources for the construction of the design. The focal company – which is played by the player – is the only primary member in this supply chain.

The structural dimensions have been significantly simplified in "Tower of Infinity". While the vertical structure is still fairly broad (there are a lot of different materials needed in the project), the horizontal structure only has a total of two tiers: all suppliers and the focal company. Because of this small structure, the position of the focal company is automatically at the end of the supply chain.

Social processes – like building a relationship with a certain supplier – are not implemented into the game. However, the different types of business processes can still be analysed. Still, the only process link that is used in "Tower of Infinity" is the managed process link; links where the focal company integrates a process with one or more customers suppliers. Because there is not more than one tier, every supplier is in direct contact with the focal company, and other links than managed process links are not possible.

In the management-aspect, the physical & technical components – planning & control, work flow & structure – have the priority in "Tower of Infinity". These are the aspects that the player has to optimise through the gameplay. However, there are also little aspects of the managerial behavioural components, for example, management methods and power & leadership structure. These aspects are fixed throughout the entire game; the power and leadership are completely done by the player.

Problems (Vrijhoef & Koskela, 2000) (Lambert, Cooper, & Pagh, 1998)
As mentioned before, social processes like, 'lack of customer-supplier focus' and 'lack of coordination and communication between participants' are not added in "Tower of Infinity". where Additionally, 'dominating focus on projects' and 'ineffective use of technology' are also not integrated into the game. The only social aspect that is present is from 'adversarial contractual relationships', where different players have to take LEGO-bricks from a shared construction-pile. There is no certainty a player can order a brick until this player actually orders the brick, which is equal to a short-contract in adversarial contractual relationships.

The remaining problems – 'fragmentation of the industry', 'separation of design and production process', and 'price-based selection' – are all aspects that are implemented into the game. All of these problems are also aspects that the player has to decide on. The player must try to blend design and production processes together, so that a smoothly running supply chain is created. . Selection between different materials should be a choice that is made while taking the current situation of the supply chain into account, instead of a price-based selection.

The results from this research and are also summarised in Table 2.

Table 2: Characteristics and implementation of construction-SCM

#	Construction-SCM characteristics	Implementation in low-tech game	Implementation in high-tech game
1.	Roles		
1.1.	Focusing on the interface between supply chain and construction site	Main focus	Main focus
1.2.	Focusing on the flow of the supply chain	Present, but simplified	Present, but simplified
1.3.	Transfer activities from the construction site to the supply chain	Absent	Absent
1.4.	Integrating the construction site in the supply chain	Main focus	Main focus
2.	Comparison to traditional approach		
2.1.	Joint reduction in channel inventories	Framed by supplier-list; no player-input	Framed by supplier-list; no player-input
2.2.	Channel-wide cost efficiencies	Goal of game	Goal of game
2.3.	Long-term time horizon	Goal of game	Goal of game
2.4.	Sharing of information for planning and monitoring	Assumed by position of player	Assumed by position of player
2.5.	High level of coordination required between horizontal and vertical levels	Goal of game, but simplified	Goal of game, but simplified
2.6.	On-going joint planning	Goal of game	Goal of game
2.7.	Small supplier base to increase coordination	Assumed by supplier-list; no player-input	Assumed by supplier-list; no player-input
2.8.	Required channel leadership for coordination focus	Assumed by position of player	Assumed by position of player
2.9.	Risks and rewards shared over the long term	Assumed by position of player	Assumed by position of player
2.10.	Interconnecting information flows, quick response across the channel	Assumed by supplier-list	Assumed by supplier-list
3.	Framework		
3.1.	Members of the supply chain		
3.1.1.	Primary	The player	The player
3.1.2.	Secondary	All suppliers	All suppliers
3.2.	Structural dimensions of the network		
3.2.1.	Horizontal structure	Two tiers	Two tiers
3.2.2.	Vertical structure	Multiple possibilities	Multiple possibilities
3.2.3.	Horizontal position of focal company	End of supply chain	End of supply chain
3.3.	Business processes		
3.3.1	Managed process links	Present	Present
3.3.2	Monitored process links	Absent	Absent
3.3.3.	Non-managed process links	Absent	Absent
3.3.4.	Non-member process links	Absent	Absent
3.4.	Management		

3.4.1.	Physical & Technical components	Priority of game	Priority of game
3.4.2.	Managerial & Behavioural components	Fixed by position of player	Fixed by position of player
4.	Problems		
4.1.	Dominating focus on projects	Absent	Absent
4.2.	Fragmentation of the industry	Priority of game	Priority of game
4.3.	Separation of design and production process	Priority of game	Priority of game
4.4.	Lack of coordination and communication between participants	Absent	Absent
4.5.	Adversarial contractual relationships	Only present due to competing players	Absent
4.6.	Lack of customer-supplier focus	Absent	Absent
4.7.	Price-based selection	Priority of game	Priority of game
4.8.	Ineffective use of technology	Absent	Absent

4.3. High-tech (dis)advantages of "Tower of Infinity"

Now that SCM in construction has been characterised and aspects that are already implemented into "Tower of Infinity" have been given, the advantages and disadvantages of high-tech (serious) games in regard to low-tech versions can be analysed. Literature regarding this comparison will be used to give a summary of a more general comparison. Then, a more specific comparison is done by using a play-test that was mentioned in the previous part. Throughout the gameplay, some observations were done, while the fellow players were asked what they thought of a potential high-tech version. At the end, a conclusion is given that summarizes the different advantages and disadvantages.

4.3.1. (Dis)advantages of traditional computer games

Before "Tower of Infinity" will be added to the context of this comparison, there are some general advantages that apply to most games (Crawford, 1984). One of the most significant differences between high- and low-tech games is that a high-tech game contains a high level of responsiveness, which arises from its plasticity. Digital games have the possibility to adapt to the wishes of the developer or the player. You cannot modify a board-game as easily as a computer-game. Another advantage of high-tech games is that it can function as a referee. Normally, other players must keep track that rules are not broken, but a computer can restrict what the player is able to do in certain phases of the game. Because a computer can perform complex equations at a much higher speed, this also gives the possibility to complicate the game rules.

The computer also provides additional possibilities due to its complex nature. Real-time play gives the possibility to let the player play the game, while the computer deals with all administrative matters. Furthermore, a computer also gives the opportunity to provide an intelligent opponent. Most games require an opponent, for example, the artificial intelligence of these computers are enough to provide a challenge for most non-rated chess players. However, his article was written in 1984, and it is safe to assume that the field of artificial intelligence has significantly grown in the last thirty years. Because the computer deals with all administrative matters, it can also decide to limit the information that is given back to the players in a purposeful way. This gives the possibility to make the game more realistic – as not everything is always known to the player – as well as more difficult. Limitation of information also encourages the player to use his or her imagination. Lastly, high-tech games give the possibility for large multi-player games. While Crawford mentions that these games will be played over telephone lines, computer-technology has developed so much over the last years, that these games are now played without use of telephone lines, but by use of the internet instead.

However, computers also have weaknesses. First of all, the limited input-output capability of most computers can cause problems. A computer can get through administrative matters quickly and is very responsive, but if the player cannot tell what it wants, all of these advantages are nullified. Graphics are the most essential output. In the article of Crawford, it is mentioned that good graphics are hard to come by, but the latest developments in game design have created very realistic games, or games with great aesthetic features. The computer games also have an advantage in the graphical aspect: animation and scenery provide a lot of possibilities that board-games cannot provide.

However, the largest restriction of computer games comes from the input. Input to the computer must come from the keyboard and controllers. This can make things very difficult for the game designer. How can key-presses accurately simulate reality? More than often, keyboard input does not represent real life actions very well. For example, walking around in real life does not feel the same as pressing the 'W'-button.

Another disadvantage is more social: computer-games are based on a single-user orientation. Two players are accepted, but more players on the same computer are largely discouraged. This results in computer-games giving off the feeling of being anti-social. Lastly, programming a computer-game is a tedious and much more difficult project than designing a board game is. Because a computer needs to programmed exactly to the wishes of the developer, this takes a lot of time, while board-games are created relatively easily, and the players itself play through the game, without a program helping them.

4.3.2. (Dis)advantages of modern high-tech games

In a relatively recently released study, Harteveld (2011) argues that the difference between digital and analog games is not clear-cut. Some analog games are supported by computers or other digital devices (a Monopoly game that uses a DVD for certain dice-rolls or events), or digital games are accompanied by real-life meanings, that cannot be imitated with digital devices. Harteveld mentions that the distinction lies in the intensity of the use of computer technology. The "intense" use of computer can make a difference (Harteveld, 2011):

i. Due to the option to automate procedures, digital games allow for more complex rules and more detailed functional worlds;
ii. When a digital game is played, a focus is initially placed on the aesthetics rather than the rules that manipulate these fictional elements;
iii. Because digital games are immaterial, they can depict worlds more easily than analog games.

The first difference is mentioned in the article of Crawford as 'additional possibilities due to its complex nature' in digital games. The third point is also mentioned in this article, under the definition of 'high level of responsiveness due to plasticity'. However, the second point is not mentioned. This is an important detail, because in serious games, the aim of the game should still be to learn the player something, and not to completely distract the player by use of aesthetics.

4.3.3. Performance of high-tech games

Research has been done to analyse whether there's a distinct difference between performances of high- and low-tech games (Ko, 2002). In this research, children between the age of seven and ten are assigned to play a simple game called 'Find the Flamingo'. In this game, a 5x5 card grid is given which has a flamingo hidden behind one of the cards. The player has to choose between one of the 25 possible cards. If the flamingo is not behind the chosen card, the back-side will show a hint if the flamingo is close to your chosen card. Every child played the game for 8 times without any interruption. The results of this research was that there was no significant difference between the performances of the children that played the high- and low-tech game.

4.3.4. Context of "Tower of Infinity"

When taking the aspects of "Tower of Infinity" into account, it can be seen that the different advantages and disadvantages are already present in this game. When the board game was played in a test-session, it became clear that the list of rules were too much to handle for the players that were new to the game. Even during the second round, it was unclear to them how some game-mechanisms worked, and what was and was not possible. A computer game will give the possibility to always request these explanations, so that the leader of the workshop does not have to walk around continuously. Additionally, at some point – when the player had to wait multiple weeks for a brick to arrive – the player forgot how many weeks had went by, and wasn't sure of the week that the player was currently in. These are all administrative matters that the computer can do instead of the player, which makes playing the game much more accessible. It was also apparent that the different players did not obey the rules to the fullest. When it became clear that the building of one player was not high enough, it placed a finished construction-brick from the lower floors to the top, without removing, re-ordering and re-assembling the brick. These kind of actions are also not possible with a high-tech version. In this case, the digital game will function as a referee. However, there are some advantages that do not apply for "Tower of Infinity". Firstly, artificial intelligence as an opponent is unnecessary, since the game is expected to be played in workshop sessions. Also, the game is a single-player game, so direct opponents are not required to play. The other advantage regarding multi-player games can also be disregarded because of the two abovementioned reasons.

The disadvantages of computer games are the current strong points of the low-tech version. Currently, the player grabs and places its LEGO-bricks in the corresponding areas. The actions of the player feel much more real, because he fulfils these actions themselves. When this changes into a high-tech game, everything will be done by pressing buttons and clicking the mouse. Because of this, the game will feel much less 'real' – since you are not placing the bricks yourself - and enjoyment might decrease. The anti-social aspect of this game is partly present. Since "Tower of Infinity" has been developed as a one-player game, there is no real restriction on that part. However, the pile of construction-bricks is shared by every participating in the workshop with the board-game, which is not possible when playing the digital game. This does degrade the social aspect of the game to a certain extent. When the participants of the play-test were asked about what they thought of a potential high-tech version of "Tower of Infinity", the reactions were dominatingly positive. The first reaction that was given is that it might give a lot of possibilities regarding choice-restrictions. As an example, a computer game of backgammon only allows a player to place its figures on the places that the dice allows them to. They also saw a possibility to make it more complex due to the adaptation to a high-tech game.

In conclusion, the advantages of a high-tech game is that it gives the possibility to focus more on the gameplay: the computer calculates every step of the game, the player only has to click to perform actions. At the same time, only clicking decreases the fun that the player has while playing the game, and it also feels less 'real'. The superior aesthetics of a digital game might also distract the player from the actual aim of the game. Still, whenever the player feels like he is stuck in the game, the digital version has the possibility to explain it to the player without the help of any other person. While the functional aspects are improved and possibilities regarding complexity open up when adapting "Tower of Infinity" to a high-tech game, the social aspects might be degraded. These social aspects must be identified, so changes can be made during the development of the high-tech game. The way that this will be done, is discussed in the next part.

4.4. Development of "Tower of Infinity"

Now that the analyses of have been completed in Chapter 4.1 to 4.3, the game can be developed. To make sure that the development of the game goes as smoothly as possible, certain design approaches will be applied. During the development of the low-tech version, Triadic Game Design (TGD) has been used as the main design approach (van den Berg, Voordijk, Adriaanse, & Hartmann, 2017). This will be further used in the transition to the high-tech game. Since the process of programming is a large project, another additional design-approach has to be used.

4.4.1. Triadic Game Design

Firstly, Triadic Game Design is a game design approach that has been created by Harteveld (2011). This approach indicates what serious games should consist of: the worlds of Reality, Meaning and Play. These worlds are all inhabited by different people, disciplines, aspects and criteria. It is argued that if these three worlds are not balanced well, the game will collapse. The different worlds mean the following:

World of Reality

The first world is the world of Reality. This world is the link between the game and the real world. Non-serious games rarely represent the real world perfectly, and more than often a completely other world is created. In serious games however, the world of Reality is essential to the effectiveness of the serious game. This connection between game and reality suggests that games contain an underlying model of reality (Harteveld, 2011), and so are able to apply the knowledge from the game into real life.

To create such a serious game, the recent developments and characteristics in SCM in construction have been analysed by van den Berg (2017). Firstly, a great part – around 75% - of the product's value is built with materials and services from subcontractors and suppliers. In the construction industry, these supply chains are also very short-lived due to its "make-to-order" nature (production only starts after an order is received). The chain should also remain flexible to match the needs during the project execution. These supply chain also feature dynamics, uncertainty and partial information sharing. Construction supply chains are typically converging (where materials are being clustered and transported to a single construction site). When talking about recent developments, integrated contracts like Design & Construct contracts (where the main contractor is responsible for both the design and construction of the building. "Stable, predictable production, multi-skilled crews that can perform multiple construction-related tasks are being recognized as a key to achieve stable, predictable production". Another tool that is being used more frequently is Building Information Modelling (BIM). The construction industry also realises that virgin resources are limited.

The world of Reality has been implemented into "Tower of Infinity" in the following ways (van den Berg, Voordijk, Adriaanse, & Hartmann, 2017):

- The player needs to purchase building materials from suppliers by choosing a product from a supplier's product guide (containing options with different costs and lead-times for the same product);
- An empty construction site of limited size is created to which purchased materials are transported after their manufacturing is completed;
- Materials are only being manufactured after the player places an order;
- The player is placed control of both the design and construction project stages and by offering four crews that can all perform several kinds of actions;
- A three-dimensional prototype needs to be 'modelled before any materials can be ordered, manufactured and eventually assembled;
- The availability of (raw) materials used for manufacturing is limited.

World of Meaning

The second world is the world of meaning. In this world, the focus lies on what the player 'learns' from the game, and what can be taken "home" (Harteveld, 2011). A game always has a meaning, but it is the choice of the player to decide to either take this meaning and apply it to real life or not. For a serious game, it is essential that the world of reality and meaning are intertwined. Else, the world of reality doesn't apply for what the player learns, and might learn something that is (partly) wrong.

The goal of "Tower of Infinity" is to acquire an intuitive understanding about optimizing project activities regardless of functional or corporate boundaries. Overall, the following three learning objectives are specified (van den Berg, Voordijk, Adriaanse, & Hartmann, 2017):

1. Coordinating design and construction tasks coherently;
2. Taking constructability aspects into account when designing;
3. Continuously balancing scope, time and cost.

Next to the players, observers and researchers also have the possibility to collect data or test a theory using this serious game. However, to be able to test a theory or collect data, the game must keep track of certain variables in the game, and these variables should represent something within its context. In the current board-game, this kind of information is kept track of by using pencils with different colours, that each individual player has to write down themselves (van den Berg, Voordijk, Adriaanse, & Hartmann, 2017).

World of Play

The last world is the world of play. Since the definition of a 'game' is different for each individual, the difference lies in the play-aspect. To develop a game, designers need to come up with a game concept which is a detailed idea of what the game is like (Harteveld, 2011). In the book of Harteveld, six different genres are given: action, adventure, puzzle, role-playing, simulation, strategy and virtual world games.

"Tower of Infinity" has been developed as a simulation game (van den Berg, Voordijk, Adriaanse, & Hartmann, 2017). According to Harteveld (2011), such games are characterised by a closer connection to reality (probably because it is a serious game), the lack of an extensive story as the player progresses and many degrees of freedom and openness. The basic game has been elaborated on by defining the four basic elements that are part of every game: its mechanics (you can only perform a specific amount of tasks per week/round), story (you are the main contractor of a Design & Construct contract), aesthetics (basic layout of the game) and technology (board game and LEGO-bricks). An extensive analysis of the game has already been given in Appendix B.

Conclusion

Because the board game has been developed using triadic game design, the transition to a digital game should be closely monitored using this approach. As mentioned in Chapter 4.3, there are essential differences between low- and high-tech games. Some things also cannot be integrated into the high-tech game (the global construction pile of LEGO-bricks). When changing aesthetics or other aspects, the changes in the three worlds should be kept into account. The focus must not lie on the aesthetics, but it should still be fun to play. To make sure that the optimal balance is once again found in the high-tech version, the game will be tested multiple times before the workshop with an organization.

4.4.2. Agile Software Development

Apart from the Triadic Game Design, the actual programming is a freestanding design-process. Since there should be possibilities for others to work further on this project, it's important that the design is very open and agile. For this reason, Agile Software Development (ASD) has been recommended to use. "ASD is a methodology for the creative process that anticipates the need for flexibility and applies a level of pragmatism into the delivery of the finished project. ASD focuses on keeping code simple, testing often, and delivering functional bits of the application as soon as they're ready. The goal is to build upon small client-approved parts as the project progresses, as opposed to delivering one large application at the end of the project" (TechTarget, 2007).

To keep the code as simple as possible, different files will be created in which different actions are done (the selecting of bricks, going to the next week and doing delay-tests). Each of these files will be created when they are needed. This means that during the process, each of these files will be created when they are needed, and updated accordingly. At the start, the code starts very simple. The specific steps that are taken will be elaborated on in the next part.

4.4.3. First version of "Tower of Infinity"

Using ASD, multiple versions of Tower of Infinity have been saved to always have a backup available. The development of these versions are explained in Appendix C. In this part, the game will be analysed on how certain aspects from the board-game have been designed in the high-tech version, as well as the reasoning behind it.

The most influential factor to take into account is that the learning-goals from the board-game are not decreased significantly. This has been taken into account by implementing the factors that have been used to test the efficiency of the board-game (van den Berg, Voordijk, Adriaanse, & Hartmann, 2017):

The player experientially learns improving a construction supply chain through:

1. Coordinating design and construction tasks coherently:
    a) Uses systems perspective to focus on the entire supply chain
    b) Tries to achieve a lean process and/or Just-In-Time deliveries
2. Taking constructability aspects into account when designing:
    a) Recognized construction sequences
    b) Adapts strategy based on product lead-times and assembling rates
    c) Bases design on availability of materials and construction site characteristics
3. Continuously balancing scope, time and cost
    a) Makes systematic trade-offs to fulfil client requirements
    b) Balances time and cost when ordering construction materials
    c) Makes trade-offs in response to manufacturing delays

These aspects have been tried to implement as efficient as possible. However, one main aspect from the board-game could not be directly translated into the computer-game. Because the board-game is always played in a workshop-situation, there was a certain multiplayer aspect present. A certain brick that was needed by all participants was deliberately not represented enough in the pile of construction bricks. On the other hand, the computer is solely a single-player game, where the gameplay is not affected by other participants. It has been decided that for now, all construction materials are available at all time, but this could be changed in the following versions, so that all characteristics from the board-game will also be present in the computer game.

Information that was acquired before developing "Tower of Infinity" has also been used to design the game. Firstly, during the design of the computer-game, the feedback that was acquired during the play-test mentioned in Chapter 4.3.4 is applied. Certain characteristics of the board-game in regard to a computer-game could be analysed during game-play of participants, and other (dis)advantages were discussed after the game was finished by all participants. The main problem of the board-game was that there were many rules that needed to be taken into account, and so rules were – deliberately or accidently – broken fairly quickly. This was also the direct response when the participants were told a computer-version was being developed. The participants explained how in a computer game, the game restricts you from performing actions and so learns you the rules while still playing through the game. In the computer-game, you cannot go to menus that are not accessible yet or perform actions that are not allowed according to the rules. In addition, the player will be told when it tries to break a rule (for example, placing an unfinished design-brick on top of another unfinished design-brick). An example of these messages is given in Figure 5.

You cannot delete model-bricks if there are no bricks that are (being) modelled yet.

Figure 5: Example of a message that the player receives when breaking a rule

Another related problem was that all rules are too much to remember at the start of the workshop. A solution for this was that all rules and controls can be found by pressing either the help- or rules-button. When the player is in the main menu, screens get called that show all information. The player cannot call the rules-screen when in another menu. When the help-button is pressed while an action is being performed, it gives custom information about that action. This way, the player can always get feedback on what needs to be done. The help- and rules-screen can be found in Figure 6, and an example of a help-message is given in Figure 7.

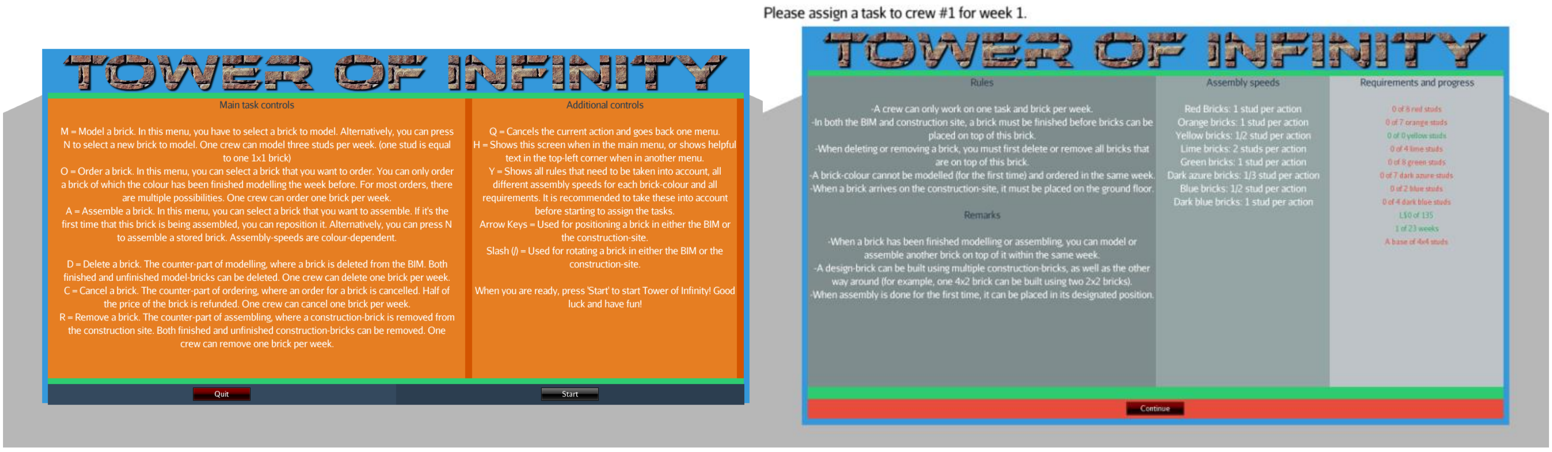


Figure 6: The screen containing all controls (left) and the screen containing the rules, assembly-speeds and requirements (right)

You are currently in the task menu of modelling. You have to select the brick that you want to apply this task to.
In the modelling- and assembly-phase, you can also press N to model a new brick or assemble a storage brick.
If you want to cancel the current task, press Q.

Figure 7: Example of a message that the player receives when asking for help

Additionally, literature research has also given some differences between board and computer-games. As mentioned in Chapter 4.3, computers can do all calculations and keep track of all administrative work. In the board-game of "Tower of Infinity", each participant had to do that for themselves. In the computer game, players get information about the progress of a certain brick whenever they perform an action on it. The planning that is present in the board game – and needs to be edited by each individual player in the board game – is automatically edited by the computer during the gameplay. Bricks are also semi-transparent when they are unfinished, and will become opaque when they are finished. This is also related to the advanced visuals that computer-games bring with them. The difference between unfinished and finished bricks can be found in Figure 8 and a filled-in planning is given in Figure 9.

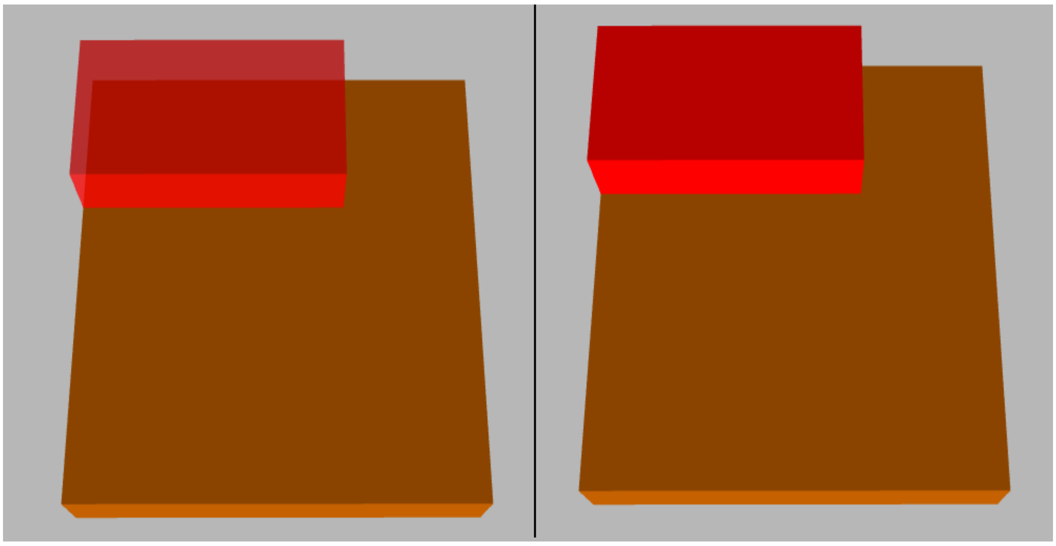


Figure 8: Both an unfinished brick (left) and a finished brick (right)

Week:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Crew 1:	M	M	O				A	A	A	A	A	A	A																									
Crew 2:	M	M	O				A	A	A	A	A	A	A																									
Crew 3:	M	O					A	A	A	A	A	A	A																									
Crew 4:	M	M					A	A	A	A	A	A	A																									

Figure 7: The planning that is shown in the main menu, where letter- and colour-combinations show what actions has been done at what time

The disadvantages of computer-games, where actions – like placing a brick – feel less real, or that aesthetics distract from the real purpose of the game, have also been taken into account. While the computer-game will still not feel as real as using the LEGO-bricks in real life, the use of arrow-keys instead of the mouse requires more action from the player. The focus of the game-design is also not on the aesthetics, so distraction because of this will be minimal.

The game has been developed using all information that was acquired in preparation for this research. However, some things might have been overlooked that isn't by others. For this reason, feedback sessions have been performed with both the developer of the board game as well as people with fewer knowledge about the game.

4.4.4. Feedback

The focus on these feedback-sessions was on the usability of the game: the game should be playable with little to none assistance, and the controls should feel intuitive. The reason for this focus was that if the game was only able to be played with a lot of assistance, it was not possible to test the game as a serious game altogether. All individual feedback-sessions are elaborated on in Appendix D.

Overall, there were a lot of comments regarding the start of the game, where the explanation was given of the entire game. Too much text was present (as seen in Figure 6), while all aspects of the game – like the delay only happening in the last 6 weeks – were still not explained. Additionally, the different places that represent the BIM and construction site are not given a name in the visualisation. The controls did also not feel intuitive, due to the switching between keyboard and mouse. Furthermore, players did not get notified enough by the game when/if actions were completed. Lastly, information about requirements, assembly-speeds and delay is not shown at useful points of the game, or not at all. All of these points of improvement are part of the world of play, since this contains all rules and goals of the game (van den Berg, Voordijk, Adriaanse, & Hartmann, 2017). If the rules and controls are not explained, the world of play is not present, and the game will collapse (Harteveld, 2011).

However, if these points of feedback will be applied to the next version, the game should explain itself much better, and the world of play should be revitalised. Firstly, the text at the start will be replaced by multiple screens explaining the different processes the player has to go through, instead of only explaining the controls. In this tutorial, the requirements will also be given. These screens will be accompanied by edited screenshots of the game, to visualise what the player has to do in the game itself. Secondly, text and images will be added in-game, to add names to all platforms, as well as indicators at the last 6 weeks of the production plant to show that delay can happen there. These indicators also show in which week delay occurs. Thirdly, the choice-screen will also contain images of the requirements and assembly-speeds. Fourthly, sounds will be added to inform the player better when actions are fulfilled. Lastly, buttons will be added throughout the game, to give the possibility to click on-screen instead of pressing keys on the keyboard.

4.4.5. Final version of "Tower of Infinity"

In the final version, the points of improvement from Chapter 4.4.4 has been applied. The development of the game-board can be seen in Figure 10. More screenshots comparing the first and iterated version of "Tower of Infinity" can be found in Appendix E. The elaboration of all Java-files and their methods is given in Appendix F. All aspects that are present in this final version is given in Table 2.

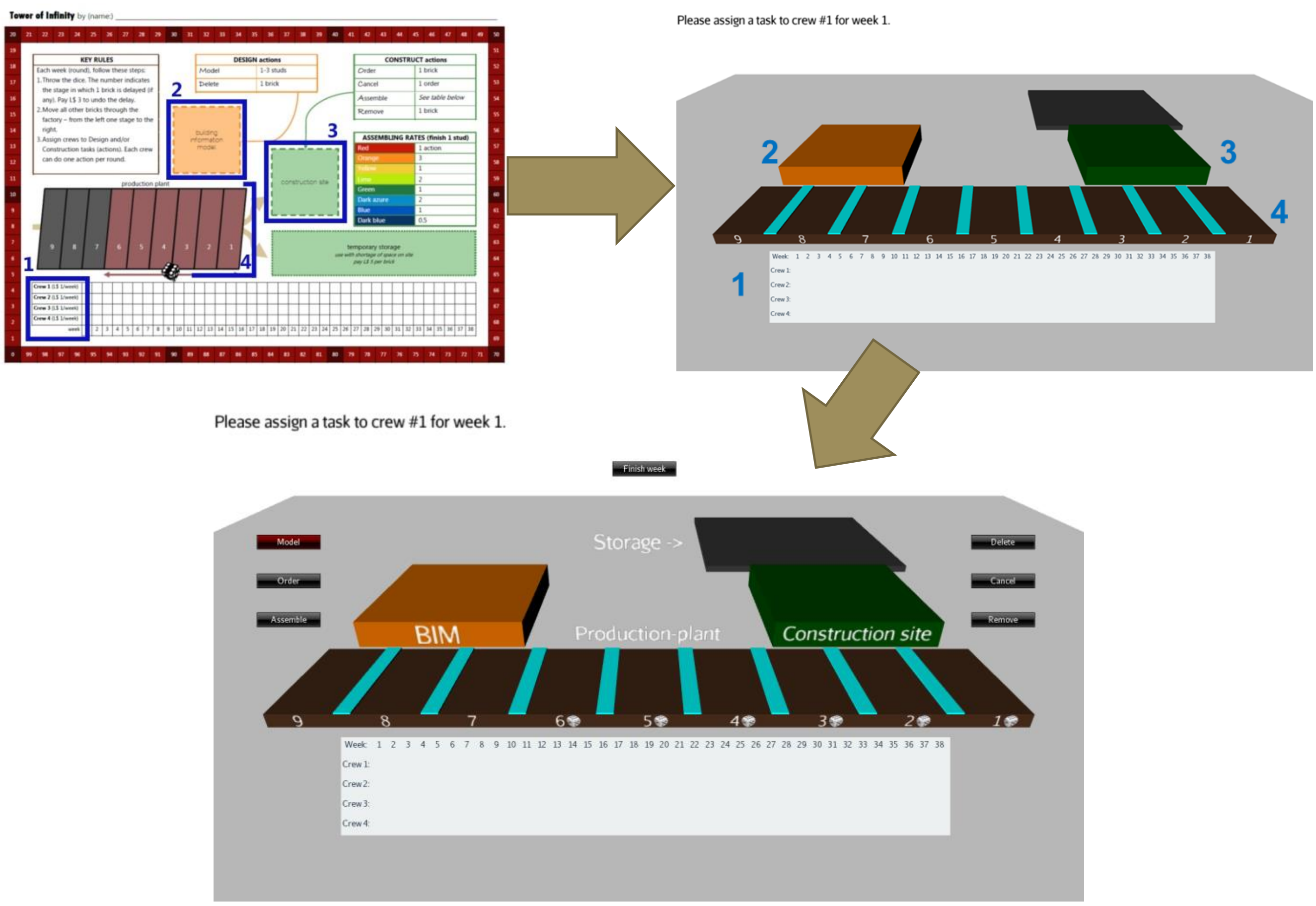


Figure 8: Transition from low-tech game to first high-tech version to final version

4.5. Creating awareness with "Tower of Infinity"

Because there was no time to plan a workshop after applying the feedback from the first version, the game has been presented in a knowledge-table-meeting, since the game could not be played using the version that was available at that time. A knowledge-table is a meeting of multiple experts in a certain field, to discuss about matters in that field and to share knowledge. This knowledge-table consisted of around twenty experts and was mainly about the fact that the focal company does not have any insight into the chain-costs of the subcontractor. There is also distrust from the sub-contractors, and due to shortage of time, companies just fall into the safe and trusted mode of the least risk and outsourcing everything to sub-contractors.

In the presentation, the game was explained and played through briefly, after which different statements were shown to start a discussion on the problems of SCM in construction. These statements were accompanied by screenshots from the game, to show how the problems of the statement were applied in the game. In total, four statements were in the presentation, but due to time-restriction, there was only the time to discuss two of these statements:

- <u>Without transparency of the suppliers, a successful project will never occur.</u>

While it took a while before the first response came, the first reactions were all that this statement is an obvious 'yes'. However, the problem lies in the reasoning behind the suppliers that do not want to be transparent, and how this can be fixed. This resulted in a long discussion mostly about while utility buildings are mostly unique, the same suppliers are still chosen every project. This means that some form of a contract or trust could be acquired, while this has not been done yet in most cases.

- <u>A contractor should take responsibility to help resolve delivery issues with a subcontractor or supplier.</u>

At first, the majority didn't agree with this statement. Suppliers should fix their own problems, and a contractor has nothing to do with it, since they would be solving the problems of someone else. After discussing the matter however, some came with the solution to act before a problem occurs instead of reacting when a problem occurs (active instead of reactive). Additionally, an example was taken from the manufacturing industry, where manufacturing-companies actually did help their suppliers when delivery issues appeared, and resulted in long-term profit.

After the presentation was finished, all attendees were asked to fill in a questionnaire which asked them questions about the world of play, meaning and reality. In Table 3, all keywords that have been used to define the game in each of these worlds are given (n = 10). Because the game has not been played by the attendees, it was sometimes hard for the attendees to give detailed feedback on the meaning and realism of the game. However, it can be seen from the results that – when not playing the game themselves – the attendees thought that the world of play was well-balanced. The reason that the attendees are positive about the world of play, while the participants of feedback-sessions were not, is that the attendees did not play the game themselves. The largest problem with the first version is that the controls are not explained well-enough. During the presentation, the attendees were only shown the different rules and goals of the game.

Overall, the response shows that it's also possible to use the high-tech serious game as a discussion-starter: a scenario is defined in the game, and a certain problem arises. A statement is made that is related to the problem, which starts a discussion and potentially a brainstorm-session. This would be much harder with a low-tech game, where every scenario needs to be either shown on a table or photographed individually.

Table 3: Results questionnaire

Play		Meaning		Reality	
Keyword	**Amounts used**	**Keyword**	**Amounts used**	**Keyword**	**Amounts used**
Effective/fills void	5	Creates awareness	3	Challenges match	2
Interesting	2	Too little knowledge in the game	3	Too little knowledge in game	2
Good setting	2	Added value after more development	2	Only realistic after development	1
Lacks graphics	1	Not complex enough	1	Too much trust assumed	1
Too little knowledge in game	1	Useful	1	Relevant for purchasing processes	1
Clear	1	Inspirational	1	Building projects cannot be compared	1
Fun	1	N/A	1	Single-player is not realistic	1
Simple	1			Attendee possessed too little knowledge	1

5. Discussion

This research does not only have the goal of designing a high-tech serious game, but there are other contributions that this research aims to provide. There is also still room for improvement in the game itself. Lastly, there are certain limits and boundaries that this research had to take into account. All of this information is given here.

5.1. Contributions

This research mainly hopes to let users of "Tower of Infinity" gain more insight into SCM in construction. This game will try to make its users more aware of certain bottlenecks that are present in construction SCM. Game designers may also find necessary information in this research about how to translate a low-tech game into a high-tech version.

In addition, the aim of this research is also that the high-tech version of "Tower of Infinity" creates an open game-framework, so that other researchers have the possibility to work further on this project, and use their own insight to further develop the game. This way, it could become one of the fundamental tools for gaining insight in supply chain management in construction.

Ultimately, this research and the developed high-tech version of "Tower of Infinity" should lead to an increase in productivity in the field of supply chain management in construction.

5.2. Further development of "Tower of Infinity"

The final version of this research is not the final version of "Tower of Infinity". After this research, the end-product will be used to develop "Tower of Infinity" further. For this reason, an analysis is performed of some of the aspects that are still missing in the high-tech game. These aspects can also be found in Table 2.

Firstly, there are too little horizontal tiers in the supply chain. Most supply chains have multiple horizontal tiers, which means that the suppliers of the focal company are dependent to other suppliers (Lambert, Cooper, & Pagh, 1998). In the current game, there is only one tier underneath the focal company.

Secondly, according to experts in the field of SCM in construction, too much trust is assumed in the game, and too much information is given to the player. In reality, no company possesses all cost-information about all suppliers. A possible solution is that production times could be made variable and/or vague, where instead of a delivery time of three weeks, it could be either three or four weeks (and the player wouldn't know until the brick arrives).

Thirdly, one of the points of improvement from the low-tech game was that the current version contained the same risk probabilities for all bricks, which is unrealistic (van den Berg, Voordijk, Adriaanse, & Hartmann, 2017). An improvement is to set a higher chance of delay for higher bricks, or to create more purchase options where costs, time and delay-chance vary.

Fourthly, as mentioned in the feedback sessions, multiple levels are needed to learn the player solve different problems in construction SCM. This suggestion was also made by one of the attendees of the presentation, who said that different projects could not be compared in the game. With different levels, different aspects of construction SCM can have the priority in each of these levels.

Lastly, social aspects are not implemented in the game. There are no possibilities to buy the same materials from the same supplier and gain a benefit from this. ´Trust´ can also not be built between suppliers. Working with the supplier as a partner in a long-term relationship would improve the quality and decrease the costs of production in reality (Deming, 1982), so this could add to the realism of the game.

5.3. Limits and boundaries

In this project, there are a number of limits and boundaries that need to be taken into account before it can be understood what this thesis plans to contribute to the field of supply chain management in construction.

Firstly, one of the boundaries that applies is that this thesis is about supply chain management in construction only. It does not take any other field into account, nor will the game be designed to gain insight in any other field other than SCM in construction. Secondly, the high-tech game that will be designed will be based on the low-tech version of "Tower of Infinity". There is little freedom in designing the base of the game, since this will be based on the low-tech version. However, it can be argued that this boundary is needed, due to the 10-week period that this thesis will be written in. If this freedom for design was present, it would take too long to create the base-game.

The most important limit is that the game will be incredibly simplified. Uncertainty is decided by a fixed value, while different forms and colours of bricks possess different properties. In real-life, uncertainty is far from that simple, and there are not as little possibilities for materials as the game suggests. Because of this, insight that is gained through "Tower of Infinity" cannot be applied to the real world directly. The insight is relevant, but the serious game does not open a window to directly gain knowledge in real-life situations. Users of "Tower of Infinity" need to gain insight in what bottlenecks are visualised with these simplifications, but have to come up with solutions for the real-life problems themselves, possibly with the help of the serious game.

6. Conclusion

In this research, characteristics of supply chain management in construction have been analysed, and placed in the context of the low-tech serious game "Tower of Infinity". After this, (dis)advantages of high-tech games in regard to low-tech games have been analysed. After all these analyses were finished, the first high-tech version of "Tower of Infinity" has been developed. This version has been tested with multiple participants that had a broad range of knowledge about supply chain management in construction. Out of these sessions, it became clear that the world of Play was not integrated into the high-tech game well. This feedback was applied to create a final version where the player is guided through the game much better: sounds and buttons give a much more intuitive game. The first version has been used in a presentation during a knowledge table. According to the questionnaires that were handed after the presentation, the response was mostly positive.

In conclusion, users of the high-tech version of "Tower of Infinity" now have more possibilities to become aware of problems in the field of supply chain management in construction. Apart from being used solely as a game, it can also be used in a presentation to create scenarios as a discussion-starter. While the efficiency of the final high-tech version of "Tower of Infinity" has not been tested, the results from the game-analysis (given in Table 2) and the presentation (given in Table 3) show that the high-tech game points to the same problems as the low-tech version, and can still be used after usability has been improved. This improvement has been done in the final version.

7. Acknowledgements

This research was carried out at the University of Twente, Enschede. It was part of a bachelor thesis for the study Civil Engineering.

Firstly, I would like to thank my supervisors Marc van den Berg and Hans Voordijk, for introducing me not only to this bachelor thesis, but also to the field of construction supply chain management and serious gaming. They also supported me through the process of learning about both of these fields, and were always ready to help whenever I got stuck in any part of my research. I would also like to thank every researcher of the office Z200-Z206 of the University of Twente. The serious but cosy atmosphere that was always present made the working space a lot more livelier.

Secondly, I would like to thank everyone that participated in an individual feedback-session or was present at the knowledge-table. If they would not have given me feedback, this research wouldn't have developed as far as it has now. I would specifically like to thank the knowledge-table for allowing me to give a presentation in such an environment.

Lastly, I would like to thank my parents. Apart from supporting me and giving me feedback where they could (as always), they also pulled me through the last two weeks of the thesis, where I got sick in both weekends. Having the possibility to vent my frustrations at home whenever I got stuck in my development made my entire thesis much more enjoyable.

Apart from learning how to write a research, I've also learned a lot from the fields of both supply chain management and serious gaming. Additionally, through the entirety of this project, I've learned how to program with Java and jMonkey, which has been a valuable experience. The knowledge-table also gave me the a look into the real world of supply chain management, and attending these discussions were very informative for the research, but also for my personal development. In conclusion, I've learned a lot of things in a fairly short time, for which I'm very thankful for.

8. Bibliography

Bankvall, L., Bygballe, L. E., Dubois, A., & Jahre, M. (2010). Interdependence in supply chains and projects in construction. In *Supply Chain Management: An International Journal* (pp. 385-393).

Cooper, M. C., & Ellram, L. M. (1993). Characteristics of Supply Chain Management and the Implications for Purchasing and Logistics Strategy. In *The international journal of logistics management* (pp. 13-24).

Crawford, C. (1984). *The Art of Computer Design.* Washington State University.

Deming, W. E. (1982). *Out of the Crisis.* Cambridge: Massachusetts Institute of Technology.

Harteveld, C. (2011). *Triadic Game Design: Balancing Reality, Meaning and Play.* Delft: Springer.

jMonkeyEngine. (2017). Retrieved April 19, 2017, from jMonkeyEngine: http://jmonkeyengine.org/

Knolmayer, G., Schmidt, R., & Rihs, S. (2007). *Teaching Supply Chain Management Beyond the Beer Game.* Bern: University of Bern.

Ko, S. (2002). An Empirical Analysis of Children's Thinking and Learning in a Computer Game Context. In *Educational Psychology* (pp. 219-233). Routledge.

Lambert, D. M., Cooper, M. C., & Pagh, J. D. (1998). Supply Chain Management: Implementation Issues and Research Opportinuties. In *The International Journal of Logistics Management* (pp. 1-20).

Lee, H. L., Padmanabhan, V., & Whang, S. (1997). *Information Distortion in a Supply Chain: The Bullwhip Effect.* Stanford: Graduate School of Business.

Mayer, I., Bekebrede, G., Harteveld, C., Warmelink, H., Zhou, Q., van Ruijven, T., et al. (2013). *The research and evaluation of serious games: Toward a comprehensive methodology.* British Journal of Educational Technology.

Navarro, A., Pradilla, J. V., & Rios, O. (2012). *Open Source 3D Game Engines for Serious Games Modeling.* Colombia: Universidad Icesi.

Segerstedt, A., & Olofsson, T. (2010). *Supply chains in the construction industry.* Narvik: Emerald.

Sterman, J. D. (1987). *Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decisionmaking Experiment.* Cambridge: Massachusetts Institute of Technology.

Susi, T., Johannesson, M., & Backlund, P. (2007). *Serious Games - An Overview.* Skövde: University of Skövde.

TechTarget. (2007, Februari). *What is agile software development (ASD)?* Retrieved Mei 16, 2017, from SearchSoftwareQuality: http://searchsoftwarequality.techtarget.com/definition/agile-software-development

van den Berg, M. C. (2015, October 21). *Introducing to Tower of Infinity - a construction supply chain management game.* Retrieved March 7, 2017, from BUILDING SCIENCE STORIES: https://www.mcvandenberg.com/blog/introducing-the-tower-of-infinity-a-construction-supply-chain-management-game/

van den Berg, M., Voordijk, H., Adriaanse, A., & Hartmann, T. (2017). *Experiencing supply chain optimizations: a serious gaming approach.* Enschede.

Vrijhoef, R., & Koskela, L. (2000). *The four roles of supply chain management in construction.* Delft: Elsevier Science Ltd.

Appendix A: Terminology

In this paper, discipline-specific vocabulary is used. For the sake of clarity, repeatedly used or obscure terms are therefore defined here.

Design space	An imaginary place from where the design is created from. The design space is mostly determined by the applied design approach and philosophy (Harteveld, 2011).
Game engine	Toolkits aimed to ease the development of videogames, acting as a superstructure of several development efforts. They are also normally packed with a set of tools to be used in the design and coding stages (Navarro, Pradilla, & Rios, 2012).
High-tech game	A game that makes use of an electronic game-device (i.e. consoles or computer).
Knowledge-table	A meeting that consists of multiple experts in a certain field to produce and share knowledge.
Low-tech game	A game that does not make use of any electronic game-device.
Serious game	Games used for purposes other than mere entertainment (Susi, Johannesson, & Backlund, 2007).
Supply Chain	The network of organisations that are involved, through upstream and downstream linkages, in the different processes and activities that produce value in the form of products and services in the hands of the ultimate customers (Vrijhoef & Koskela, 2000).
"Tower of Infinity"	A serious game developed by M.C. van den Berg, that this thesis will be based on.
Triadic Game Design (TGD)	A game design used for serious games that involves a triad that consists of the worlds of reality, meaning and play (Harteveld, 2011).

Appendix B: Characteristics of "Tower of Infinity"

Here, aspects from the game will be analysed to link these aspects to SCM in construction. The game has been played in a group to understand the structure of the game and the characteristics of the gameplay. Firstly, a short summary is given of "Tower of Infinity", after which the procedure of the game is given. Then, analysis is done about what characteristics the game in- and excludes, and how these are connected to SCM in construction.

"Tower of Infinity" is a one-player board game, specifically developed for supply chain management in construction. The player takes on the role of the main constructor, and is given the responsibility for designing and constructing a multifunctional skyscraper. This skyscraper comes with requirements regarding materials, time and costs. The player has the mission to try to satisfy all these requirements by assigning different activities to four different crews through every week. Of course, in such a project, there are many uncertainties, so the player must keep these uncertainties in check, while still trying to keep the supply chain flowing.

B.1. Procedure of "Tower of Infinity"

The explanation of the game will be divided in three parts: pre-game, gameplay and post-game. This is done because some aspects do not become clear if the pre- and post-game are not discussed. Personal experiences with "Tower of Infinity" will not be included here, but in the analysis of the characteristics.

Pre-game

"Tower of Infinity" is a one-player game, but mostly played by multiple players at the same time. Players each have their own playing board, product list and coloured pencils.

The game board is given in Figure 11, where:

1. The team-planning: Here, the different tasks of the design-and-construct team can be filled in;
2. The design space: Before anything else, the tower has to be designed first. This part represents the Building Information Model, where the different crews work on designing the model.
3. The construction site: This space is where the assembly of the final construction is done. This represents the building site, where the different parts of the construction are assembled together.
4. The construction plant: This part is where the bricks are 'made' before being transported to the construction site. This represents the supply chain.

Another paper that is needed to play "Tower of Infinity" is the product list. This paper shows all costs and produce-times for all different colours and forms of bricks. The same brick and color combination can also have different offers, where higher costs give a lower production time. This represents different companies and materials.

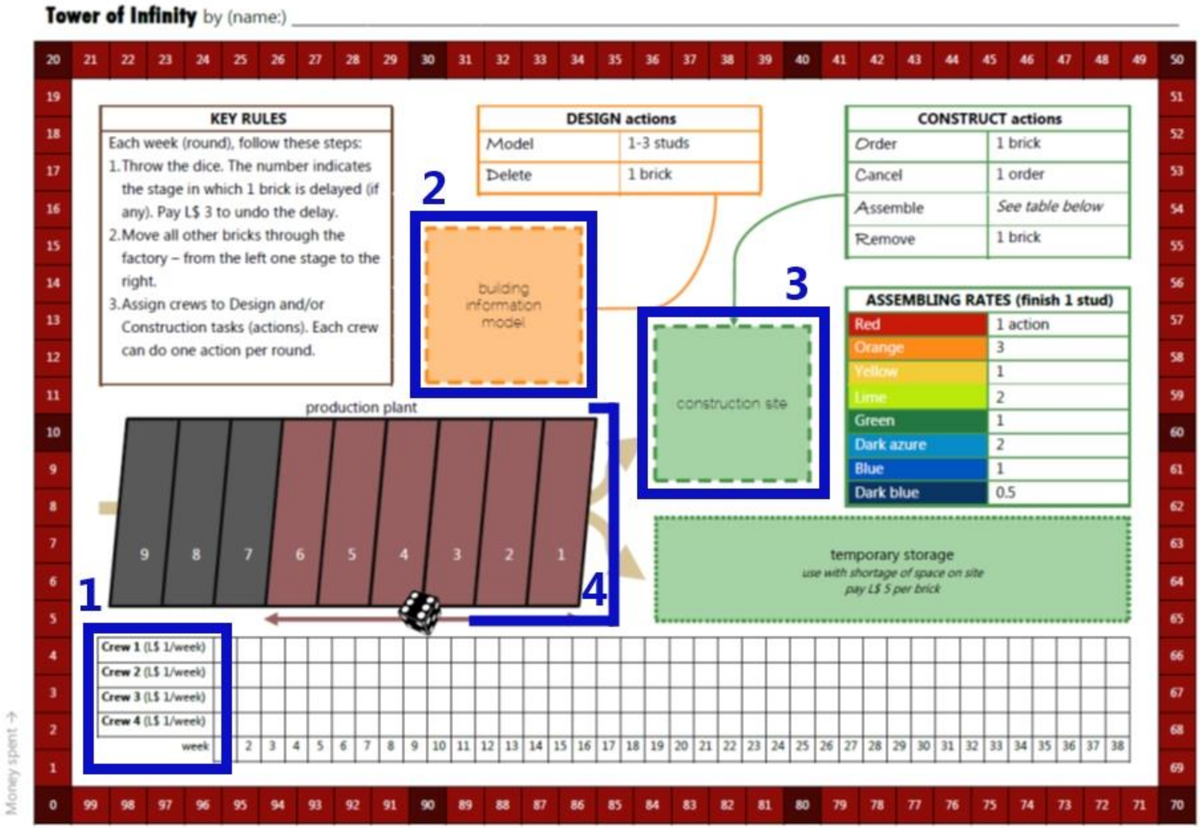


Figure 11: Play board of "Tower of Infinity" (van den Berg, 2015)

Materials are visualised as LEGO bricks. There are different forms and colours of LEGO bricks that represent different types of materials. Every player is given a sack of LEGO bricks that represent 'design-materials'. The contents are the same for every player, and players cannot share their design-materials with others. Next to these design-materials, there are also construction-materials. These construction-materials are shared with every player, and are placed unsorted in the middle of the table. After the procedure of the game is explained, all requirements are given to the players. These requirements differ between:

- Payment of the project (given in LEGO-dollars L$)
- Amount of weeks in where the design must be constructed
- Material-specific requirements (certain amount of studs of a colour)
- Building-specific requirements (certain amount of levels in the building)

Now that the players understand the game and know the requirements, all players start the game at the same time.

Gameplay

The game is played in time-steps of weeks. Before the week starts, a dice has to be thrown to decide on possible delays. If a number is thrown where bricks lie in the production process (this will be elaborated on later), these bricks will be delayed. However, this delay can be nullified by paying a fixed amount of LEGO-dollars.

After the delay has been dealt with, four crews can work on one task each. However, some tasks take longer than others, whereas some tasks even differ between different colours. The speeds of these tasks are either given in an entire brick per action or in studs per action. Studs are the small, cylindrical bumps that are on top of the LEGO bricks. The following tasks are available to each crew:

- Primary tasks:
    - <u>Model:</u> Place one of your design bricks on the design model (max. 3 studs);
    - <u>Order:</u> Choose an offer from the product list (representing supply offers), pay the sales price and place the construction brick on the corresponding stage in the production factory (max. 1 brick);
    - <u>Assemble:</u> Construct the brick on the construction site (max. amount of studs dependant of colour, ranging between $\frac{1}{3}$ and 2 studs per crew).
- Secondary tasks:
    - <u>Delete:</u> Take one brick out of the design model (max. 1 brick);
    - <u>Cancel:</u> Remove one brick that is in the production factory. 50% of the original price is returned (max. 1 brick);
    - <u>Remove:</u> Take one brick out of the constructed tower (max. 1 brick).

There are a list of rules that apply to these tasks:
- A crew can only work on one task and one brick per week;
- A brick on top of another brick cannot be modelled if the brick underneath has not been fully modelled yet;
- A brick on top of another brick can be modelled if the brick underneath is modelled, even if it is only finished in the same week;
- A brick can be placed on top of another brick, even if they are connected by only one stud;
- If a brick cannot be deleted directly (because other bricks in the design are on top of it), these other bricks need to be deleted first;
- A brick cannot be modelled and ordered in the same week;
- A brick can be assembled the same week as it arrives on the construction site;
- A design-brick can be built using multiple construction-bricks (For example, a brick with dimensions of 4x2 in the design-phase can be built using two 2x2 bricks in the construction-phase);
- If there is no room for a brick on the construction site when it comes out of the construction plant, it has to be stored in a temporary storage (this costs extra L$ per brick);
- A brick can be put onto its designed position at the moment assembly is started, not when it is completed;
- A brick on top of another brick cannot be assembled if the brick underneath has not been fully assembled yet;
- A brick on top of another brick can be assembled if the brick underneath is assembled, even if it is only finished in the same week;
- If a brick cannot be removed directly (because other bricks in the construction are on top of it), these other bricks need to be deleted first.

When all four crews are assigned a task (or if there are no tasks available to the crews), the tasks are fulfilled and the next week starts with rolling the dice. The game ends when all material- and building-specific requirements are met.

Post-game

After the game ends, the total costs of the project can be calculated. It is then checked which requirements have been met, and which haven't. When all players have completed their game and calculated their costs, a discussion is started with these players to understand how they gained awareness of some problems in SCM in construction. The aspects that were not noticed will be explained, so that the player understands all the characteristics of SCM in construction that were implemented in "Tower of Infinity".

B.2. Game-analysis

The analysis of "Tower of Infinity" will have the same layout as the explanation of the game, except that this game-analysis also contains a global discussion. Some of the SCM characteristics have not been mentioned in the procedure, because they are not directly visible before understanding the entire game.

Pre-game

Before the game starts, there are already characteristics of SCM in construction implemented, to make sure that the players do not have the freedom they think they have:

- In the sack with design-materials, there are far less different materials then the product list shows. This means that the players can only design – and construct – with a fixed number of materials from the start;
- However, the pile of construction materials does not take this restriction into account. This means that a lot of the construction materials cannot be used in the first place, due to the restriction of the design materials;
- A certain type of LEGO-brick that will have a high demand due to the requirements, is deliberately scarce in the construction pile, so that player that do not realise they will have to make unplanned changes to their supply chain.

This is to make the players of the game realise that they have to analyse the situation well before even starting on their project. These pre-game characteristics are mostly about material-scarcity in a construction project.

Gameplay

When all players start their project, they all have a vague planning about how to design and construct their project. However, there are many aspects in the game that need to be taken into account to let the project meet all requirements. First and foremost, the design, supply and construction chain must flow into each other nicely, so no time is lost. If the chain flows smoothly, the crews should also have a task to do every week. Choices about what material to design and construct first are also essential to keep the flow as smooth as possible, since some materials have longer assembly-times than others.

Secondly, the uncertainty of delay in the construction plant is of high importance, because this is something that *might* happen. However, even if it happens, there is the possibility to nullify this delay by paying a certain amount of LEGO-dollars. This trade-off should be considered every time a delay happens. The trade-off is between the time that there is left for the project, the amount of money that is left, combined with the workload of the four crews for the upcoming weeks.

Thirdly, the requirements should also be taken into account. If some requirements are forgotten or not kept in check, this might result in a significant costs increase or delay. In the test-round that has been played, one of the players forgot about a building-specific requirement (the tower had to be 4 level high) after he claimed to be done with his game. Because of this, this player had to remove and delete one of the old bricks, and model, order and assemble a new one. Especially if this is only realised at the end of the project, this will cost only more time and money. Also, at some point, a requirement might be rejected to make sure another requirements is still met. If it is certain that the time-limit is not met, one might try to at least keep the project inside of the budget by letting the project deliberately take longer.

Lastly, if the aspects that were mentioned in the pre-game are not taken into account well, this will also have an influence on the planning of the design. For example, in the construction-pile, there was only one LEGO-brick that all of three players had in their design. Because of this, two of the three players had to construct this brick with multiple smaller bricks, which costs more money and time. If they had known this sooner, they might've put this brick as the foundation of their structure, so they were the first to take this brick.

Post-game

After all players are done with their project, they are asked what choices they took and their reasoning behind it, as well as the question whether they saw any problems in their project that they would improve next time. This part is mostly for the players to understand all the aspects of SCM in construction, and to learn how earlier choices affected the problems they encountered later.

In the test game that was played, the two participants that had to use multiple bricks for one design-brick hadn't realised that the one other participant had taken the only single brick as quick as possible, and did not realise that this was part of one of the purposefully created situations. Here, one of the participants was also notified of the fact that he did not fulfil all construction-requirements yet, since he had only built three floors.

Another player also realised himself that his planning was not well-spread, because there were three weeks in-between all the work, where the crews were not able to fulfil any tasks. However, this was partly caused by delays that happened on a certain brick. If the player realised the gap that was being created, he should've nullified the delay by paying extra costs. While the player did realise this, it was when the gap was already at a reasonable level, and ultimately resulted in the time limit that was not met.

Appendix C: All versions of "Tower of Infinity"

For every version, all things that were added or edited with respect to the previous version are given here. At first, visualisation was not present, but lines were only printed in the console. It will be mentioned when visualisation is added. Of course, during the entire design-process, many bugs occurred and have been fixed, but because these do not represent the design-process, most of these bugs have not been included in this roadmap.

Version 1

The first version started with the code for each week. When the left mouse-button was clicked, a test was done on if there was delay (1/6 chance). If delay occurred, the player had the possibility to press P or D to pay and nullify the delay or to wait for the delay. Parameters (costs) were changed accordingly. In total, two .java files were available:

- Main.java
- NextWeek.java

Version 2

- Added three new .java files:
    a) Help.java: would be called at the first time a task had to be applied.
    b) Tasks.java: contained all code for applying each task.
    c) Bricks.java: contained a basis for creating bricks with a colour, width, and length. For now, only one brick was created with a red colour and 4x2 dimensions.
- Added an introductory code.
- Now, you could only go to the next week after a total of four tasks were applied.
- However, these tasks did not contain any action: it only said it was performing an action.

Version 3

- Added TaskActions.java: code to perform task-actions on the brick that was given as input. For now, this input was the first brick of the list; selection was not possible.
- Bricks.java:
    a) Now contains more information, including progress and all order-possibilities.
    b) Added code for creating a new, unique brick with its own object-code.
- Tasks.java:
    a) Added code for going to the next week, even when some crews are not doing anything yet.
    b) Edited code so that first, you have the press one of the tasks-buttons to go to the respective menu and click to 'select' a brick. Only for modelled, selecting the brick has been coded.
    c) Added code for testing if brick are finished modelling at the end of the week (for ordering)
- Help.java:
    a) Edited code so that this file is called every time H is pressed.
- Main.java
    a) Added code for cancelling a certain action and going back to the (main) menu.

Version 4

- Added input.java: code to set requirements.
- Tasks.java & TaskActions.java:
    a) Now contains all code for modelling, ordering and assembling a brick. Actions are also fulfilled for these. This way, the game can be finished as it should (except for deleting, cancelling or removing bricks)
- Help.java
    a) Added more help-code and requirements.
- General:
    a) Added code in different .java files to finish the game when all requirements are met, as well as showing all statistics when finishing the game.

Version 5 <u>(final framework version)</u>

This version is the final framework-version. It contains all information for each task, and is able to finish a game when all requirements are met. Because no visualisation is done yet, all tasks apply their action on the first item in the list that is given for each individual task.

- Tasks.java & TaskActions.java:
    a) Now contains all code for deleting, cancelling and removing a brick. Actions are also fulfilled for these.

Version 6

Visualisation starts from version 6 and further. Since the basic logic of the code is not changed (apart from selecting a specific brick instead of the first in the list), most changes will now be explained by visualisation.

- Visualisation:
    a) Added the design-platform, construction-platform, construction-plant and storage-site.
    b) Added testing bricks on these platforms to get the right vectors to place newly selected bricks
    c) Bricks on the construction-plant are moved one space to the right whenever a new task has to be applied (this is not according to the game, but was done to test the method)

Version 7

When clicking, changes camera and shows all model-able bricks. This version was a test to see if the method for showing the model-bricks worked accordingly.

Version 8

Added all code for modelling, including selecting existing and new bricks. Camera-work and visualised bricks are all done as well. Positioning the brick is possible in the BIM, but rotating the brick is not available yet. There is also no restriction on where to place the brick (bricks can be placed inside of each other) and bricks cannot be placed on top of each other.

Version 9

Added a Nifty-GUI screen as a start-screen.

Version 10

- Nifty: Added screen for selecting a certain order-choice when ordering a brick
- Visualisation:
    a) Added code for bricks going through the production process (delay does not stop the brick in visualisation yet, but does in background-code)
    b) Added code for placing a brick that arrived at the construction site, as well as placing a brick when it is assembled for the first time.
    c) Note: Bricks are still not placed on top of each other, and bricks can still be placed inside of each other.
    d) Added code for selecting and performing an action on either a brick on the construction site or from the temporary storage.

Version 11

- Visualisation:
    a) Bricks now stack on top of each other and are restricted on only being placed on top of finished bricks. The assemble-restriction still contains bugs.
    b) It is now possible to delete bricks from the BIM.

Version 12

- Visualisation:
    a) Fixed the bug from version 11 and so finished the visualisation for the game.
- Logic
    a) Added code that tests if the design and construction are identical.

While the game can now be played, any guidance through the game or information about the controls is missing.

Version 13 <u>(final first concept)</u>

- Nifty, added and expanded the following screens:
    a) Start-screen: basic introduction about the game.
    b) Main-screen: added planning that shows which task was assigned when
    c) Controls: contains all buttons that can be used and explanations.
    d) Rules/requirements/remarks: contain all rules regarding all tasks, as well as all requirements (and each progress) and assembly-speeds.
    e) End-screen: contains all endgame-statistics.
- Visualisation:
    a) Unfinished bricks are now semi-transparent.
    b) Text is added and shown in the top-left and bottom-left part of the screen that guides the player through the game. In the top-left screen, it is shown what the player has to do. In the bottom-left screen, progress and other results of actions are shown.
- Logic:
    a) Added the requirement that the construction must have a 4x4 base.

Appendix D: Extensive feedback of first version

Here, text from all individual feedback-sessions can be found. In the main report, this text has been compacted to points of improvement, as well as how these points will be implemented into the new version of the game.

Firstly, a feedback-session was done with the developer of the board-game of "Tower of Infinity". When starting the screens, the first main point of feedback was that the current explanation-screens contained far too much text. If this text is given to the players at the start of the game, a large portion will not read this. A tip was to use screenshots from the game to explain the game. When starting to play the game, the developer commented on how the keyboard-keys that he had to press for each task – M, O, A, D, C and R – didn't feel intuitive. It would feel more logical to click on the planning that is also shown on-screen. After playing the game for a while, he partly withdrew this feedback, because he became accustomed to the controls. What still did not feel as intuitive was that Q was equal to going back in the game, while escape quit the game. These two controls should be switched, or the escape-key should at least not quit the game. There were also some minor text and grammar issues that needed fixing. About the visualisation of the actual gameplay, there was two points of feedback. Firstly, the different parts of the game (like the BIM and construction site) are not named in the game itself. Players should know what these different places mean, and should be visualised in the game. Secondly, in the game, delay can only happen in the production plant in the last 6 weeks of the production. However, this is indicated nowhere. This should not also be added in the game itself, but also in the explanation at the start.

Secondly, the game was played by someone who had little knowledge about "Tower of Infinity". This participant also commented on how there was too much text at the start. The tip that was given was to add a tutorial at the start of the game instead of all the text. After scanning through the rules and controls, he started up the game. The participant started clicking on the planning, and didn't know what to do until it was instructed which keyboard-key needed to be pressed. Throughout this entire play-test, it was more guiding than actual gameplay. The requirements were also overlooked, due to the fact that these are not shown at the start. Because the participant mostly got stuck, feedback on the actual gameplay could not be given.

Thirdly, a feedback-session was performed with someone that had played the board-game of "Tower of Infinity". He also thought the rules and controls contained too much text, and could be explained better by use of a video. After it was explained that a video would be hard to implement, he agreed that screenshots of the game might also be enough. At least, an edit to the controls and rules-screen was needed. Additionally, he also didn't know what to do when he started the game, he had the feeling he had to click somewhere, while he actually had to press M to model a brick. After pressing N to model a new brick, clicking is needed to select a brick, and the arrow keys are used to position the brick. The participant mentioned that the constant switch from mouse to keyboard is very unintuitive. An idea from this participants was to add buttons on-screen as a copy of the keyboard-presses. This way, the player can choose by themselves what controls are used. The M-key can be pressed to model a brick, but there should also be a button on the screen labelled 'Model'. There could also be a button for modelling a new brick, but buttons for going back by pressing escape (it was explained that changing the back-key was already-given feedback) and for positioning the brick shouldn't be changed, because both of these actions are present in many games already and feel intuitive enough in the current gameplay. During the gameplay, this participant also didn't read the requirements due to this not appearing on a screen at the start of the game.

Lastly, the game was reviewed by two people that have knowledge in the field of construction SCM. Apart from the comments that are already mentioned above, they also mentioned that the aesthetics are underdeveloped. A background is missing, just like textures for the bricks. The solution for the tutorial containing too much text was to add multiple levels, where each aspect of the game would be explained one by one. Another point of feedback was that the model- and order-screen did not contain any information about assembly-speeds or requirements, while these two are essential for choosing bricks.

A problem that were not stated by the participants but observed by the observer, was that all players but the developer of the board-game did not realise when and if a brick was finished modelling. After the sequence for modelling a new brick was explained, each played kept modelling new bricks, without finishing any of them. The change from semi-transparent to opaque was not enough to inform the player that a brick was finished.

Appendix E: Images of development of "Tower of Infinity"

In this appendix, the visualisation between the first and final high-tech version of "Tower of Infinity" are compared. Sometimes, a picture from the low-tech game is also used for comparison. All of these comparisons can be found in Figures 12 to 15.

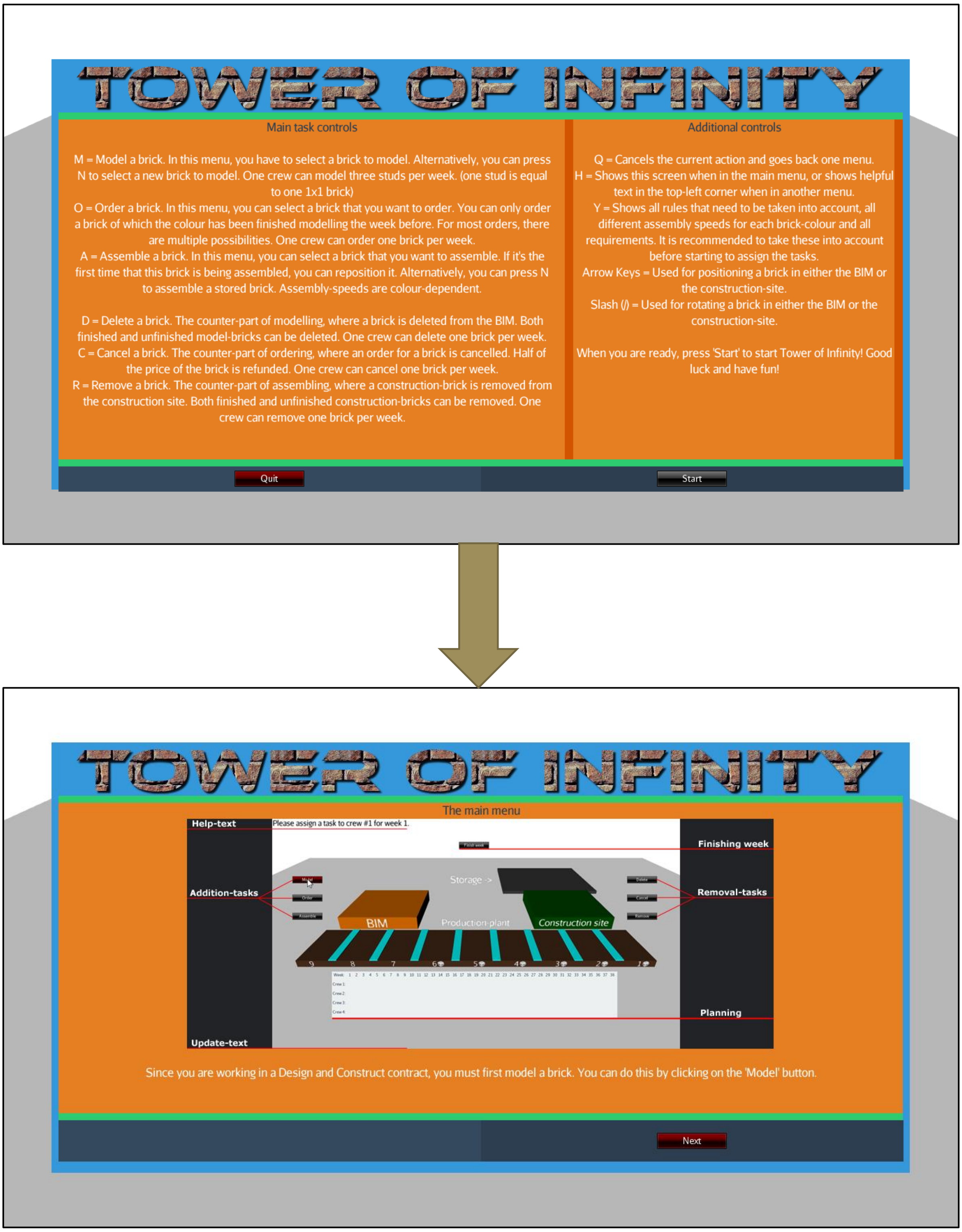


Figure 12: Instead of putting all information and controls on one page with text, the explanation is divided between multiple screens, where every screen is accompanied by a screenshot from the game for clarification

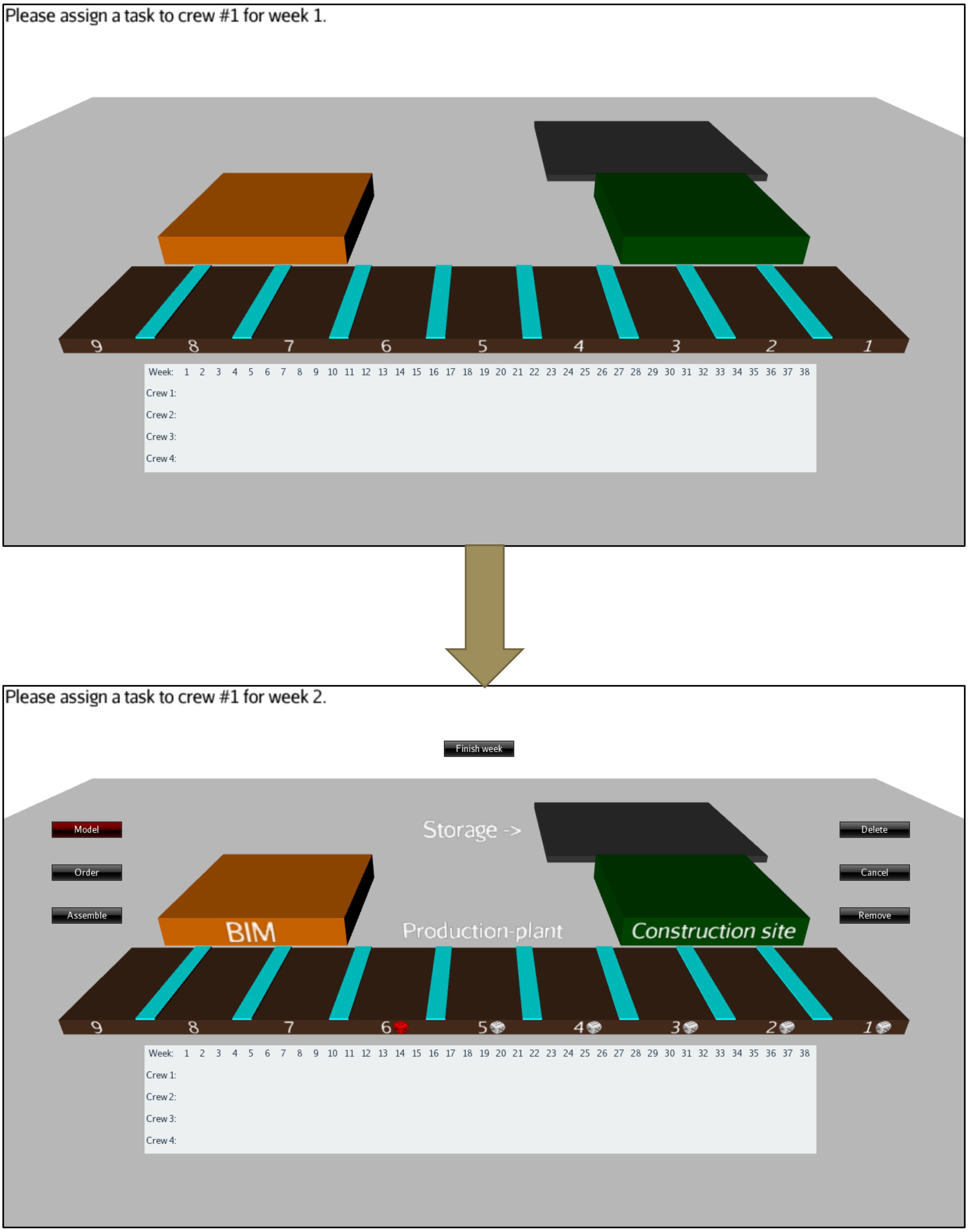


Figure 13: To guide the player through the game better, buttons and text has been added to the main menu. Additionally, dice have been added to the first six weeks to visualise delay, where the red die shows the week where delay occurred

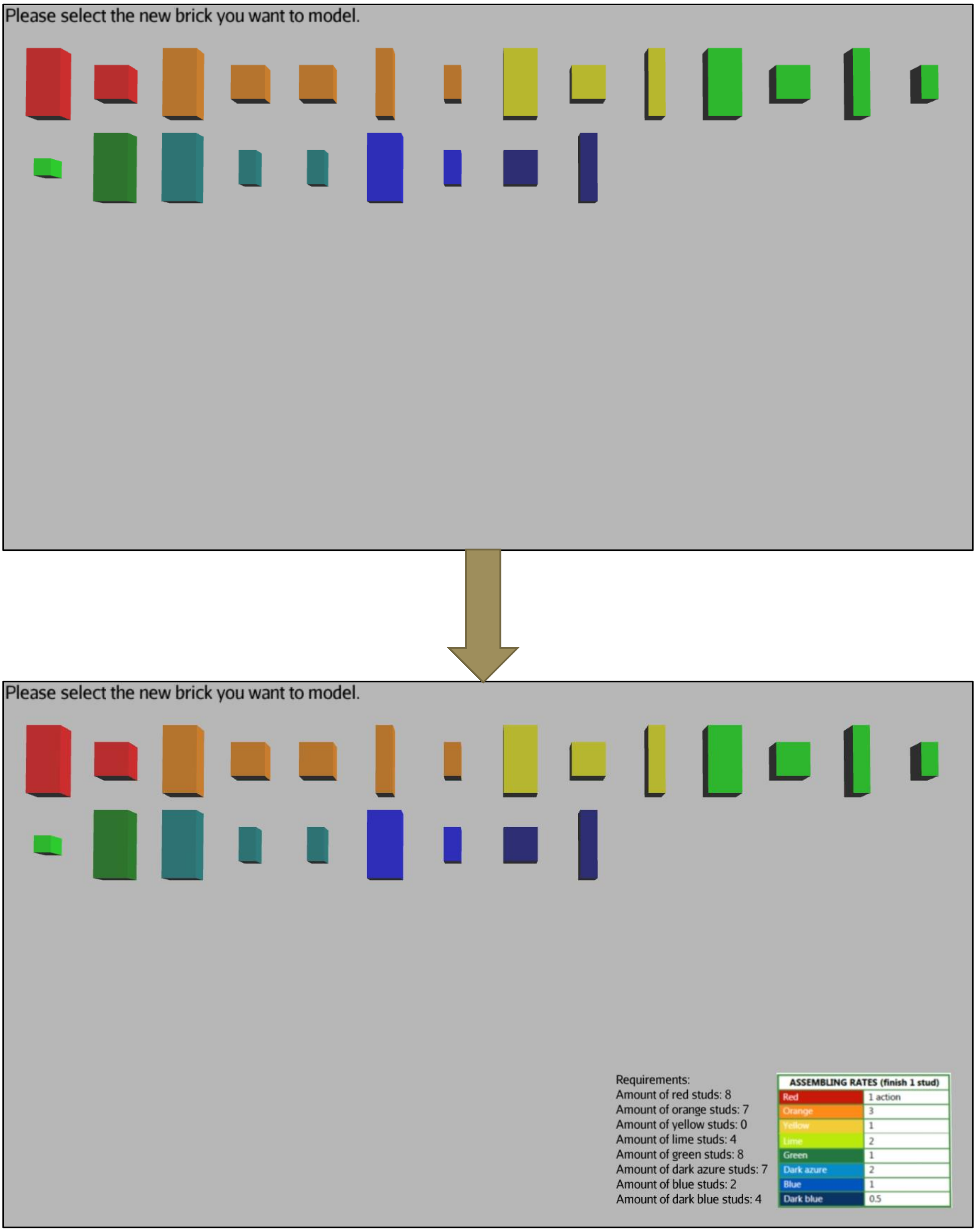


Figure 14: Because players should have all information available when modelling and ordering bricks, the requirements and assembly-speeds have been added to these screens

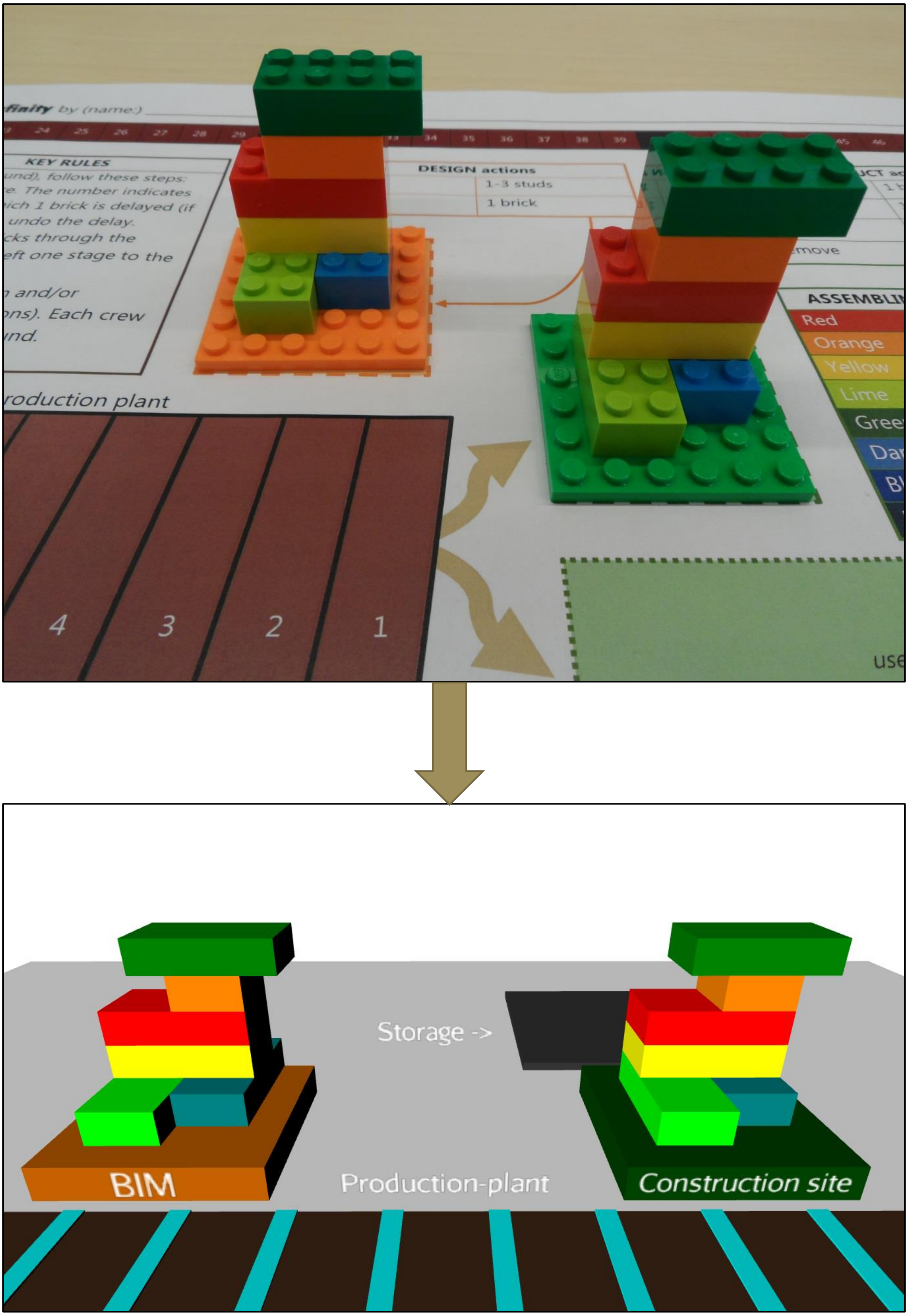


Figure 15: A finished project in both the board- and computer-game. Picture from (van den Berg M. C., 2015)

Appendix F: Elaboration of programming code

To give global insight into the programming code and its logic, a summary is given of each of the .java files.

Main.java
This file is the basis of the entire model, where all requirements, sounds, keys are initialised, and starting-values are assigned. A lot of variables are also initialised, mostly for remembering which brick was selected for cancelling actions.

The available methods of Main.java are given in Table 4.

Table 4: Methods of Main.java

Method	Explanation
Main(String[])	- Game-settings like resolution are assigned; - App is started.
simpleInitApp()	- Screen-display is set; - Screenshot-state is initialised; - Sets visualisation-basics like camera-location, -rotation and light-source; - Initialises parameters to call root-node and other variables from other .java files; - Initialises audio, key-presses and requirements of the game.
initAudio()	- Initialises audio.
initKeys()	- Initialises key-presses.
onAction(String name, boolean keyPressed, float tpf)	- Sends a signal to other .java files and methods based on keys that are pressed.
cancelAction()	- Contains all information for cancelling an action and going back to the previous menu.

Factory.java
This file contains all code for initialising and displaying the playing board, as well displaying all available bricks when modelling or ordering a brick. Lastly, the code that changes the material of a finished brick is also given here.

The only constructor in this file is Factory(AssetManager as), where through the board gets initialised through the AssetManager.

The available methods of Factory.java are given in Table 5.

Table 5: Methods of Factory.java

Method	Explanation
initBoard()	- Initialises all materials for the playing board.
makePlayerBase()	- Creates all geometries for the playing board.
visualiseBricks(int i)	- Returns a geometry based on a brick-class.
setFinishedMaterial(Geometry geom)	- Sets the material of a finished geometry to opaque.

GameAppStates.java
This file only possesses one constructor that calls the makePlayerBase method, and adds it to the root-node.

Bricks.java
This file is the basis of the ´Bricks´ class, that contains all information about each combination of colour and dimension of bricks.

In this file, a constructor is written that creates a 'Bricks' class. The information that is applied is given in Table 6. The available methods of Bricks.java are given in Table 7.

Table 6: Constructor of Bricks.java

Name	Explanation
ID	Number of the brick, as an alternative way to call a certain brick.
name	String in the layout of '[Colour] brick with [length]x[width] dimensions'.
colour	Colour of the brick.
length	Length of the brick.
width	Width of the brick.
progress	Used to keep track of the progress during modelling or assembling.
numOfStuds	Number of 'studs', which is equal to the area of the brick.
numOfSuppliers	The amount of different ways that this brick can be ordered.
price1	The price of the first choice.
time1	The production-time of the first choice.
price2	The price of the second choice (if not available, this value is null).
time2	The production-time of the second choice (if not available, this values is null).
price3	The price of the third choice (if not available, this value is null).
time3	The production-time of the third choice (if not available, this value is null).
timeUntilArrival	Used to keep track of the progress during production-time
choiceMade	Used to keep track of the price of a brick, for when it gets cancelled
numOfActions	Amount of actions needed to finish assembling the brick.

Table 7: Methods of Bricks.java

Method	Explanation
initBricks()	- Returns an ArrayList that contains all bricks from the game.
showBricks()	- Shows bricks that are available for the action that is selected.
createBrick(Geometry brick)	- Returns a Bricks based on characteristics of a Geometry.

ScreenStates.java

This file contains all code for changing and updating screens. Because the task-actions can also be called from buttons on the screen, a lot of code from Main.java has been copied to this file.

The available methods of ScreenStates.java are given in Table 8.

Table 8: Methods of ScreenStates.java

Method	Explanation
initialize(AppStateManager stateManager, Application app)	- Initialises the screen.
cleanup()	- Cleans up the screen.
bind(Nifty nifty, Screen screen) **onStartScreen()** **onEndScreen()** **update(float tpf)**	- All overwritten methods that do not contain any code yet.
tut1() **tut2()** **tut3()** **tut4()** **tut5()** **tut6()** **startGame()**	- Switches to respective screens.
quitGame()	- Quits game.
model() **newBrick()** **order()** **Pay()** **Wait()** **store()** **assemble()** **storageBrick()** **finishWeek()** **delete()** **cancel()** **remove()**	- Methods that can also be called by onAction() from Main.java; - Switches to respective screens; - Changes camera-positions; - Performs actions in-game.
productGuide() **goBack()**	- Switches between product-guide and order-screen.
createScreen() **updateRequirements()** **updateBase(boolean base)** **finishGame()**	- Updates screen according to a selected brick or requirements.
makeChoice(String stringChoice)	- Sends a number to choose a certain order-choice.

All code that visualises the different screens is given in the Interface-file screen.xml.

Tasks.java
This file functions as the main menu of the game. From this file, camera-positions are changed and text on the screen is updated.

The available methods of Tasks.java are given in Table 9.

Table 9: Methods of Tasks.java

Method	Explanation
startTasks()	- Gets called whenever a new task has to be applied; - Notifies the player when all crews are assigned a task.
finishDay()	- Allows the player to skip to the next week, but the method has to be called twice to confirm the skip
modelBrick() **orderBrick()** **assembleBrick()** **deleteBrick()** **cancelBrick()** **removeBrick()**	- Changes screens to all respective menu's; - Changes the "phase"–string to allow certain actions.
testProgress(Bricks brick)	- Tests if bricks have been finished in the previous week. If so, add them to a list of finished bricks.
pause(int time)	- Pauses the game for a set amount of time (mostly used for debug-purposes).
updateText(String text) **updateText2(String text)**	- Updates either the text at the top or bottom respectively. - When the top text gets updated, the bottom text also

TaskActions.java
In this part, all code is written for performing a task. Bricks are used as an input, and user-data from this brick is edited.

The available methods of TaskActions.java are given in Table 10.

Table 10: Methods of TaskActions.java

Method	Explanation
modelAction(Bricks brick)	- Adds 3 progress-points to the specified brick; - Progress of specified brick is checked and performs actions if brick is finished; - Adds an 'M' of the brick's colour in the planning.
orderAction(Bricks brick, Node rootNode, Camera cam)	- Places a brick in the order-process; - Adds costs to total and order-costs; - Adds an 'O' of the brick's colour in the planning.
assembleAction(Bricks brick)	- Adds 1 progress-point to the specified brick; - Progress of specified brick is checked and performs actions if brick is finished; - Adds an 'A' of the brick's colour in the planning.

deleteAction(Bricks brick)	- Deletes a brick from the BIM; - Adds a 'D' of the brick's colour in the planning.
cancelAction(Bricks brick)	- Deletes a brick from the production-plant; - Refunds half of the brick's costs; - Adds a 'C' of the brick's colour in the planning.
removeAction(Bricks brick)	- Deletes a brick from the construction site; - Deletes progress of requirements if brick was finished; - Adds a 'R' of the brick's colour in the planning.

PlaceBrick.java
This file contains all information about placing bricks and updating its positions.

The available methods of PlaceBrick.java are given in Table 11.

Table 11: Methods of PlaceBrick.java

Method	Explanation
placePrototype(Geometry brick, Node rootNode, Camera cam)	- Places the selected brick in the top-left corner of the BIM-grid; - Allows this brick to be moved; - Calls placeBricksUp.
placeOrder(Geometry brick, Node rootNode, Camera cam)	- Places brick in production plant based on its time until arrival; - Calls UpdateOrders().
placeOrderedBrick(Geometry brick, Node rootNode, Camera cam)	- Places brick in top-left corner of the construction site - Allows this brick to be moved; - Calls placeBricksUp.
placeStorageBrick(Geometry brick, Node rootNode, Camera cam)	- Places brick in storage based on amount of bricks that have already been placed in the storage. - Adds and removes brick to respective nodes; - Adds storage-costs.
placeConstructedBrick(Geometry brick)	- Allows the selected brick to be moved;
placeBricksUp(Geometry brick, Node siteNode)	- Checks if the brick has collision anywhere; - If there is, repeat until a situation without collision is found; - If there is not, check if there is collision when placing the brick down; - If that is the case, place the brick down until there is collision.
isThereCollision(Geometry brick, node siteNode)	- Checks if the brick has collision anywhere; - In this case, it is also checked if there is a brick directly above the brick that was selected; - Returns true if there is no collision, false if there is.
UpdateOrders()	- If there are more than two bricks in one week of the production-plant, sort these on basis of their area.

NextWeek.java

NextWeek contains all information for starting the next week. Delay is checked, and bricks are put forward in the process. Requirement-checks are also written here.

The available methods of NextWeek.java are given in Table 12.

Table 12: Methods of NextWeek.java

Method	Explanation
testDelay(Node rootNode, Camera cam)	- Resets week-related values; - Adds crew-costs; - Sets week (1-6) for delay to occur; - Changes colour of dice of said week; - If there is a brick in the delay-week, go to the delay-menu; - If there is not, iterate to the next week.
Wait(Node rootNode, Camera cam)	- The delay occurs; - Delayed bricks are put back one step in the production process.
Pay(Node rootNode, Camera cam)	- Adds delay-costs to total costs;
iterateWeek(Node rootNode, Camera cam)	- Moves all bricks one step further into the process; - Checks if a brick has arrived at the construction-site.
testRequirements()	- Returns true is all requirements regarding materials are met, false if not.
updateLevels()	- Updates the requirements that keeps track of the height of the construction.
testSimilarity()	- Returns true if the BIM and construction are identical to each other, false if not.
isThereBase()	- Checks if there is a 4x4 base on the ground floor of the construction.

Help.java

This file contains all text that is shown when the player asks for help. There is only one method, that updates the top-left text based on the phase that the player is currently in.

Input.java

Input.java initialises all requirements and bricks that are available for modelling and ordering.

Only one method is available, which changes the initialised Arrays at the start of Input.java into ArrayLists, so that these can be edited throughout the game.