BACHELOROPPGAVE:

Games for Science

FORFATTERE:
Christian Stokkebekk
Christian Holt
Øyvind Nordli
Morten Møller Andersen

DATO:
15.05.2013

Sammendrag av Bacheloroppgaven

Tittel:	Spill for vitenskap	Nr: -
		Dato: 15.05.2013
Deltakere:	Christian Stokkebekk	
	Christian Holt	
	Øyvind Nordli	
	Morten Møller Andersen	
Veiledere:	Jayson Mackie	
Oppdragsgiver:	Simon McCallum	
Kontaktperson:	Christian Stokkebekk, christian.stokkebekk@gmail.com, 40487477	
Stikkord	Spill, Vitenskap, Undervisning, IMT, Bachelor	
Antall sider: 109	Antall vedlegg: 6	Tilgjengelighet: Åpen

Kort beskrivelse av bacheloroppgaven:
Produksjonen av et rammeverk som tillater enklere tredjeparts utvikling av spill for realfaglig utdanning gjennom ImpactJS. Rammeverket inkluderer loggføring av rådata om spillerens oppførsel, som kan analyseres for pedagogisk verdi. Rammeverket håndterer også menysystemene i spillet. I tillegg innholder prosjektet et spill laget med dette rammeverket. Spillet er et 2D fysikkbasert hjernetrim-spill basert rundt magnetisme.

Summary of Graduate Project

Title:	Games for Science	Nr: -
		Date: 15.05.2013
Participants:	Christian Stokkebekk	
	Christian Holt	
	Øyvind Nordli	
	Morten Møller Andersen	
Supervisor:	Jayson Mackie	
Employer:	Simon McCallum	
Contact person:	Christian Stokkebekk, christian.stokkebekk@gmail.com, 40487477	
Keywords	Games, Science, Education, IMT, Bachelor	
Pages: 109	Appendixes: 6	Availability: Open

Short description of the main project:
Production of a framework that allows easier third party development of games for science education through ImpactJS. The framework includes logging of raw data about the player's behaviour, that can be analyzed for pedagogical value. The framework also handle the infrastructure of the game. In addition, the project contains a game made with this framework. The game is a 2D physics based puzzle game based around magnetism.

Preface

We would like to thank Simon McCallum and Jayson Mackie for all their help and guidance throughout this project.

A thank you to Vitensenteret Innlandet for their willingness to help with pedagogy matters, and for lending us the textbooks.

Thanks to Dominic Szablewski for giving us free licences to a helpful development tool.

Thanks to artists of both graphics and sound who joined in on the project with charitable enthusiasm.

A thank you goes to friends and classmates for their eagerness in testing.

Without all their help and contributions our project would not be where it is today.

Contents

Preface . . . . . **iii**
Contents . . . . . **iv**
List of Figures . . . . . **vii**
1 Introduction **1**
1.1 Project description 1
1.2 Group structure and organization 1
1.3 Employer 2
1.4 Supervisor 2
1.5 Target group 2
1.6 Purpose 2
1.7 Group proficiency and background 2
1.8 Limitations 3
2 Planning phase **4**
2.1 Target audience 4
2.2 Expected outcome 4
2.3 Game ideas 5
3 Outsourcing of asset creation **9**
3.1 Communication 9
3.2 Moral boost 9
4 Choice of development tools **11**
4.1 Comparison of development environments 11
4.2 Development tools used 12
4.3 Difficulties Encountered 12
5 Documentation **13**
5.1 Git - Bitbucket 13
5.2 Jira 13
5.3 Meeting logs 13
5.4 Work logs 14
6 Game design **15**
6.1 The overall visual theme of the game. 15
6.2 Recreating game levels in the real world. 15
6.3 Visual visualisation of the idea of a magnet. 15
6.4 Learning curve. 18
6.5 Unused features. 19
6.6 Arcade level 20
7 Entities **22**
8 Simulating Magnetism **35**
8.1 Calculating the magnetic force at a given point 35
8.2 Applying magnetism to the target 37
8.3 Making the effect of a magnet visible to the user 38

8.4 Manipulating the size and magnitude of a magnetic field 38
8.5 Exploring other types of magnets and magnetic fields 39
9 Framework 40
9.1 Development tools 40
9.2 Weltmeister 40
9.3 Button 40
9.4 Info screen 41
9.5 Pop-up 41
9.6 Shield 41
9.7 Metrics 42
9.8 XML 42
10 Metrics 44
10.1 Introduction 44
10.2 Privacy Policy 44
10.3 Current game metrics uses 45
10.4 Future possibilities 45
10.4.1 Replays 45
10.4.2 Player press reset 46
10.4.3 Heat maps 46
10.4.4 Most played maps 46
10.4.5 Quit rate on levels 47
10.4.6 Comparison on first and second playthrough of a level 47
10.4.7 Highscores 47
10.5 Implementation 47
11 Multi platform and browser support 49
11.1 Operating systems 49
11.2 Mobile 49
11.3 User input 50
12 Depth Jam 51
13 Learning Management System Integration 52
13.1 Possibilities 52
13.2 Tracking learning 52
14 Vitensenteret 53
14.1 Testing group 53
14.2 Overdue 53
15 Dragonbox 55
16 Further Development 57
16.1 Complete test-phase 57
16.2 Refining what we have 57
16.3 Balancing 57
16.4 New features 58
16.5 Achievements 58
16.6 Additional content 58
16.7 Expanding the framework 59
17 Conclusion 60
Bibliography 61

A Meeting logs **63**
B Email Logs **65**
B.1 Aquiring ImpactJS 65
B.2 Dragonbox 66
B.3 Artist 66
C Hand drawn images **70**
D Commit logs **80**
E Depth Jam Report **89**
F Pre-Project plan **94**

List of Figures

1 Old game . . . 10
2 New game . . . 10
3 Debug panel . . . 12
4 General look . . . 15
5 All magnets . . . 16
6 Magnet field . . . 16
7 The magnet . . . 17
8 Springboard . . . 17
9 Pendulum . . . 18
10 Rail . . . 18
11 Electromagnet . . . 19
12 Random peg board . . . 21
13 Arrows . . . 22
14 Button . . . 23
15 Credits button . . . 23
16 Credits Screen . . . 24
17 Electromagnet . . . 25
18 End Screen . . . 25
19 Gate . . . 26
20 Goal . . . 26
21 Info Screen . . . 26
22 Language . . . 27
23 The progress of lolgate. . . . 27
24 Magnets . . . 28
25 main Menu . . . 28
26 Menu button . . . 29
27 Mute Button . . . 29
28 Particle . . . 30
29 Pendulum . . . 30
30 Player . . . 31
31 Rail . . . 31
32 Respawn Field . . . 32
33 Restart Button . . . 32
34 Shield . . . 33
35 Springboard . . . 33
36 Switch . . . 34
37 Distance Squared plot . . . 36
38 Distance Cubed plot . . . 36
39 Plot over multiple exponential decay functions . . . 36
40 Exponential Decay Function Plot . . . 37

41 Iron shavings demonstration 38
42 An example level in weltmeister 41
43 Shield entity example 42
44 Database metrics 44
45 Mobile resolution example 49
46 Vitensenteret event 54
47 Email 1 from Dragonbox 67
48 Email 2 from Dragonbox 68
49 Email from artist 69
50 handDrawn1 71
51 handDrawn2 72
52 handDrawn3 73
53 handDrawn4 74
54 handDrawn5 74
55 handDrawn6 75
56 handDrawn7 75
57 handDrawn8 76
58 handDrawn9 77
59 handDrawn10 77
60 handDrawn11 78
61 handDrawn12 78
62 handDrawn13 79
63 Commit log page 1 80
64 Commit log page 2 81
65 Commit log page 3 82
66 Commit log page 4 83
67 Commit log page 5 84
68 Commit log page 6 85
69 Commit log page 7 86
70 Commit log page 8 87
71 Commit log page 9 88

1 Introduction

1.1 Project description

The goal of this project is to create a framework that could be shared with other developers interested in creating games for education. This framework handles common elements between these games. Menu systems, metric event logging and a level editor are all part of the framework.

In addition, we have made one game within this framework. The game is titled Magnets, and is a 2D platform based puzzle game that revolves around manipulating magnetic fields in order to guide a metal ball to a finish line. A great deal of time has been put into creating magnets and designing puzzles. Magnets features custom graphical and audio assets made by artists recruited for the project.

A cooperation was set up with Vitensenteret Innlandet. They were to act as a user group and provide testers in the target audience. Sadly, due poor communication, this fell apart and was not included in time for the project deadline.

1.2 Group structure and organization

The group consisted of the following personnel, each with their own role(s).

Christian Stokkebekk - Group Leader, Producer, Level designer, Programmer

Christian Holt - Lead gameplay programmer

Øyvind Nordli - Lead systems programmer, Webmaster

Morten Møller Andersen - Level designer, Programmer, Secretary

The group leaders role is to keep track of the main interests in the project, and to lead the team through any uncertainties. The group leader's vote breaks ties in the event of a stalemate in a group vote. The producer is the person responsible for communication with anyone outside the group. This includes artists and Vitensenteret. Additionally, the producer orchestrated and performed an interview with an external source. The level designers were the ones to conjure the levels and the puzzles within them. The gameplay programmer derived the necessary mathematics, and created the basic magnet entity which the entire game revolves around. The systems programmer created the framework. This includes metric logging, menus and game navigation. Programmers assisted in creating game entities, and general tweaking, balancing and bug fixing. The webmaster has provided and administered a web server for temporary hosting of the game for testing. The secretary wrote meeting reports and noted work hours.

1.3 Employer

There was no employer in the traditional sense in this project; a company that orders a product to be developed. Instead, a project with serious games was suggested by the lecturer, Simon McCallum, with academic interest in mind.

1.4 Supervisor

The official supervisor was Jayson Mackie, but Simon McCallum was just as much a helping guide. Jayson readily assisted with matters of math and code. It was with Simon, however, that the scheduled friday morning meetings were had. It was in these meetings that the important decisions that steered the project.

1.5 Target group

This project has three target groups. The first is any developer who might wish to use the framework. Said developer would have to possess proficiency with javascript, and a licence for impactjs. The framework gives a nice platform to start developing an educational game.

The second group are the players. Magnets is intended for children in upper elementary school, ages 8 - 12. This age group was chosen based on the curriculum taught in 5-7th year of norwegian elementary school [source norwegian læreplaner]. These children are taught science at a level significant enough to allow an interesting game.

The third user group are elementary school teachers that wish to utilize Magnet as - or in conjunction with - a lecture. An especially tech-savvy teacher might even be able to use and analyze the raw data collected by the metrics system in its current state.

1.6 Purpose

The overall purpose of this bachelor project has been to set what we, as game programming students, have learned the past three years into practice. It would also be nice if anyone was ever to actually use the framework to make more games. If Vitensenteret were to get anything out of Magnets despite our failure to use them as a user group, that would be very cool.

1.7 Group proficiency and background

Everyone in our group is attending the same bachelor degree, Game programming. None of us have considerable coding experience prior to the degree, so all developers have roughly the same background. Proficiency still varies, naturally.

One group member have some experience with media, but this is unlikely to impact the project, as graphical and auditory asset production is outsourced to freelancing artists.

The only considerable experience with the development tools lies with one group member who attended a game jam [ref ref ref] where he used impactJS.

All group members have experience creating 2D games, and have been part of other

projects of comparable size. The courses of the game programming degree have all been leading up to this project, and the student should be able to handle it well.

1.8 Limitations

All user data will be stored and saved in compliance with Norwegian laws regarding privacy policies. Because our target audience is beneath the age of consent, these laws are even more strictly enforced. [1]

There will not be any economic gain from this project, its goal is to educate, not profit. There are no salaries involved, and no foreseeable expenses other than traveling to meetings and interviews with external resources, i.e companies within the genre we're aiming at.

The software will be web based and accessible from all modern browsers. It is also important to take into account that computers in schools and other public facilities might have below average hardware.

2 Planning phase

2.1 Target audience

An important part of any project is to decide on the target audience for the product. The target audience affects a lot of the decisions made during the development process, and therefore has to be decided upon early in the process.

The decision was made to target children attending elementary school, specifically grades 5 through 7. This is around the time when natural sciences are first introduced, making it an opportune time for learning through new means. Aiming at elementary school children also ensures that the developers have a good understanding of most of the curriculum. Children at this age should be old enough to be familiar with a keyboard and mouse, or even mobile devices such as tablets and smart phones.

Generally children enjoy playing games, although preferences vary. Learning through games can therefore be interesting for a child in the target age group. This allows us to make some general assumptions as to gaming experience and abilities. Children absorb knowledge quicker than most [2], and may have an interest in topics outside the standard curriculum. This allows some exploration outside the standard curriculum when creating a game.

2.2 Expected outcome

Each game created has to have a set expected outcome. Without this the project will not have a clear track to follow. With the target audience set, it is necessary to decide what the game is expected to do. To do this a list of wanted elements was combined:

The game has to follow the curriculum

A teacher should be able to substitute a lecture with either a part, or the entirety of a game, choosing to either spend half the lecture lecturing, then playing the game, or perhaps vice-versa. The game could be used as a substitute for textbook problems, or even as homework. Playing a game as a homework task may tip both ways, either the game will not be played at all, or they will embrace the game and play it because it is fun.

The game should be an easy way for the teacher to provide information to his students in an innovative way. It should provide the teacher with a way to communicate subjects to the students without going entirely away from the standard method of teaching. It could be substituted for animations or illustrations, offering the students the opportunity to play around and manipulate the objects linked to the subject.

The game should teach by doing, not telling

The game should utilize game play and game mechanics to teach a subject, not just rely upon flow breaking, time stopping, questionnaires with a game built around it. The game play should be related to the subject in some way, so the player can make connections through the game to real life. A racing game that informs you about photosynthesis when you win a race would be a bad example. The game should teach the concepts of the subject, even if the player doesn't realize at first sight. Dragon Box would be a great example of such a feat.

The game should be fun

There are plenty of teaching games on the market. The important aspect is to try and take the best of both worlds and combine them into a product that is both fun, and that teaches the subject successfully. This is a very hard thing to do, and Shavian reversals [3] are easy to create. A game may start to feel like a chore if it isn't fun, doing nothing to encourage play. If the game is fun, the player may be immersed within the game itself, forgetting the fact that it is supposed to be a school task, that are usually considered to be boring. If the game can be enjoyed on its own, without considering the educational elements, and still do a good job at teaching the subjects, it will be a major success.

2.3 Game ideas

Coming up with game ideas that can relate to the curriculum of the target audience, while still fulfilling the requirements set by the expected outcomes is essential to the success of the project. Vitensenteret provided textbooks used to educate the target audience in schools, all over the country. Scanning through these books for exciting subjects and potential game ideas proved helpful in the design process.

A list of interesting game ideas was compiled:

- **Earth exploration**
- **Magnets**
- **Electric Circuitry**
- **Spore**
- **Chemical Reactions**
- **Sim city energy**
- **Space station**
- **Aerodynamic rocket builder**
- **Atoms & Molecules**

After compiling a list of potential games, the next step was to come up with preliminary designs for each one, then have a majority vote to decide which idea to pursue further. The preliminary designs were as follows:

Earth Exploration

Earth exploration would be a game centered around excavating the earth. It would be a side-scroller where you collect precious gems, minerals, oils, gas and fossils. There would be a shop where you can sell your cargo and purchase upgrades to various aspects of your mining operation. Perhaps moving to other worlds to explore the subterranean sections of other stellar objects.

The game would encompass missions in different categories, like exploration, collection, and discovery. Where you would explore a new area, collect a certain amount of mineral X, or discover a new element, a new fossil. Unexplored areas will have 'fog of war', in other words, it will be dark until the player has visited the area.

The game should be controllable by the mouse only to allow for touch-screen support.

The game should teach the different layers of earth, the nature of various gems and minerals, the fact that pressure rises as one ventures further down into crust. The game should be playable for anywhere between 20 minutes, up to 10 hours.

Magnets

Magnets would be a game where you move around as a character named Magne to manipulate the strength and polarity of magnets in the level to guide a metal ball to a goal. Magne starts out by only being able to move things, and then as you play you gain new abilities to do other things like changing the polarity of objects, or manipulating their field strengths. Magne may be able to magnetize himself so he can utilize magnetic fields by other magnets to solve puzzles.

In the beginning when Magne doesn't have the ability to magnetize himself, he navigates around the levels using ladders and slopes, pushing objects and magnets around to guide the ball into the goal. Each level should take only a few minutes at most, focusing on teaching small pieces at a time.

The game should teach the strength of magnetism, the decay of strength over distance [4], the polarity aspect of magnetism, explaining how magnets can pull or repel objects of different or same polarity.

This idea eventually evolved into the final product after removing the character.

Electric Circuitry

The game would be centered around guiding electricity from a power source to an electronic device. The game board would start with an incoming power level from different sources of electricity, like wind, coal, nuclear etcetera. Manipulating this power level through the use of resistors and creating closed circuits and powering different devices.

The game should educate the user about the inner electrical workings of a normal household, from how the power can be wired to which items conduct electricity. In

essence the should teach the basic principles of power, voltage and resistance. Each level should take a few minutes to play.

Evolution

Evolution would start you off in the dark sea of primordial earth where you kill and eat other fish or plants to gain experience points, or Darwin points. These Darwin points may be used to evolve parts of yourself that you want to make stronger, jaws for more damage, fins for faster movement speed, or scales for a more durable body.

Eventually this could allow you to evolve lungs, carving the way for a life above the dark curtain of the sea, allowing you to walk amongst the trees or soar amongst the clouds.

The game should teach the basic principles of evolution and natural selection.

Chemical Reactions

There were no real ideas about this game, although it could potentially be a good one.

Sim City Energy

Sim City Energy would have a city that the player would have to supply with energy. The player would have to decide which energy sources to use, choosing from wind, water, solar, nuclear, coal, fossil fuels etcetera.

The game should teach the importance of clean energies by polluting the cities of players who choose to prioritize the available polluting energy sources without researches clean energies. It should have an emphasis on teaching which energy sources are renewable, and why they are renewable. It should also teach which locations work best for the different energy sources, where to get the best effect from solar and wind energies, and avoiding building nuclear power plants besides the town park. The game should last around 10-30 minutes per level.

Space Station

Manage, upgrade and improve your own personal space station. The player will have to explore the solar system to find minerals, gases and other resources. These resources can be used to upgrade your space station, or your spaceship allowing for further and longer exploration trips into ever more dangerous destinations.

The game should teach the layout of our solar system, and central traits of the different planets, both gaseous and terrestrial. It should also introduce the nature of other celestial entities like asteroids, dwarf planets and moons. The game should play for about 10 minutes to 3 hours.

Aerodynamic Rocket Builder

Paragraph

Atoms and Molecules

There were no real ideas about this game, although there are several possibilities around mixing atoms into molecules and their interactions.

These ideas were presented to the advisers, and with their input it was decided the best idea was the magnet game. It looks very promising for the time frame and scope, and it's a subject the developers are familiar with. Magne was however dropped, as he was an unnecessary complication of the game play.

3 Outsourcing of asset creation

The projects programmers have a severely limited set of skills when it comes to asset creation. Programmer art is well, and good for placeholders, but for any final project it is a good idea to have some proper assets. Because of this it is beneficial to outsource the asset creation process to qualified personnel.

3.1 Communication

Communication with the artists were initially done by email or face to face meetings at game development events and seminars. As development progressed to later stages where more rapid conversing was necessary, communication was moved to the instant messaging client Skype.

Two graphical artists were brought in to create visual assets. They each fulfilled different tasks, and had no interactions with each other.

Eivind Eriksrud created the in-game sprites and tile sets. The project leader worked with Eivind on a game jam [ref?] prior to the start of this project. The level of cooperation was satisfactory, and both parties agreed on further cooperation.

William Stewart made sprites for menu screens, buttons and other UI elements. One of the group members is part of a longer term project of which Will is the founder.

Two sound artists were also brought in to work on sound effects and background music. A core group member had worked with Martin Kvale on a previous game jam, and was confiding in Martins abilities to create satisfactory assets. During the Join Game conference, further cooperation was discussed, and Martin seemed enthusiastic. Because of Martins affiliations to Krillbite (www.krillbite.com) and their project Among the Sleep, Martin had limited time to work on assets, and therefore introduced an associate that could pick up on some of the load. Stefan Hedberg then joined the project and has supplied several sound effects, in addition to Martins music and effects.

3.2 Moral boost

The radical change from having programmer created place-holder art, to the properly created art assets from a qualified artist can give a substantial boost to morale. The game feels and looks far more complete in a very short about of time. This effect was know to the development team, and so the timing of implementing assets were coordinated with when the effect of the moral boost had the greatest impact on the project as a whole.

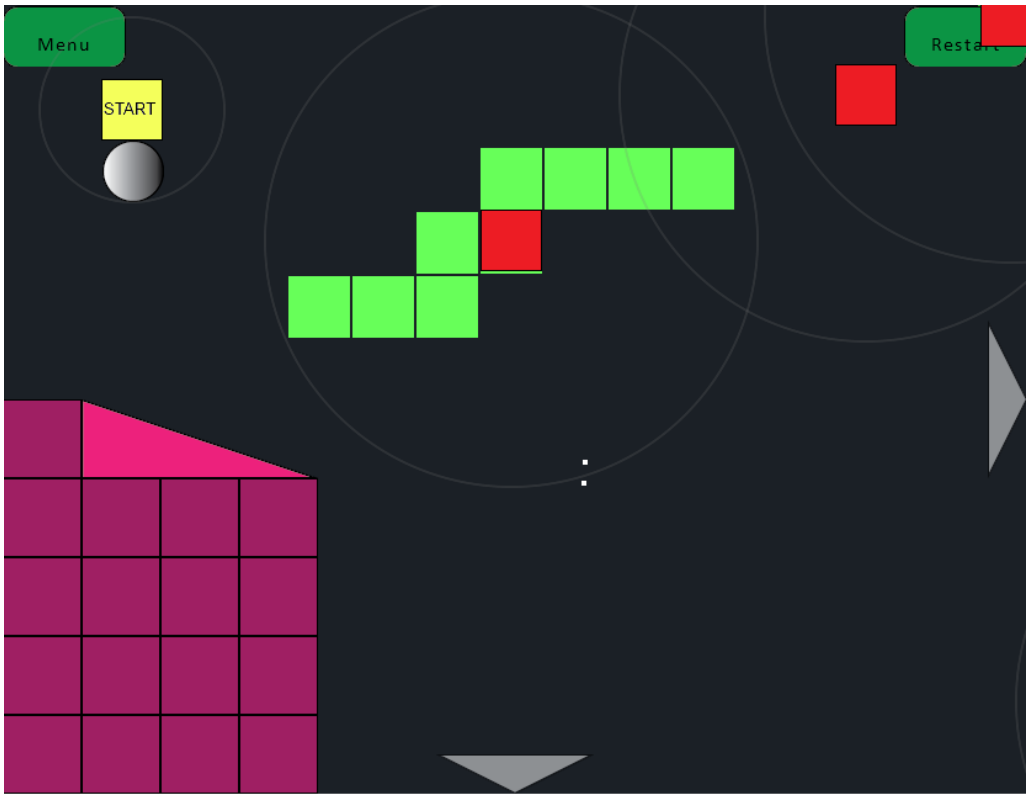


Figure 1: The old look of the game

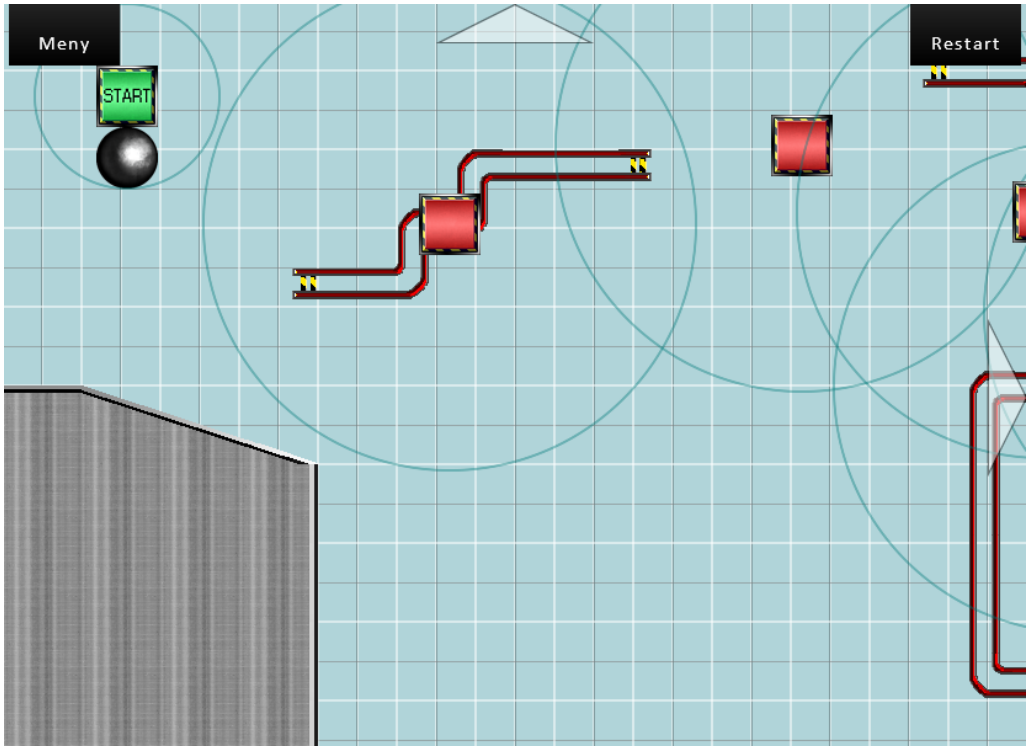


Figure 2: The new look of the game

4 Choice of development tools

Four different development environments were considered for the project. All of them support web based applications that span multiple devices and platforms. This reduces the required commitment to play the game by avoiding any extra installations or specific computer set ups. The four environments considered were:

1. Unity
2. LimeJS [citations / links to these?]
3. PhoneGap
4. ImpactJS

4.1 Comparison of development environments

When figuring out which environment to use, comparing their respective pro's and con's was paramount. Each environment has its strengths and weaknesses, and exploring these proved crucial in the decision.

Unity is an obvious option, with its ever increasing popularity and its support for multiple platforms and web utility, albeit this requires a plug-in. High licensing costs and strict licenses are a major downside. Unity requires licenses for every developer who interacts with a project. This includes asset creation. With the four core members, two graphics artists, and two sound artists involved in the development process, the costs will sky-rocket. Although school licenses are available, these require a school machine. This restricts working hours and opportunities as the development would have to be done inside a school lab.

LimeJS is an HTML5 game framework. It enables you to build games for touch-screens and browsers. Being open source is a great plus as it enables developers to read the source code, as well as develop without the extra licensing costs.

Phonegap is another free, open-source framework that enables you to create apps using HTML5, CSS and JavaScript. Phonegap supports compiling through the cloud with their phonegap:build project for those who don't want to get their fingers dirty.

ImpactJS is a JavaScript game engine that runs in all the popular browsers, even on mobile platforms. The only requirement is that the browser has to be HTML5 capable. It resembles LimeJS, but Impact comes bundled with weltmeister, a great level editor that really boosts productivity. ImpactJS has a wide range of code snippets available, and it has support for linking with Box2D for physics. Impact also comes with a great set of debugging tools. A debug panel, see figure **??** and debug code that enables drawing of collision boxes, velocities, trigger targets and more. Impact has a licence cost of $99, but through an agreement with its creator Dominic Szablewski it was offered to us for free,

including its source code. See appendix for email log.

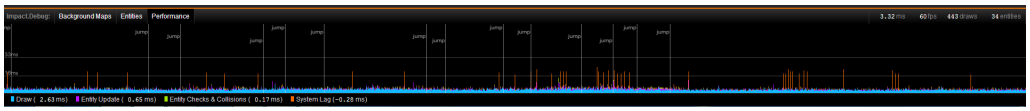


Figure 3: The debug panel of impact. It tracks events, lag, collisions, drawing and more.

4.2 Development tools used

The project makes use of ImpactJS together with Box2D for physics. The decision was made on grounds of costs and previous experience. A group member had experience with ImpactJS and its level editor from a previous Game Jam, though this is also the case for Unity, its main competitor for this project, ImpactJS comes free of charge.

Box2D enables us to simulate accurate physics without the extra work. The project makes a wide use of Box2D bodies, shapes and joints. The springboard and pendulum entities are based around a RevoluteJoint to enable its rotary motion, while making use of a WeldJoint to fasten a magnet to the entity body. See figure **??**

Making use of Box2Ds bodies and shapes have enabled us to have weights on the entities calculated from the shape, size and density of an object. The density can be used to increase the weight of smaller entities, like the iron shavings. This is done so they have roughly the same weight as the ball to better simulate the effect a magnetic field will have.

Impacts debug tools can help a lot when creating joints. It enables you to show the bounding boxes and their shapes around each entity, including the joints. It was very helpful throughout the project.

The set of development tools we chose have been crucial to the progress of the project.

4.3 Difficulties Encountered

The project encountered some difficulties, especially with the implementation of Box2D through ImpactJS. Within Impact, Box2D exists in its own namespace b2, and has been rewritten as JavaScript. The only documentation provided from Impact however is a link to the Actionscript version of Box2D. This meant that the documentation was not entirely accurate. Impact only supports version 2.0.2 of Box2D, although newer versions exist.

Box2D and ImpactJS also operate on different coordinate systems. This gave us several headaches over the course of the project. Without being mindful of which coordinate system one is operating on, entities can easily disappear off-screen. The main point of confusion here was when trying to convert between the two systems,e.g taking a spawn coordinate from the level editor, and trying to move the Box2D body to that position when resetting.

5 Documentation

Documentation is a big part of any project. Tracking and labelling was important so that significant information wouldn't be amiss at a later stage in the development.

5.1 Git - Bitbucket

The distributed version control and source code management software git was used in the project. The repository was hosted at Bitbucket.org. Bitbucket is a part of the Atlassian tool kit. Bitbucket offers free, closed repositories for up to 5 participants, and because of its affiliation with the Atlassian tools, allows for connection with JIRA (see section 5.2).

Git is widely used, and the core group members all had experience using git, and its console commands. Having a version control system in place for such a large project is an obvious choice, and git fits well with the mindset of the group.

5.2 Jira

JIRA is a project management tool. It was created to ease the task of managing a huge project. The Atlassian tool kit has a large selection of plugins that work well together with JIRA. One of these is Greenhopper, that is created to track Scrum, or other agile development models' sprints and issues.

This project was very small compared to the enterprise size projects JIRA was created for. With no set group member to manage and update JIRA daily, keeping up with issue closing, assignment and creation, it proved very cumbersome for a small development team such as this. Github and Bitbucket both provide issue tracking with a smaller system built around it for small projects that do not require the amount of development control of enterprise sized projects. Due to this JIRA was dismissed a tool only a few weeks into development, and the time was spent elsewhere.

5.3 Meeting logs

Every meeting throughout the project were logged by the group secretary. The logs were saved on Google drive for easy of access for every group member. These logs were also kept when the group met to work in pairs or as a group. The layout for the log files are as such:

Attendees: A list of who was present for the meeting.

Agenda: What was the plan for the meeting.

What was done: What was actually done.

Other notes: Notes from the meeting, plans for next time etc.

5.4 Work logs

The were no work logs kept for each working hour as most of the work were done in pairs or groups. Commit logs were however kept on Bitbucket with information on what was developed throughout the project with accurate timestamps and log messages. The meeting logs also kept the combined hours of the members attending the meeting. See appendix A for meeting logs and appendix D for commit log.

6 Game design

6.1 The overall visual theme of the game.

The graphical style of the game is simplicity, clean, sterile, and real.
The overall visual theme was used to give the game a scientific and realistic feel. This was picked because it was important that the game showed the target audience that what they are doing in the game, is possible to recreate in the real world outside of the game.

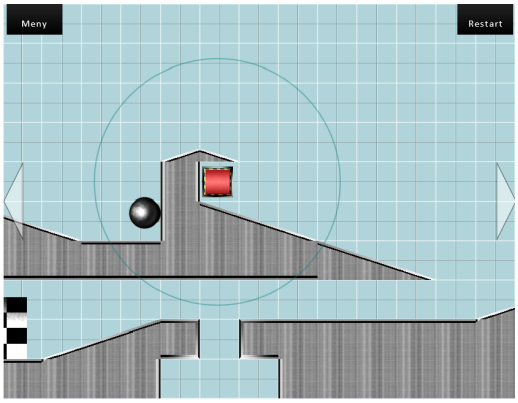


Figure 4: The general look of the game

6.2 Recreating game levels in the real world.

Each level was made in a way that makes it possible to recreate it using a simple set of tools in the real world.
How to make the level:
Start by making a background of wood.
On the background you put wood into the shape of the level.
Then add all the magnets and parts that are moving, hide all wiring behind the background.
End with adding a plexi-glass covering the experiment.
Now all that is needed is to drop the ball into the game and see what happens.

6.3 Visual visualisation of the idea of a magnet.

The magnet will have a base look, and then they will differentiate by geometric shapes, size, texture, magnetic strength and polarity, depending on what kind of magnet is used.

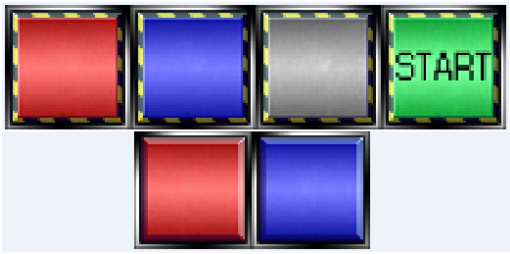


Figure 5: How the magnets appear.

Strength will be shown by the size of the magnetic field around the magnet.

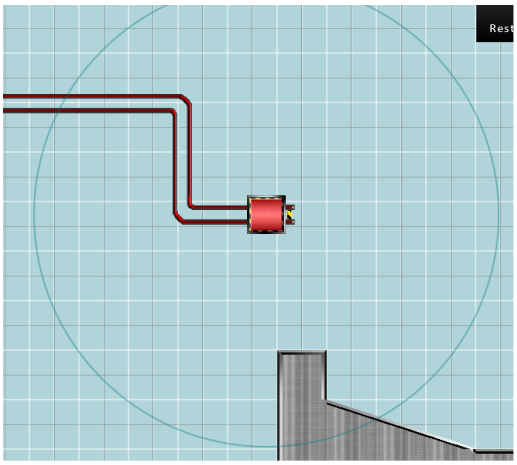


Figure 6: The big field around the magnet

A magnet can have one or more attachment, or characteristics. These are ways for the player to interact with the magnets.

A magnet with no attachments is called a Base Magnet. These are magnets that the player cannot interact with, they have a fixed position, strength, and polarity. Base magnets are represented by having a plain, solid texture.

The different types of magnet attachments the player will encounter throughout the game are:
Variable Magnet - Variable magnets are identified by moving the mouse over the magnetic field that are around the magnet, and it turning dark blue. The color change indicates that the player can drag the field closer or further apart from the magnet, by making it smaller, it makes the magnet weaker. And by making it bigger, it increases the magnets power. Almost all magnets in the game are Variable magnets, but those that are not have the field around them stay gray when the player mouses over the field.

Movable Magnet - Those are all the magnets that are not fixed permanently to

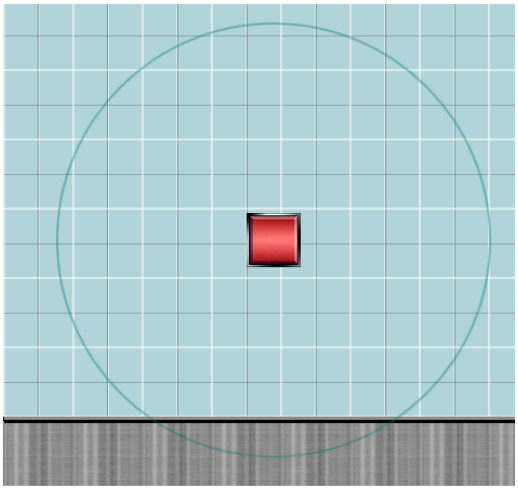


Figure 7: Basic Normal magnet

the background. Those can be differentiated from other magnets by not staying in one place.In the game there have been used three different movable magnets.

Springboard:
The magnet is now fixed to the end of a moving platform that starts of horizontal is hinged in the end that the magnet is not fixed. This magnet moves by getting pushed away from a base magnet with the same polarity, located under the springboard.

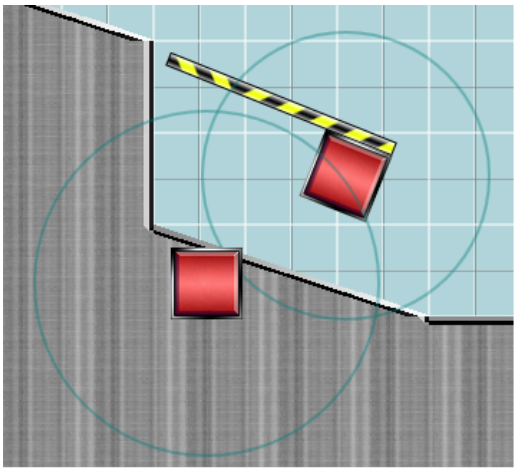


Figure 8: The Springboard

Pendulum:
Here the magnet is also fixed to a moving platform, only vertical. The pendulum can be differentiated from other platform, by it moving on its own from left to right, swinging.

Rail:

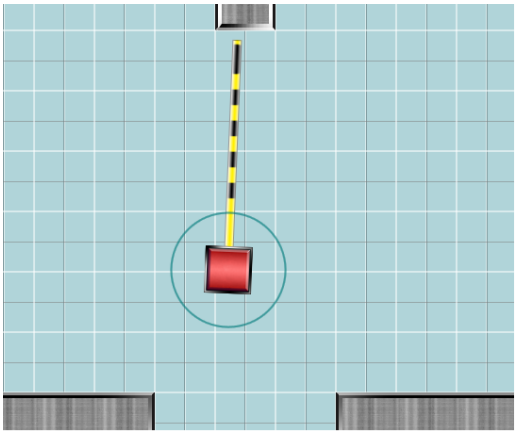


Figure 9: The Pendulum

The magnet is fixed to a rail on the background, and it moves following the rail by it self. The magnet will move from one side of the rail, to the other in a given speed, at a given pace.

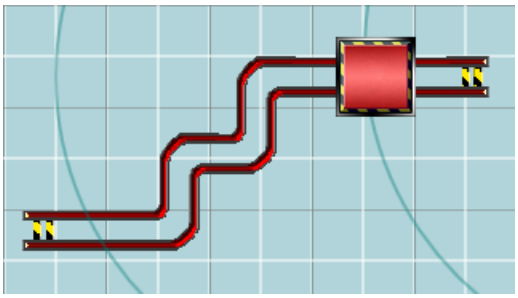


Figure 10: The Rail

Electromagnet - Electromagnets are controlled by an electric current, which means that the player can turn them on and off. The electromagnets will have a visual indicator in the form of a band of caution tape around the magnet core. Clicking this magnet will turn it from its normal color, to gray, and this will make it so the magnetism of the magnet is turned off.

Magnets can have more than one trait. For example there could be an electromagnet that is movable, or a magnet that is bipolar and variable.

6.4 Learning curve.

The order in which the levels are introduced, is designed to teach the player how the different obstacles in the game, are meant to be cleared.
So the flow through the game is used to teach new obstacles, and at the same time give

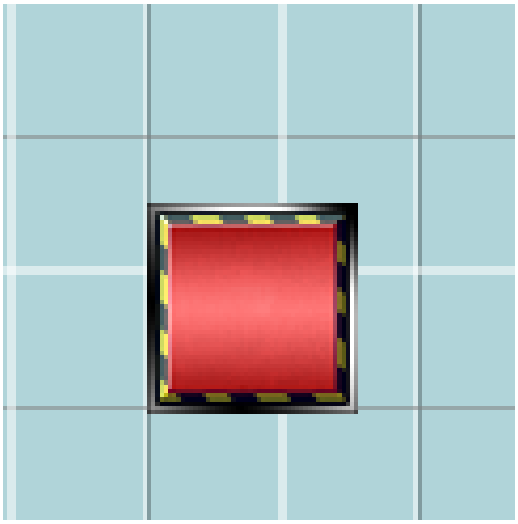


Figure 11: The Basic Electromagnet

more challenging obstacles using the techniques the player learned in earlier levels.

6.5 Unused features.

Features that was designed, but never used or developed:
Bipolar Magnet - Bipolar magnets can have their polarity altered. Clicking one of these magnets will swap the magnets polarity from positive to negative or visa versa. The player can identify a bipolar magnet by its distinct dual-coloured blue and red texture. The current polarity is represented by the color of magnetic field surrounding the magnet.
This mechanic was not implemented because the level design did not include any levels or puzzles that implemented this feature.

Fixed Step Variable Magnet (using clicks to change the power) - On some magnets, the strength of the magnetic field can be changed by clicking on the magnet. It will have three stages of strength; low, medium and high. You can spot these magnets by the three lights they will have on them. If the magnet is on the low setting, only one of these lights will be lit. Clicking the magnet will increase the strength of the magnetic field to medium, and the second light will be lit. This can be done again to set the strength to high, lighting the third and last light. Clicking the magnet on the high setting will set it back down to low.
This mechanic was not implemented because we instead implemented a draggable field that change the strength of the magnet instead of having it change when you click it.

Movable Magnet (that are moved by the player using a finger or the mouse) - Magnets can also be movable. This is shown by a rail underneath the magnet, the player can then drag the magnet along the rail to move it.
Because clicking most magnet will turn it off or on, and the act of dragging when clicked

close to a magnet will change its strength. Those features makes clicking the magnet and dragging it a nightmare for the player, so we had to abandon the mechanic.

Wire - Switches need to be connected to something, and this will be done through wires. Wires will run in the background. The ball cant collide with the wires, they will show what the switches are connected to.
This feature was not implemented because of time restrictions.

Conveyor belt - The conveyor belt will be a way to move one or more magnets in a loop. We will use the conveyor belt will be used to transport the ball or move platforms. Because of the magnet on a rail, this mechanic was ruled obsolete.

Entities that was developed but not used in the game:
Basket - This is an alternative to the normal goal. This mechanics notable features are, Its flat because it was meant to be inside a groove at the bottom of a level. Also it gives a score when the player hits it, used to add different score at the end depending on which one you end up hitting.
This entity was not used because there was not made any levels or puzzles that use it, the levels was not made because of time restrictions.

Slider - This entity was meant to control the level of different variables inside the game, most notable, the magnets power and field radius. Changes to how the magnet works, that power and field radius is changed by dragging, makes it so that the sliders was not used. There was not made any other mechanics or entities that this feature could be a intricate part of.

Slope - This was a feature that was implemented before the implementation of box2Djs and became obsolete after.

6.6 Arcade level

An arcade level was initially planned for the game. The design of this level were supposed to be similar to a peg game. The objective of a peg game is to drop a ball from the top of a board, and when it falls through it will hit the different pegs. The object is to hit the goal that give the player the most points.

This arcade level followed that concept, only instead of pegs, magnets were to be utilized instead. This would not give the same randomness as in the original game, but would make use of the players skill instead. The player would have to tweak magnets to get the right field strength to increase the chances of a ball hitting the higher valued goals.

A seed would be used to randomize the levels. This would give a nice diversity in the levels. It would also give the player the opportunity to play the same level twice if he chose to save the seed number.

An arcade level would also give the project a nice way to implement a highscore ta-

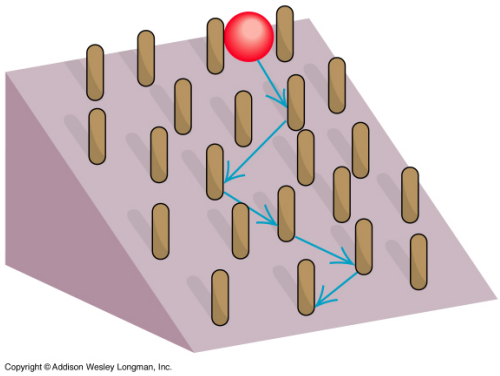


Figure 12: Example of random peg board. Illustration from J. F. Becker [5]

ble. This could increase the replayability of the game. If some of the seeds have a higher chance of earning the player a high score, those could be added to a black-list. This would remove the levels that is too easy to gain a highscore on.

When trying to implement this level, the view of the game was too narrow. This caused a problem when trying to play the game, because the level got to crowded with all the magnets. When trying to zoom out to get a more spacious view the drawing functions controlled by ImpactJS needed to be overridden. Most of the mouse targeting became mismatched as well. This ended up being to much work, and the level was cut from the final project.

7 Entities

To functionally utilize the level editor (Weltmeister), the various mechanics of the game is structured into Entities. Weltmeister places these entities to the desired coordinates, where they perform their designated functions. This chapter contains a complete listing of the entities featured in Magnets, and a discussion of their design.

Arrow

Arrows are made to indicate that the entirety of the current level cannot be displayed on the screen simultaneously. An arrow will appear at the edge of the screen in the directions in which the player is able to move the camera. Testing quickly revealed the need for indication that the level continues outside the default position of the camera. Without the arrows, testers were looking around confused as the ball rolled off the screen, as opposed to now, when they quickly realize how to keep up with the ball as it tries to escape.

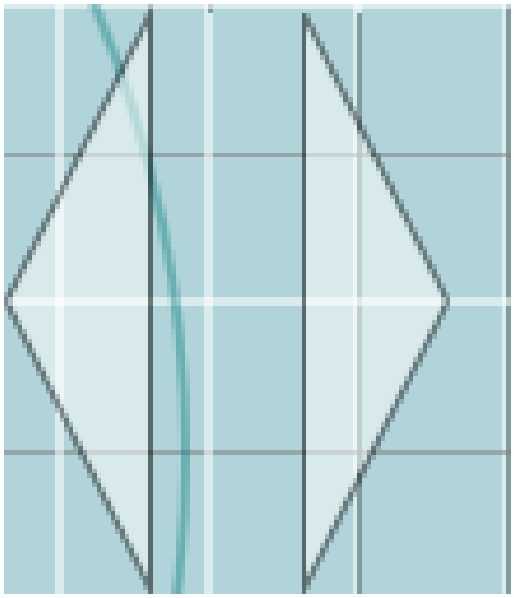


Figure 13: Arrows

The arrows are drawn from a sprite sheet that contains two sets of arrows for each of the four directions. One is transparent and the other is solid. The transparent arrow is drawn whenever the camera can be moved. The solid arrow is drawn in the directions the player is actively scrolling.

Button

The Button is part of the menu system included in the framework. Nearly every graphical user interface includes buttons, as it is a basic element in most graphics systems. The buttons in Magnets does not in any significant way differ from how buttons usually function. They can be clicked to perform the desired action, which in this case is to transport the player from the level selection menu to the indicated level. The button changes sprite to a highlighted version when the mouse is positioned over the button. This also plays a button highlight sound. Buttons can be outfitted with a custom text stored in an XML-file.

Figure 14: Menu buttons in different modes

Credits Button

The Credits Button is simply a button on the menu screen. Clicking it brings the player to the Credit Screen.

Figure 15: The Credits button

Credit Screen

This is a screen that can be accessed from the main menu. It contains text crediting programmers as well as artists to their work on this game.

Figure 16: The credits screen

Electromagnet

Electromagnet is an entity inherited from Magnet. It differs from its parent in a few ways. The main difference is that an Electromagnet can be clicked to be turned off. Turning off an electromagnet stops it from affecting other entities within its magnetic field. The strength and radius of the field is not altered, the effects of the magnet is merely suppressed. It can be compared to placing a switch on the electric current running through a real electromagnet.

The Electromagnet is graphically different from the basic Magnet. When turned off, it becomes a greyed out version of its usual red hue. In addition, whether on or off, electromagnets have a distinct yellow and black striped border. These visual cues are of course paramount to the players understanding of the puzzle at hand.

Electromagnets can be connected to one or more Switches. When the switch is pressed, the magnet is turned on if is off, and vice-versa. Some puzzles revolve around finding an alternative route to the finish line.

End screen

The end screen entity exists to show the user that they have completed a level successfully. A special congratulation sound effect plays and the screen tells the user which level they completed. The text is dynamically loaded from an XML-file.

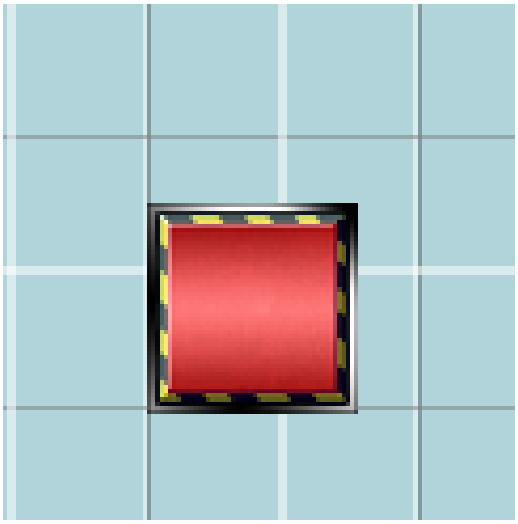


Figure 17: The base electromagnet

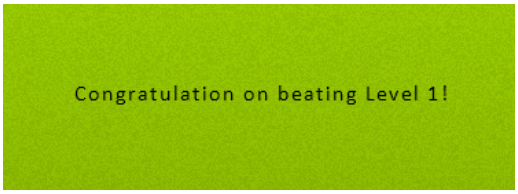


Figure 18: This is a general end screen

Gate

The gate was made as a simple wall. Its made of iron so magnets attract it. It is mainly used with an electromagnet to create a door that locks when some trigger happens. Normally the trigger is the user turning off the electromagnet keeping the gate open, by clicking on the magnet or the ball activating a switch.

Goal

The goal entity is the end goal of a given level. The goal activates when the ball hits it, ending the current level and showing the end screen for that level.

Info Screen

This entity is used to show the user information before each level. The information varies depending on what level it is affiliated with. The Info screen contains text and potentially an image to provide the user with new information.

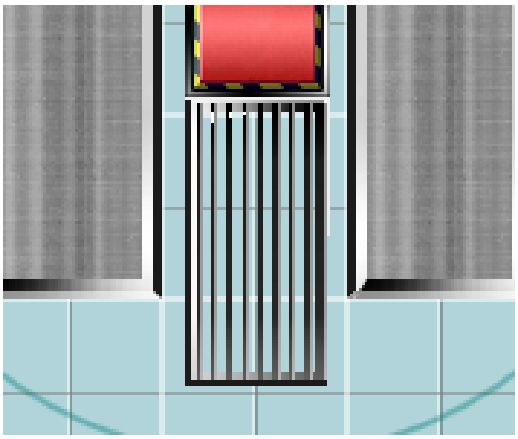


Figure 19: The gate

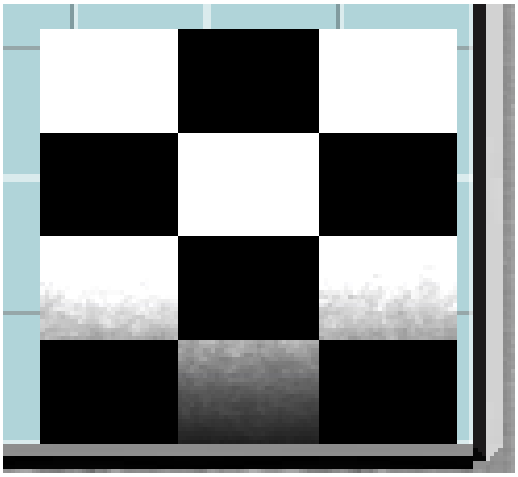


Figure 20: The goal

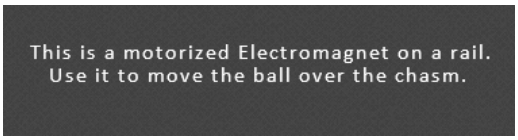


Figure 21: The general look of a info screen

Language

The language entity is similar to the Button entity. This button has an image instead of the text that other buttons have. The button's function as language selector in the game. The image on the button reflects which language it changes into by displaying the flag of the most prominent country that uses the language. The Norwegian flag is used to indicate that the language will be changed to Norwegian. When pressed will trigger a shield entity following the shield pattern.

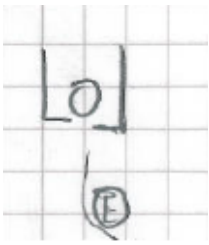
Figure 22: The language buttons

Lolgate

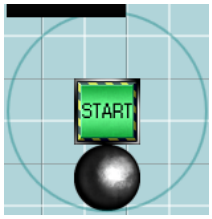
The lolgate entity is an electromagnet that acts as the starting button for each level. It has a unique color and text to show that it is the start button. The name comes from the original drawn picture of the start area for the ball, it looked like an "L" holding up an "o" close to another "l" (Lol). It was originally designed to have the lower line of the "L" drop down so the ball started rolling. During development it changed to the electromagnet because it was an easy way to get the user familiar with how the electromagnets visually looked. It could also give the user an "aha"-moment at a later stage when he realized it was actuality an electromagnet and not just a button.

Magnet

The magnet entity is the entity that the game is built around. For detailed information see chapter 8.



(a) The old Lolgate



(b) The new Lolgate

Figure 23: The progress of lolgate.

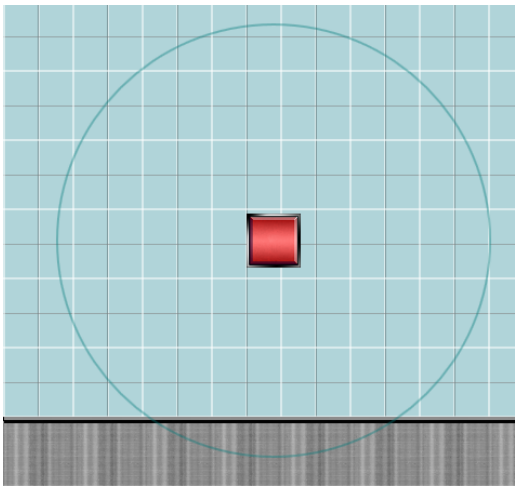


Figure 24: The magnet

Main Menu

The main menu is a separate entity from a normal background. It plays a unique sound track. The entity keeps track of which buttons are highlighted and plays sounds and effects corresponding to the unlock status the user has for the levels. The entity also keeps track of keyboard and mouse inputs to navigate the level buttons.

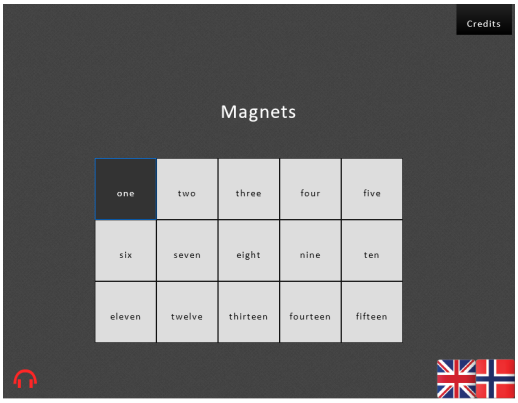


Figure 25: The main menu look.

Menu Button

This button entity exists within the game levels. It brings the user back to the main menu from the game. It brings up a box in accordance with the shield pattern.

Figure 26: The look of the menu button

Mute Button

This entity is a button that's only function is to turn the game volume on or off. This is not just music, but all game sounds.

Figure 27: The mute button

Particle

The particle entity is created to help the player by finding the right strength on the magnets. The particle is essentially iron shavings falling out from the users mouse pointer. They can be used by hovering the mouse inside a magnetic field. This will show the player where the ball is most likely to go. There isn't a lot of mechanics in the game that requires this kind of finesse, but for the few that exists, it is a good help.

Pendulum

The pendulum is a long thin platform that is anchored with a revolution joint at one end. This makes the platform able rotate around its top anchor point. An electromagnet is spawned at the lower end of the pendulum. In the revolution joint there is a motor that can power the pendulum from side to side. This entity can be used both as an obstacle

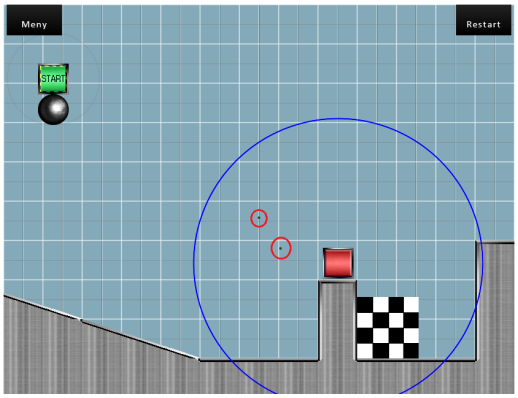


Figure 28: The Particles.

or as a benefit depending on how the level is designed, and how the entity is specified in the level editor.

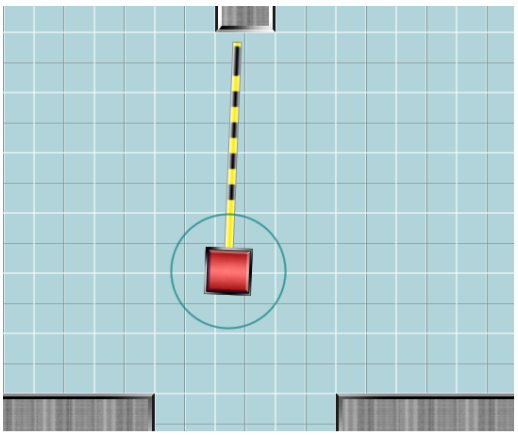


Figure 29: The pendulum

Player

The Player entity is a sphere shaped object with a metallic sprite added on. The user cannot directly control the Player entity. Instead the user manipulate the world around the Player entity, and guides it to the goal. The Player entity is in essence just a ball that is affected by magnetic forces.

popUpMessage

This entity is not used in the game, but it is part of the framework so it is still in the usable-entities list. This entity was intended to pop-up bringing information to the user containing anything from hints, to achievement unlocked messages.

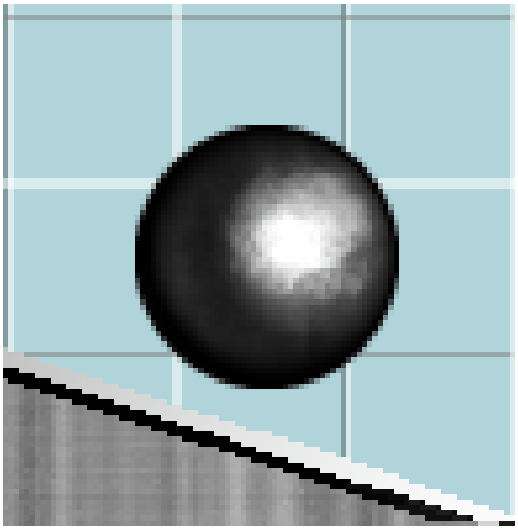


Figure 30: The Player, or ball

Rail

The Rail entity is a track that a magnet follows from one end, to the other. The User will use this to move past obstacles or over great distances.

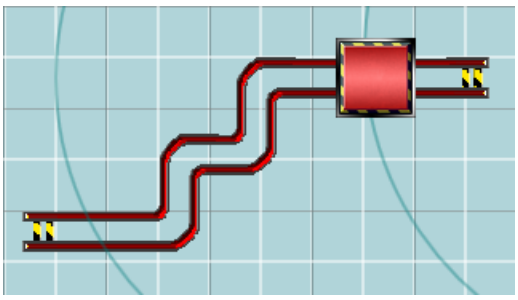


Figure 31: The Rail

Respawn Field

This entity activates the restart function of the level when the ball hits it. It is added to all borders of the game level, including down all the holes. This triggers a game restart if the ball falls outside of the level.

Restart Button

This entity is a button that activate the restart function on all the entities that has one. Magnets does not have a restart function so that if the players have made changes to the magnetic fields and power statuses, they will not revert back to their original positions. Rails have a restart function, so that the player can send them back to the start, this is to shorten wait time in some instances.

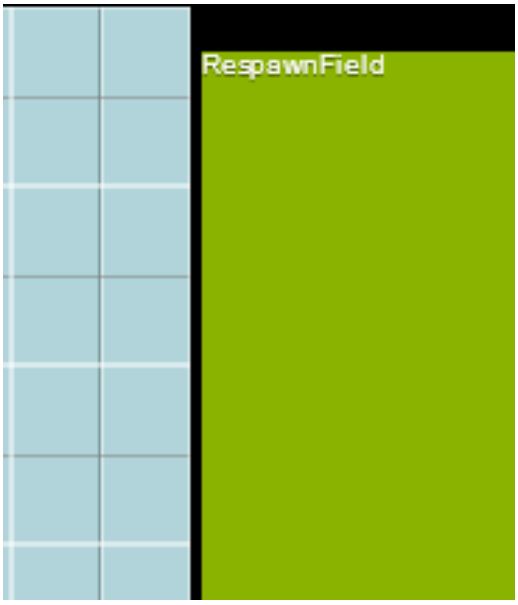


Figure 32: The Respawn Field

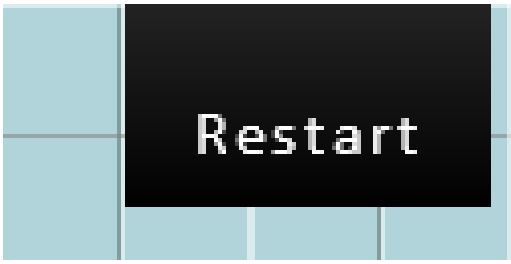


Figure 33: The restart button

Shield

This entity functions like a query window. It is used to ask the user “Are you sure?” type questions. It has customizable texts, as well as two buttons with customizable text.

Springboard

The springboard is a platform that is hinged at one side. The other side have a magnet attached. This entity is used as an adjustable ramp for the ball to jump obstacles or cross gaps. The springboard is adjustable by changing the power of the magnet that is positioned under the springboard magnet. The similar polarity of the two magnets push each other away and this raises or sinks the ramp.

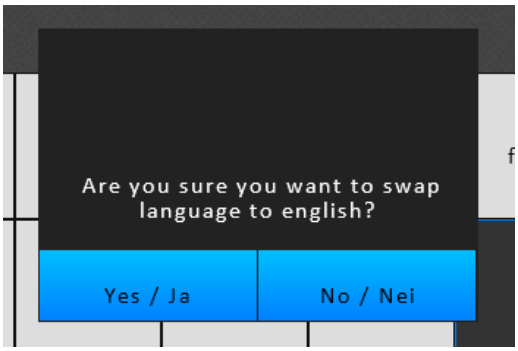


Figure 34: The shield example

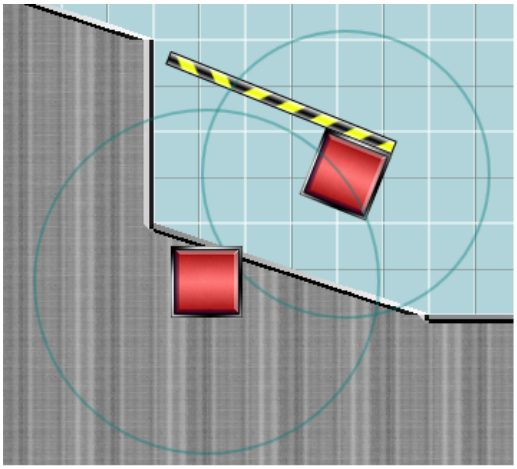


Figure 35: The springboard.

Switch

The switch entity's main function is to turn magnets on or off. It changes its own state and calls the switchTriggered function of the magnet it is connected to when the ball rolls over it.

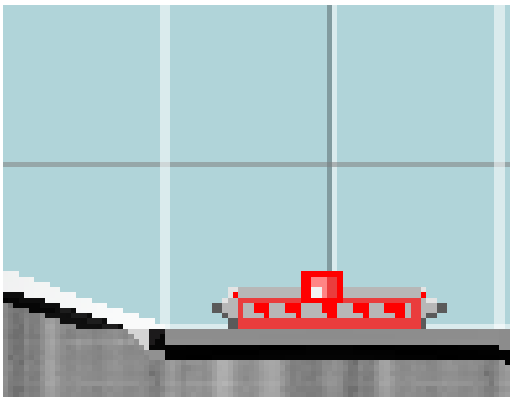


Figure 36: The switch.

Trigger

The trigger entity utilizes impactJSs 'check' function to check for 'collisions' with certain types of entities. It can therefore trigger certain events when the ball rolls over a certain location. This entity could be utilized in many interesting ways.

8 Simulating Magnetism

This project requires the simulation of magnets in a visually realistic manner. There was no requirement to have the simulation be mathematically perfect, so long as the user could see the connection between events in the real world and events in the game. To achieve this we broke the problem down into four different, smaller parts:

1. How to calculate the amount of magnetic force to apply.
2. How to apply the magnetism to the target entity.
3. How to make it visible to the user where and how the magnet has an effect
4. How to manipulate the field size and magnitude interactively.

Initially we investigated mathematically exact simulation of magnets. The computational complexity of the full simulation was beyond the scope of this project. [6] The primary objective of this game is that the player will understand the general concepts related to magnetism, while having an enjoyable experience. The simulation therefore is close to reality without being a perfect simulation. Most players are unlikely to notice a difference between the performance of the virtual magnets. and real life magnets. [7] It is a common feature of computer games to abstract some parts of the physics of the real world to allow for more interesting interactions. We have chosen to keep the interactions close to real interactions, but have allowed for some slack.

8.1 Calculating the magnetic force at a given point

When calculating the force vector, a $magneticForce$ variable is used. This represents the amount of force, from 0 to 1, that is going to be applied. When figuring out how to calculate this variable, we tried a couple of different mathematical functions to see how they worked in-game.

We wanted to try and simulate the important factors of a magnet, namely the strength of it when you are close to it, which is far stronger than gravity, and the rapid decay of strength over distance. [4]

In reality the calculations are very complex [6], but because of our limitation of scope we kept to simpler functions, such as $\frac{1}{distance^2}$ and $\frac{1}{distance^3}$ illustrated below by figure 37, and figure 38 and exponential decay functions.

Both of these equations give a very rapid decay. In-game the default magnetic field radius is 250 pixels, so a direct correlation has to be ruled out if either of these functions were to be used. An exponential decay function provides higher control over the decay rate by changing exponential component, demonstrated by figure 39.

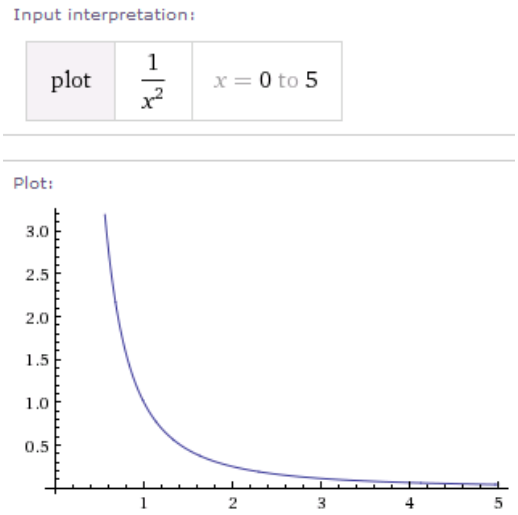


Figure 37: Plot of the function $\frac{1}{\text{distance}^2}$ from $x = 0$ to $x = 5$.

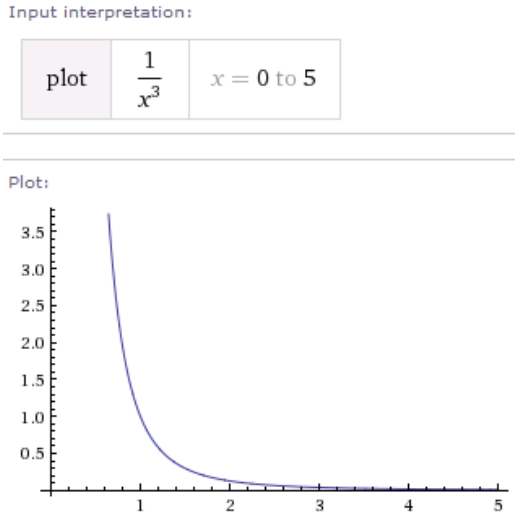


Figure 38: Plot of the function $\frac{1}{\text{distance}^3}$ from $x = 0$ to $x = 5$.

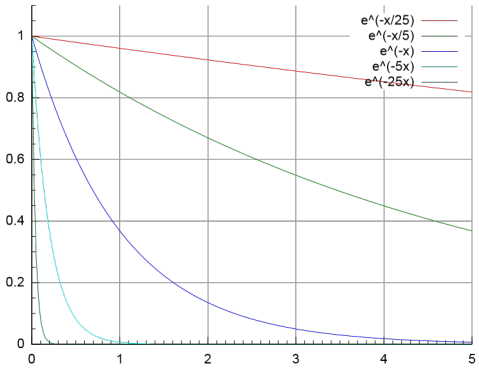


Figure 39: Plot of multiple exponential decay functions that demonstrate the possibility of higher control and flexibility. Source: http://en.wikipedia.org/wiki/Exponential_decay

we tested several different exponential components, using the formula

$$e^{-(|\mathtt{distanceVector}|/\mathtt{ratio})}$$

where $\mathtt{distanceVector}$ is $|(X_1 - X_2, Y_1 - Y_2)|$ and $\mathtt{ratio}$ is a quotient of the field radius. For a default sized field $\mathtt{ratio} = -250/5 \rightarrow 50$. This looks very lenient if you compare it to figure 39, but when the x axis is directly linked to the distance in $\mathtt{pixels}$ it works very well. As can be seen in figure 40, it resembles the functions shown in figure 37 and figure 38 closely. The decay rate transfers very well into the game as well as resembling actual decay functions in gravitation and electromagnetism.

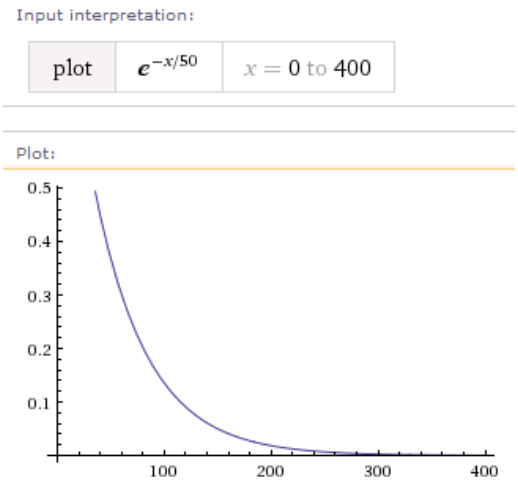


Figure 40: Plot of the exponential decay function $e^{(-x/50)}$.

8.2 Applying magnetism to the target

To solve issue nr. 1 each magnet holds a list of entities it will apply magnetism to. Entities such as the ball, the gate and other magnets are added into this list. Each magnet also holds a variable 'fieldRadius' that represents how far the field reaches. In our simulation the magnetic field is represented by a circle. This seems to work well for gameplay purposes.

During the update loop the distance between the magnet and each of the entities in the list is calculated using the standard pythagorean theorem:

$$c = \sqrt{(a^2 + b^2)}$$

where a represents the difference in distance between X_1 and X_2, and b represents the difference in distance between Y_1 and Y_2.

If the target entity is within the fields radius, several new calculations are done. First the unit vector of the X and Y components of the distance vector are calculated. Then, the magnetic force is calculated (details in section 8.1). Lastly, the polarity of the target entity is checked. If it has an equal polarity to the magnet performing the check, the force direction is turned. In other words it now repels.

The total sum of the force applied to the target is then calculated for x, and y separately, using this formula:

$$force = forceDirection(\frac{magneticForce * unitVector * fieldMagnitude}{mass})$$

$forceDirection$ signifies whether to repel or attract the target entity and will either be 1 or -1. $unitVector$ is the x and y components of the unit vector calculated earlier. $fieldMagnitude$ is a coefficient used to increase the strength of the field. This coefficient is necessary because the function used to calculate the $magneticForce$ returns a number in the range $[0..1]$

The resulting force vector is then applied to the entity using box2Ds ApplyForce function.

8.3 Making the effect of a magnet visible to the user

In order to make the effectual radius of the magnetic fields visible, we draw a circle around the magnet corresponding to the $fieldRadius$ variable that each magnet object has. Alternative ways of doing this was considered, especially weighing on the possibility of drawing a filled circle with a transparency related to the power of the field at each point. This could have led to a nice overlap effect when magnetic fields interacted and combined their powers.

To make it easier for the player to see how the magnetic field affects the ball, we added iron shavings, or particles, that drop from the mouse as you move it around the screen (see figure 41). These iron shavings can be used to observe the power of a magnetic field at a certain point, in a visual way. These cues can then be used to plan the next course of action.

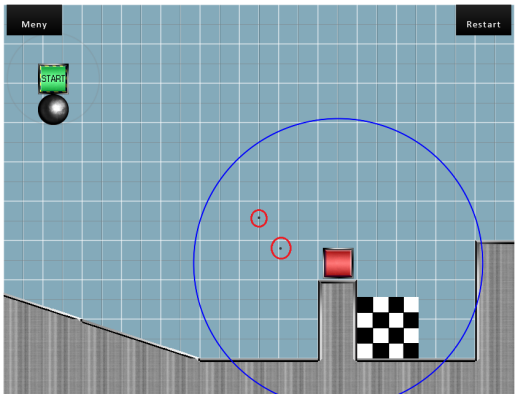


Figure 41: Iron shavings (marked in red) in action. An active magnetic field, and it's circumference circle can also be observed.

8.4 Manipulating the size and magnitude of a magnetic field

A major part of the game revolves around manipulating the magnetic fields of magnets. In the real world, electromagnets can have their magnetic fields manipulated by increasing the electric current running through them [8]. In an attempt to mirror this effect, our game changes both the radius and the magnitude of the magnetic fields when the player

utilizes the drag & drop mechanic of the fields. Standard, non-electromagnets also exist in the game. They can, however, be manipulated as opposed to reality, where they vary depending on size and composition. Without this feature, the game would have a hole in its gameplay, that we felt had to be filled. This small lapse from reality should not have a major effect on the primary objective of the game, namely to teach the core concepts of attraction and repulsion.

Two ways of manipulating the magnetic fields were implemented, sliders and drag & drop touches. The sliders were dropped early on, in favor of the much more intuitive and visually pleasing drag & drop mechanic. The click and drag was inspired by touch screens, and works by clicking with the mouse within the field of a magnet, and holding it down while dragging the field in or out to manipulate its size and magnitude. Having a user interface that natively supports touch screen devices, combined with a framework that works on multiple platforms, gives us a broad potential audience for the game.

Limits were added to control the maximum and minimum size and magnitude of a field. These limits can be specified in the level editor if the default values are not wanted. The rate of magnitude change is fairly small by default, but can also be specified in the level editor. Tracking of which magnet is the closest to the mouse pointer at any given time has been implemented to compensate for overlapping fields. This involves a couple of steps, first, calculating the distance to the mouse in the same way as the distance to any other entity is calculated. Then, the distance is compared to the field radius and to the magnet currently closest to the mouse. If it is both within the field, and closest to the magnet, its circumference circle changes to a deep blue color. This effect can be observed in figure 41.

One aspect of magnetism that we would've liked to cover more closely is polarity. If one overlooks the springboard entity, there is not a lot of functionality in the game centered around polarity and its powerful effects on magnets.

8.5 Exploring other types of magnets and magnetic fields

The game currently supports two different kinds of magnets, although it could be argued it only supports electromagnets. In reality there exists magnets of many types and shapes [9]. With more time it would be interesting to add in more of these magnets, and explore their own special mechanics and workings. Potential candidates may be superconductors, ferrofluids, or something as simple as a horseshoe magnet. All of these have something in common. They require a more rigorous framework for simulating magnetism. It would be necessary to simulate the fields more accurately, so that the two ends of a horseshoe magnet affect each other accurately. For superconductors and ferrofluids this is even more the case, as they start to delve into highly advanced subjects, and would be considered out of scope for this project. A second game, or perhaps an expansion or a bundle of bonus levels could be used to include some of these concepts.

9 Framework

The framework is created to ease the production of future games. The initial idea was that a community site for educational games could be established. The project has evolved around that and the thought was that if one game is created, a framework could be extracted from that to shorten the time spent developing future games. Because of ImpactJS it is quite easy to extract the parts that other games might benefit from.

9.1 Development tools

The framework contains two prominent development tools. ImpactJS and Box2D. Impact JS is a javascript and HTML5 engine that allows plugin-free canvas applications in all browser that support the HTML5 canvas. Box2D is a physics engine that is available in many programming languages, but in our particular case we use version 2.0.2 linked with Impact JS. Box2D was added to simulate accurate physics without having to spend weeks reinventing the wheel.

If any third party were to utilize the framework, it would be required of them to purchase a new licence for ImpactJS. This might discourage others from using the framework, but considering that most other plug-ins and engines require licenses as well this is acceptable. Programs like Unity costs a lot per license and all members of the development team and all platforms require a licence each. In ImpactJS it is one licence per programmer and the licence covers all platforms.

9.2 Weltmeister

Weltmeister is the level editor that comes with ImpactJS. It has allowed us to rapidly create, and edit levels. This allows us to easily test out our idea, and allows for a high degree of flexibility. Variables and attributes may be set through weltmeister, allowing us to easily have variations in objects where wanted. Weltmeister itself is a fairly simple level editor that communicates with ImpactJs through JSON strings.

In the future it may be an idea to open up usage of the level editor for teachers or enthusiasts who want to create their own levels. Creating an open equivalent of weltmeister may be necessary to achieve this due to licensing however. Figure **??** shows an example level. On the right hand side one can observe the way the level editor can interact with, and change the setup of the electromagnet entity selected.

9.3 Button

The button entity was mainly created to display a text field and to send the player to a new level when it is clicked. It works for this purpose, but it should have been changed later in the development to take more than a next level variable to instead take a param-

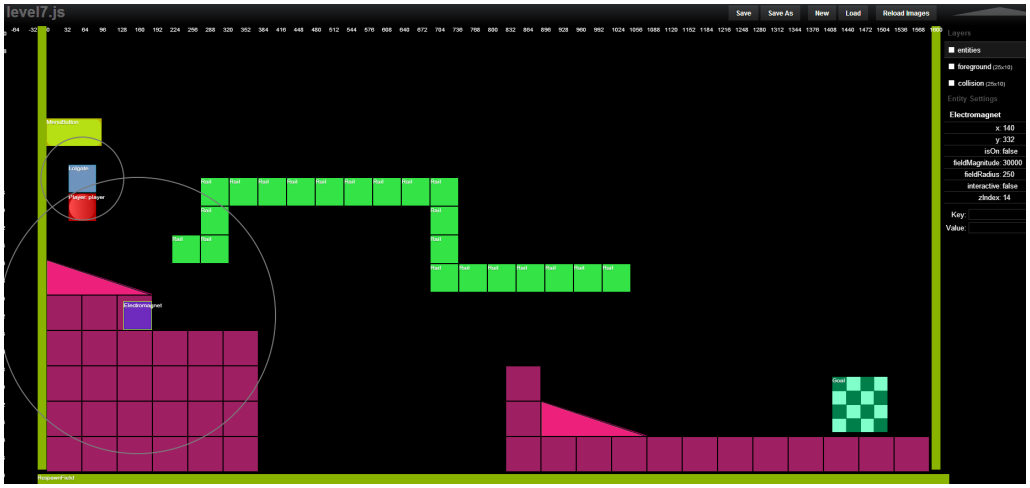


Figure 42: An example of a level in weltmeister. To the right it is shown how one can use the level editor to change the attributes and variables of an entity

eter to a switch case. This would have made this button a lot more flexible to work as any type of button with a clickable action.

9.4 Info screen

This entity consists of a text field and an image. This entity is mostly used to create a small tutorial or help screen to each level. The entity works best on a level alone. Because it have a variable called nextScreen. This will be the name of the level it goes to when clicked on. Then the player can click with the mouse or press enter and it will automaticly send him to the next level. There are two other variables in this entity as well, that will need to be filled out. They are imageName and messageName, these will be the xml path for the image and message that is to be displayed on the info screen.

9.5 Pop-up

The pop-up message is created to help guide the player through a level or by telling the player of new achievements and score he earned. The pop-up message inherits from the info screen. Essentially the pop-up and the info screen is the same, only that the pop-up is a lot smaller and is created to give the user info while the level is ongoing, not a information poster at the start of a level.

9.6 Shield

The shield entity is created to take any kind of message. The message will be sent with the settings when the shield entity is created. The creating entity will also need to tell the main loop that the shield exists. This will cause the main loop to pause all entities except from the shield. When the shield is closed by the player pressing yes or no it will send a boolean back to the creator, which will in return break the pause loop. This will make it possible to get a yes or no answer from the user with a custom message. The creating entity can then act on the data returned from the shield.

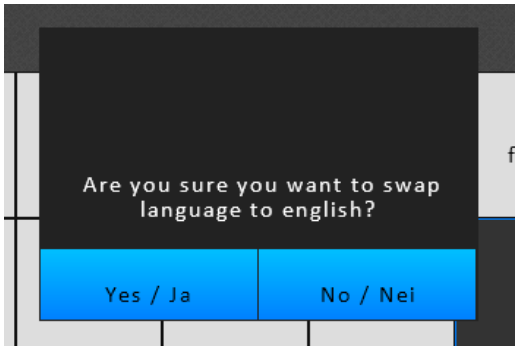


Figure 43: Example of shield entity

```
settings = {creator: this, message: ig.game.xml.loadTextFromXML("language", 0,
"lib/game/xml/strings"+ ig.game.language +".xml")};
this.shield = new EntityShield( 100, 100, settings );
ig.game.paused = true;
ig.game.pauseEntity = this.shield;
```

The entity creating the shield will also need a variable called shieldResponse which will be true or false depending on what the user pressed in the shield.

9.7 Metrics

The metrics system is collecting data from the game. The data collected is the users actions throughout the game. See 10 for more information.

9.8 XML

In this project we utilize a xml based structure to load text into the game. This is inspired by Google Androids [10] approach to save strings. We selected this method because of it seemed like the best choice concerning change of language. It is important for the project to have the option of changing the language. This is because the games that is created or planned for this framework is assumed to be as relevant in Norway as in England or Spain.

The developers can create a stringsXX.xml sheet for each language they want in the game. The XX is the country code for the language used in the xml. Examples of this could be stringsEN.xml or stringsNO.xml.

The contents of the xml documents cannot contain any tabulation whitespace. This will prevent the xml from loading correctly. Any other type of whitespace will be fine.

Example of xml layout:

```
<?xml version="1.0" encoding="UTF-8" ?>
<game>
    <splashScreen>
        <info>Press 'enter' or left mouse button to continue</info>
    </splashScreen>
```

```
    <mainMenu>
        <title> Magnets </title>
    </mainMenu>
    <levelOneInfoScreen>
        <info> Welcome to Level 1! </info>
        <image>media/infoScreen/infoScreenOneImage.png </image>
    </levelOneInfoScreen>
</game>
```

To obtain the data stored in the xml it is necessary to use the loadTextFromXML function.

```
loadTextFromXML: function(label, index, document)
{
var xmlHTTP = new XMLHttpRequest();
    xmlHTTP.open("GET", document, false);
    xmlHTTP.send();
    var xmlDoc = xmlHTTP.responseXML;
    var nodes = xmlDoc.querySelectorAll(label);
    return nodes[index].textContent;
}
```

The function takes three parameters. Label which is the path within the xml document. For instance, if the string wanted in the game is the title for the main menu, the label would be game>mainMenu>title. The second parameter is the index. This is if you have more than one label with the same path. This will usually not be the case, so generally the index will be 0. The last parameter needed is the path and name of the document.

10 Metrics

10.1 Introduction

This Metrics system is based on the concept by [11]. This is used to gather information from the game. The data that could be gathered with this system could be almost anything. Some examples could be players position, time spent on a level, or how many times the game have been completed.

Mr. Prik [12] [13] goes in depth on how to use the metrics system to create a heatmap of the game to see where the player is occupied most of the time. He uses a javascript from Patrick Wied [14] to achieve his goal.

The metrics system could also be used to track every action the player does in the game, and create a complete replay of the session the player had. This could be used to see if any of the user intractable elements is hard to use and should be different, or to see if the players beats the levels in the intended fashion.

The game metrics systems is also fully capable of keeping saves, highscores and other user data. Since the metrics system is more of a database concept than a closed software, it is easy to add features to fit with the game being developed.

10.2 Privacy Policy

No personal information is saved about the user. Registred users choose a username for login purposes, this will give them a user id which is connected to the data gathered. The data gathered about the user is for debugging purposes only. All specific information tracked is a serial number which include the user id, browser, browser version and operating system and the version number of the game.

In this example on line one, we have user id 0, which is a user who have not logged in. The user users Chrome version 26 and windows. The version of the game he is playing is 1.

			rowID	serialNumber	metricMD5	openTime	closeTime
☑	Edit	Delete	98	0Chrome26Windows1	8649b0ba614aabfc2c70c6a6bf2e8a65	2013-04-30 16:16:50	2013-04-30 16:17:02
☐	Edit	Delete	99	0Opera12.15Windows1	8bfcf7c658cfe869b4d18c52d42407f7	2013-04-30 16:17:02	0000-00-00 00:00:00
☐	Edit	Delete	100	0Chrome26Windows1	603986108fb7ab9d3d7dd1d1056c3f77	2013-04-30 16:17:04	2013-04-30 16:17:28
☐	Edit	Delete	101	0Chrome26Windows1	18ee261e867dcaf5c5c392342417a077	2013-04-30 16:17:29	2013-04-30 16:17:57

Figure 44: Example of table entries in the metrics database

The data gathered will not be released to the public, it is collected solely for the purpose of gathering data to improve the quality of the game. Any registered user can at any

time use the contact form on the page to request a deletion of the data gathered from his user id. Anonymous user can not do this, because it would be impossible to track the data back to them.

10.3 Current game metrics uses

For this project the game metrics is used to ensure the quality of the game. To make sure there are no bugs, user interface elements or game mechanics that are too hard for the player.

At this moment the game only track when the player opens and closes the game. What level the player starts and how long it takes to finish the level. Lastly the metric system is tracking what language the player uses and when the player unlocks a new level.

Right now this data could be used to check if the levels are too hard depending on the time spent on the level. It could also be used to track if the majority of the users use norwegian or english as their primary language and set that as the default language.

The system could easily be extended to take in a lot more data and be used to a lot more. We have not experimented a lot with this, because of lack of testers for our game. The system is implemented, the only difference will be to send more events to the database. It will also be needed to create some scripts to read the information gathered in the database. To gain some visualization of the information gathered.

10.4 Future possibilities

This section will delve deeper into the subject of expanding the metrics system for a wider and more extensive use.

10.4.1 Replays

A full replay of a players play session could be relevant to see exactly what the player did to make the game crash, or to see if a player with a unrealistic fast playthrough managed to exploit the game in any way.

This could be accomplished by logging every movement and action the player does. This could be done with the logging system already implemented into the game, but that would be highly inefficient. Since the logging is done one event at the time now, it should instead save the events in bulks and send more than one at a time. So not to put too much pressure on the web server. This would decrease the amount of times needed to access the server and decrease the load on both sides.

To be able to read the events logged to the game metrics a script would have to be written to read the data and send it to a modified client. The modified client would read data sent by the script instead of a user and then replay and emulate the session played by the user. This would create review that the developers could watch too increase the

quality of the game.

Starcraft II [15] is one of many games that utilizes metrics to create replays. Here it is usable by the user as well. This is done so they can watch the game over again to see how they could counter an enemy, or to improve their early game to get higher resource income later in the game. This was not a high priority in this game, but for future games created in this framework this could be more relevant. Both for the player, and for the developers.

10.4.2 Player press reset

To track when the players hit the reset button could help the playability of the game tremendously. This would show the developers where the players get stuck the most and a reset it needed. The developers could either remove the part that the player gets stuck in, tweak it for improvement or if the player spends more that a certain amount of time close to the area people have restarted the most. It will flash the reset button or send a pop-up message and tell them that they are stuck.

10.4.3 Heat maps

A heatmap is a nice way to visualize where the player is occupied most of the time in a game. This is mostly used in first person shooters to see where most players is occupied throughout the game and where they usually die [16]. This data could be used by both developers and players to increase their knowledge of player patterns in the game.

Developers could use this data to improve the quality of the game. For example in a first person shooters this can be done by removing big bottlenecks on some maps, or maybe to create bigger clusters of players to increase the action of the game. It could be used to improve the bots in a game. By seeing where most players move, they could use the data to increase the chances of a bot flanking or outmaneuvering a player. They could also see where players usually kill bots and with which weapon, and with that either counter the weapons of the player or walking another more safe path.

The players might use the heat map to increase their understanding of player movement for a map. Much the same way developers use the data to improve the bots, the players could learn from the player pattern where the most efficient way to walk is. One example could be in a free for all deathmatch, the player could use an earlier heat map to see where most players died and use that data, to eliminate opponents there. Another example could be to use the data to see the safes route in capture the flag, to increase the chances of not encountering an enemy.

10.4.4 Most played maps

We could use the metrics system to check which levels most players play. This could gain some inconsistent data regarding that players have to start at level one and work their way through. What could be track instead is which levels the players comes back to and play again. This could help the design of more levels, to know what the players like and

not.

10.4.5 Quit rate on levels

The metrics system could be used to see which levels most players seem to close the level at. This could either mean the level is too hard, or is just so confusing that the player won't bother. This might help the developers rearrange the order of the levels to push the harder levels back or to remove too hard levels completely.

10.4.6 Comparison on first and second playthrough of a level

This game is mainly created for educational purposes. We could use the metrics system to see a players understanding of the game mechanics. For example the first time the player reaches the level with the rail taking him over the gap, it might take him a few tries. Later the time taking him to complete the level should decrease.

If a level take the same amount of time the second and the first, this might indicate that the level is too easy. This would not be a problem with the shorter levels with only a few simple elements employed, but the later levels that is supposed to be more complicated should be faster to complete the second time as long as the player understood the puzzle the first time and was able to gain advantage from his earlier experience.

10.4.7 Highscores

Right now, the metrics system does not have a fully operational highscore system. It will be fairly easy to add in. The tables for the database is already there, and all that is needed is to create a cron script on the web server. At a scheduled interval it would wipe the daily, weekly and monthly highscore tables. When a player then get a new score, a php script will check if the score he got is higher than the ones already in the highscore tables. If the score he got is higher than the lowest in the table he will update that and sort the tables after score on viewing.

10.5 Implementation

The implementation of the game metrics is done the same way explained by Yann Prik [12]. The database is exactly the same. The only difference is how we submitted the data. In our project we contact the database through an ajax post request to a php script. The post requests contains the data that the metrics system takes as parameters in the database tables. The php script will the connect to the database and insert the new data, if the post request contains the necessary data.

For further implementation of the system, which will be to create more scripts to read the entries in the database and convert it into data that will be usable for the developers. One way to create usable data could be to follow Yann Prik and implement Patrick Wieds heat map code. Other ways could be to create tables or graphs which select similar data and display it in a sensible way.

Example of code

```
startMetricSession: function()
{
this.metricStart = ig.game.getTime();
this.serialNumber =  userId + BrowserDetect.browser + BrowserDetect.version +
BrowserDetect.OS + this.version;
    this.metricMD5 = CryptoJS.MD5(this.serialNumber + this.metricStart).toString();
    var request = $.ajax({
    type: 'POST',
    url: "pages/metricsession.php",
    data:
{
serialNumber: this.serialNumber,
    metricMD5: this.metricMD5
    },
async:true
});
},
```

This will detect the time of when the metric session starts, as well as the user id, browser and version, operating system and the version of the game. Then it sends a post request to the php script called metricsession.php. The request will contain the serialNumber and the md5 of the metrics. This will run asynchronized in the background and will never bother the user in any way.

```
<?php
if(isset($_POST['metricMD5']) && isset($_POST['serialNumber']))
{
include('connect.php');
$sth = $db->prepare(" INSERT INTO metricsession( serialNumber, metricMD5)
VALUES (?, ?)");
     $sth->execute(array($_POST['serialNumber'],$_POST['metricMD5']));
     $sth->closeCursor();
    }
?>
```

This is the php script that will insert the metrics session start into the database. It utilizes a PDO connection to connect to the database, and then it prepare a sql statement. Lastly it runs the execute with an array corresponding with the ? in the prepare. This will stop sql injections. Then the connection to the database is closed.

11 Multi platform and browser support

By using HTML5 and Javascript, availability is ensured across multiple operating systems and browsers. Because of ImpactJS marvelous support in HTML5 the game is supported in the top four most used browsers on Microsoft Windows [17].

11.1 Operating systems

The game works on most browsers supporting HTML5. Because of problems in obtaining all operating systems with all HTML5 supported browsers some testing have been done on browsershots.org, but the data gathered for a free user is not very descriptive.

11.2 Mobile

The game runs on all android based mobile phones with an internet browser supporting HTML5. The only problem is that the game canvas exceeds the density on the phone. On tablets running on the Android OS the game runs without problems as long as the resolution on the tablet exceeds the resolution of the game. If the player chooses to play on a device with a screen with lower resolution the users touch input is off by more and more pixels depending on how far down the user have scrolled.

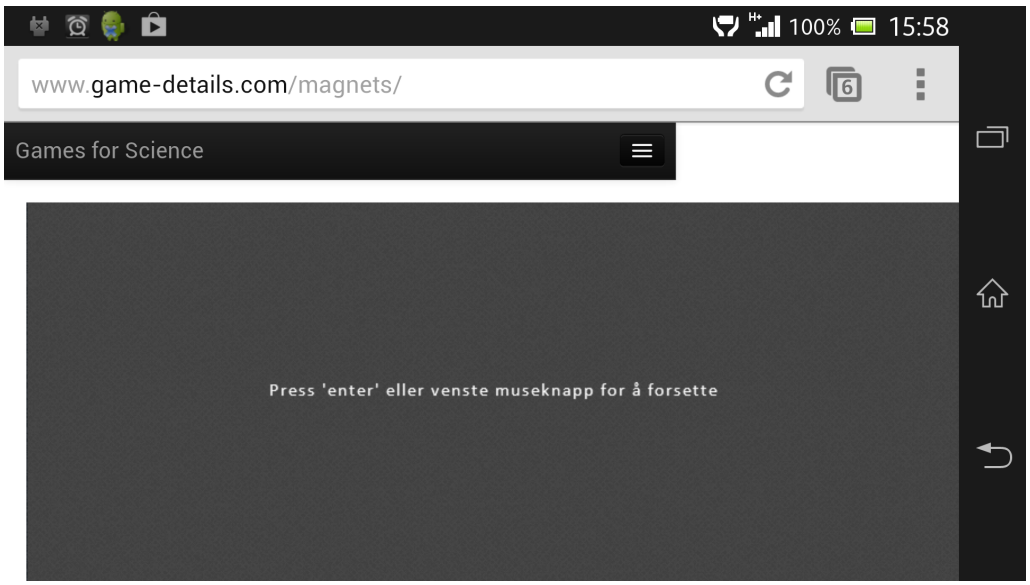


Figure 45: Example of game resolution on a Sony Xperia Z

The resolution part can be fixed, but it was not a huge priority in this project [18].

The game does not work in iOS devices to this date. This can be fixed through Ejecta [19] [20], which have a few prerequisites for the game to work correctly. This have not been prioritized throughout the project, but can be fixed without a huge effort from the developers.

11.3 User input

All the user input in the game can be done with mouse clicks only. This implies an easy conversion to tablet and phone support. Right now all user input works on mobile devices, the only issue is the camera movement, which is not optimal. This should be done with swiping and not by hovering the mouse on the edges as it is on desktop computers.

12 Depth Jam

A game jam is the process of sitting down as a group and designing and creating a game over the course of only a few days. There exists several known, large game jams such as:

- Global Game Jam
- Ludum Dare
- PyWeek

Game jams usually involve setting a theme, then designing and creating something that revolves around this theme. Some Game Jams are location specific, while others span the entire globe. The time frame of which a Game Jam lasts also differ from Game Jam to Game Jam. Jams usually last from somewhere around 24 hours, to 120 hours.

A depth jam is another variety of a Game Jam, where instead of creating a game, a group sits down to design and create something new for an already existing project, e.g a new feature. The development team for this project did a depth jam where two main new features were added, springboard and pendulum. Doing a depth jam proved very useful for the project as a whole, because it functioned as a mid-project crunch. The two new features alone were not the only things that were done, because the two new features made use of older code. This forced us to refine, finish and polish many of the older features, such as the electromagnet, the buttons for navigating the menu structures, and camera panning. In addition to this, we also found time to create new levels, and add in smaller features such as switches. To see the full written report from the Depth Jam see appendix E.

As a tool for boosting progress, or catching up in a project a depth jam seems to work very well. It allows its participants to focus solely on the development of these new features, ignoring everything else. It is definitely something the team will recommend, and hopefully recreate in other projects in the feature.

13 Learning Management System Integration

This projects framework is a good candidate for integrating with a learning management system such as Fronter or Moodle. A framework connected to a learning management system can be used to perform graded tests and provide the educators with metric data. Potential data that can be logged are where students are stuck, how long each student spend on solving the tasks, and more. For a detailed explanation of metric logging see chapter 10.

Although it is a good candidate, integrating with an LMS is out of scope for this project. It would require a large chunk of work, especially if aiming to integrate with more than one learning management system. For further development of the framework however, it could be a good idea to explore.

13.1 Possibilities

A game framework integrated into learning management systems can offer an educator a new set of possibilities that were previously unavailable. An educator can hold a lecture, then test the students skills, or have them review by playing a game within the framework. For it to function as a test the metrics system will have to be rewritten to allow for specific events to be logged at the educators request. It is necessary for the system to be able to look for specific answers or solution for problems and puzzles, allowing grading or personal review for users. To do this it is necessary to sit down with educators to deem what data is relevant, and how this can be used in a sensible way.

For it to work as a test it would be necessary to implement a time-lock system that, so that the educators may have the test open for only a certain amount of time, and from a set time and date. A limit on the amount of tries may also be sought after, so that the user can only do the test once.

13.2 Tracking learning

As a game framework with the intent to help science education, it is important to have a way to track its effectiveness. A learning management system may be used to do this. Following the scientific method when testing it is important to split students into several test groups. Each group should do a benchmark test at the start, then learn the subject in mind through different sources and methods. One group may learn through playing a game, as well as using the standard lecture, another group may just play the game. It is important to have a group that just has standard lectures as a control group.

Tracking the effectiveness of learning can prove essential for the life of a project such as this. If the game doesn't do a good job at teaching the subject, it may not be worth continuing development on that specific game. One experiment for testing the effectiveness is however not enough to disprove the process itself. It is important to take the feedback given, and do other tests to see if it yields the same results.

14 Vitensenteret

Vitensenteret is a science center located in Gjøvik. It is devoted to teaching children in all the stages of elementary school and middle school about science. They have school classes visiting for educational activities during the week. The center is open for visitors of all ages on the weekends, where children and adults can join in on activities and explore the center together.

The activities hosted by Vitensenteret revolve around several themes, such as chemistry, physics, anatomy, astronomy, mathematics and the climate. The activities are created to engage and inspire the children to learn, often involving role-playing by the staff.

14.1 Testing group

Vitensenteret is an obvious location to test an educational game. They already have classes from the target audience coming to their activities. They are also looking for new innovative ways to inspire children to learn about science. A game revolving around magnets is a good choice regarding this. This may open up an opportunity for Vitensenteret to create a real life version of the game, so that the children can interact with the curriculum and the primary concepts of magnetism on several fronts, linking the virtual world with reality.

Vitensenteret introduces parts of the events to the children through their teacher and school before they come to attend the activities hosted at the center. A game could prove to be a useful asset in this part. An example usage is then to introduce the concepts of magnetism through the game at the school, then see the real life model when they visit.

A game could also be used to draw a bigger crowd to Vitensenteret through in-game achievements. With a good and established reward system in the game, Vitensenteret could utilize QR-codes to give them achievements for attending major events and activities.

14.2 Overdue

Initially the alpha version of the game was to be tested at Vitensenteret in early April. This was somewhat postponed because of a delay in the development phase. When the game was ready to be tested, communications had slowed down and the project approached its deadline so fast the testing got delayed until after the delivery is due.

The game will still hopefully be tested at Vitensenteret in the end of May. This is after the project is due, but will still be an interesting experience concerning the educational value of the game, and may be mentioned during the verbal presentation.

Figure 46: An event at the Vitensenter, during their science research days. Photo: vitensenteret. [21]

15 Dragonbox

With the focus of our project being on serious games, we felt the need to contact and interview the creator of a successful educational game.

WeWantToKnow is a French-Norwegian start-up company that produces high quality educational games. It can truly be said to be a successful company who create and sell serious games. They have relaesed two games so far, DragonBox and its successor DragonBox 2.
These are games that teach children aged 5+ about equations and 12+ about algebra. The DragonBox games excel at abstracting away the fact that player is doing maths, and sneakily teaches you core math ideas without the player realizing.

The DragonBox games are fantastic examples of how games for science education can be done successfully. It has received very good reviews, and won several awards.

The first DragonBox game was awarded Best Educational Game at the Fun and Serious Games Festival 2012 and both App of the Year and Children App of the Year at Gulltasten 2012

Plenty of good reviews can also be referred to. (http://wewanttoknow.com/reviews/)

DragonBox is making me reconsider all the times I've called an educational app "innovative." Wired, geekdad

My eight year old son immediately sat down and ran through the first two banks of problems without hesitation. It was amazing. Christopher Wanko, CoolTools

These questions and answers have been rewritten to match the literary style of this report, but content should be semantically equal. The e-mail with the exact wording is attached in [appendix B.2]

How is your developer team composed? How many programmers, designers, artists etc do you work with?

"at the moment 4 developers, 1level designer, one graphic designer. sound and illustrations are consultants."

At the moment, we are four developers. In addition, there is one level designer and one graphics designer. On top of this, consultants have been brought in to create sound and illustrations.

Has development and testing caused the concept to change much from the initial idea?

“development has not changed the concept. Key controls and ideas were defined before coding. but small details and refining have been implemented continuously. we still have ideas that can implemented, meaning we have a very solid backbone.”

Development has not changed the concept. Key controls and ideas were defined before coding. However, small details and refining have been implemented continuously. We still have ideas that can be implemented, meaning we have a very solid backbone.

Are you actively trying to hide the math from the player to trick him into doing math, or is that a kind of byproduct of abstracting the numbers and letters used in maths?

“hiding math is natural since math the way is taught is suboptimal to represent math ideas to beginners. We have started two new games, and u can t see the math at first.”

Hiding the math is a natural approach, because new ideas are traditionally presented in a suboptimal way.

How have you been marketing the game?

“mkting thru pr, but we will change that now with virality in the game.”

So far, we have been marketing through PR, but we will change that now, with the current virality of the game.

Who is the target costumer? Kids usually don’t have money to pay for apps. Is this a "homework app"?

“target is parents”

Even though the game itself is targeted to children, the parent will usually have to pay for the app and present it to the child.

Do you have any advice for students trying to make apps in the same category?

"general advice be serious and passionate about what u re doing.”

General advice: be serious and passionate about what you are doing.

16 Further Development

Upon reaching the deadline for the this project, there are still several things the developers feel the game needs, or the framework needs. To make the solutions profitable the game and framework will need further development in the form of those steps.

16.1 Complete test-phase

If development was to continue beyond the deadline, the first priority would be to polish and improve the game in its current state. To do this it is paramount that the game goes through a rigorous test phase.

Experiments are needed to find out the educational value of the game. By having children of the target group testing the game, the statistics for the games entertainment and educational values will emerge. Equally important for the development of the game is that educators also test the game, and give feedback on the content. It is necessary to see if the content is appropriate for the educational purpose that is expected of the game, and to find out if it is possible to use the game as an educational tool.

16.2 Refining what we have

On the basis of the information gathered from the test phase, refining of the current game can be initiated on a grand scale.

Removing placeholders

This in essence means the removal of any remaining placeholders from endScreens and infoScreens. The infoScreens should have screenshots of game mechanics they aim to explain. The strings in the XML-files would be rewritten to be more coherent, and generally be more helpful and better at conveying the information that is required to tackling the puzzle at hand.

Updating of outsourced assets

If any assets have been disliked, or deemed unfitting or obstructing for the gameplay or visual feel of the game, it may also be necessary to have some of the assets, graphical or auditory, redone.

16.3 Balancing

The game in the state it was turned in includes quite a few bugs, mostly related to the difficulties of each level, and the flow throughout the levels and game. There are some difficulty spikes in the mid-levels that should be redesigned, or perhaps moved to a later stage of the game. Some levels are not even possible to complete in their current state without a large amount of luck or previous experience.

16.4 New features

After refinement of the original status of the game, it can go back into development phase for new features.

Score

A fun way to challenge the player is by using a score after each level that represent how good the player is at the game and at that level, to encourage replaying the level more than one time, in the pursuit to gain a better or higher score.
If the game have completed the test phase then there is either proof of the educational effect from playing the game, or it has been redesigned or dropped. Using score to encourage replaying the levels will also increase the exposure of the game to the players. Up to a certain point more exposure will equal a higher understanding of the concepts the game is trying to teach.

16.5 Achievements

An effective way to reward the player in a game is by prizes or trophies, and one of the more effective digital prize-systems is the achievement system adapted by many games. Achievements are earned by mastering the games, or doing exceptionally good events.
This game will mostly use achievements as a way for the children and the educator to track the progress in the game. Achievements are rewarded after completing each level, and the achievements will be user specific. This makes it so the educator can see what child completed, which level, this can be used for checking homework.

Special events and feats of strength within the game may also award achievements for more individuality. An idea from Vitensenteret to use QR-codes to award achievements when checking in at a magnetism event at their center may be one of these special events.

16.6 Additional content

The next major step would be to create new content for the game.

New levels and mechanics

This entails creating new levels that utilize the current mechanics in new or convoluted ways to increase the difficulty curve of the game. New mechanics will also be introduced as a way to keep the game interesting for players that keep playing the game outside of the educational purpose, or to create new educational value.

Level editor

If we are able to create an open level editor it may encourage the player base to create new content for us, and may also encourage social behaviours by the players sharing the new levels. To make levels the player will need to understand the core concepts that the game is teaching, so the level editor can also work as part of the education.

16.7 Expanding the framework

By expanding the framework new features can be added. One idea here is to create a portal for games, specific to be used for educational purposes, that users will keep using and come back to.

The game portal

The main use of the framework is the encourage making more games by other developers or us. The framework will work as the base that binds the games together. To do this the best way is to create a web portal that will link together all the different educational games with a common user.

Social features

Let the user add friends to a friends list. This will make it easier for users to share achievements and accomplishments. It will also help users know if their friends are using the platform.

Chat functions

Add a chat system internal on the portal, that will let users join channels of their own creation or those that are made default by the portal. Things like class chat rooms, country rooms, game specific rooms, friend rooms, and many more that the users expect from a multi game platform.

More games

When the development of the Magnets game is done and it is completed, we wanted to start creating a second game using the framework. The frontrunner for our next game is:

War of the atoms!

The game will use units with atom-sounding names (Hydrogen soldier, Oxygen pikeman, Water cannon), that the player can use to defend against non-player characters or other players.

The player can acquire better units by using recipes that will mimic real life molecule combinations. Two hydrogen soldiers and an oxygen pikeman may be used to create a water cannon. This means that the core mechanic of the game is the upgrade system that will teach the player about the way different molecules are built up by proxy of them remembering how they build stronger defences. The same way the game will encourage the player to try out combinations that might not be successful or safe.

By drilling the player into repeating and remembering combinations, they will be learning how molecules are built up by atoms, and how they can combine to make new elements or make existing reactions. If the game becomes fun the player may have an easier time remembering the combinations than by repetition.

17 Conclusion

This project has been the biggest project we have worked on so far. We have gained a lot of experience by managing and developing this game and framework. The research around serious games gave us a whole new aspect to games and how children learn in general. The experience with gathering a team to create the necessary assets we could not create ourselves was immensely helpful. This gave us a better view into how a real life project would be managed, additionally it gave us new contacts within the game development field.

To create a complete game ready to be fully tested by the target audience granted us more knowledge of how a full development phase would play out. We would like to have started the testing phase at Vitensenteret, though sadly we never got that far. It would have been nice to get some real test data from children in our target audience to confirm the educational value of the game.

The extraction of the framework from our game never happened. This would not be hard, but would take a considerable amount of time. It could have proven useful and someone else might have been able to continue the development of additional games and extend the framework in the future. If someone else were to continue the project this would require a new ImpactJS licence key.

When talking to other developers and educators in the field we gathered huge amounts of important information, though some of our contacts could have proved more useful, like DragonBox.

Bibliography

[1] Justis & beredskapsdepartementet. 2000. Lov om behandling av personopplysninger (personopplysningsloven). http://www.lovdata.no/all/hl-20000414-031.html.

[2] Spitzer, M. 2006. Brain research and learning over the life cycle. *Personalising Education*, 47.

[3] Papert, S. 1998. Does easy do it? games and learning. *Game Developer*.

[4] Faraday, M. 1832. Experimental researches in electricity. *Philosophical Transactions of the Royal Society of London*, 122, pp. 125–162.

[5] Becker, J. F. Dc circuits. http://www.physics.sjsu.edu/becker/physics51/dc_circuits.htm.

[6] Maxwell, J. 1873. A treatise on electricity and magnetism. *Clarendon Press series*.

[7] Price, C. B. 2008. The usability of a commercial game physics engine to develop physics educational materials: An investigation. *Simulation & Gaming*, 39(3), 319–337.

[8] Feynman, R. P., Leighton, R. B., & Sands, M. 1979. *The Feynman lectures on physics, vol. 2: Mainly electromagnetism and matter*. Addison-Wesley.

[9] Kahn, O. 1999. Condensed-matter physics: The magnetic turnabout. *Nature*, 399(6731), 21–22.

[10] Android, G. String resources. http://developer.android.com/guide/topics/resources/string-resource.html.

[11] McCallum, S. & Mackie, J. 2013. Webtics: A web based telemetry and metrics system for small and medium games. In *Game Analytics*, Seif El-Nasr, M., Drachen, A., & Canossa, A., eds, 169–193. Springer London.

[12] Prik, Y. 2011. Game metrics.

[13] Prik, Y. 2011. Game metrics source code. https://docs.google.com/file/d/0Bw4odICvlZY8MWI0Y2ZmOTQtZGFmNS00ZmFmLTlmOTMtOGQxYTllZDU2Yjlm/edit.

[14] Wied, P. 2011-2013. Heat map js. http://www.patrick-wied.at/static/heatmapjs/.

[15] Entertainment, B. 5 2013. Replay file enhancements in patch 2.0.8. http://eu.battle.net/sc2/en/blog/8191668/Replay_File_Enhancements_in_Patch_208_-08_05_2013.

[16] GameAnalytics. 2 2013. Heatmapping. http://blog.gameanalytics.com/blog/heatmapping.html.

[17] W3Schools. Browser stats. http://www.w3schools.com/browsers/browsers_stats.asp.

[18] Szablewski, D. Impact on mobile platforms. http://impactjs.com/documentation/impact-on-mobile-platforms/.

[19] Szablewski, D. Ejecta. http://impactjs.com/ejecta/.

[20] Szablewski, D. ios overview. http://impactjs.com/documentation/ios/overview/.

[21] Vitensenteret. Photograph. http://www.vitensenteret.no/images/besokoss/glimtfravitensenteret/Forskningsdagene.jpg.

A Meeting logs

Meeting Logs		
Meeting date	**Agenda**	**Total man hours**
11/01-13	First bachelor planning meeting. Meet the group. Learn the Bachelor thesis. Talk about how to Bachelor.	20mH
14/01-13	Bachelor planning meeting. Group name. Meet the partners. What is the pure facts of the thesis. Test group. Privacy policy. Group rules. Guidelines.	20mH
16/01-13	Bachelor planning. Group rules decided. Composed letters for artists. Questions for Vitensenteret.	16mH
17/01-13	Bachelor planning. Code guidelines done. Group routines done. Tried new ways to find composers and artists. Group rules updated. Folder structure discussed	16mH
18/01-13	Bachelor planning. The legal document done. updated the contract with meeting recording device. Made a vote delay veto for Simon. Scheduled a meeting with Vitensenteret. All signed the Documents.	20mH
21/01-13	Rapport meeting.	20mH
22/01-13	Rapport. Games for science presentation.	14mH
23/01-13	Rapport.	16mH
24/01-13	Rapport Done. Printed, signed, and delivered all documents. Wrote pre-documents . Vitensenteret questions final.	8mH
25/01-13	First vitensenteret meeting. Vitensenteret as our user group.	16mH
28/01-13	Pre-rapport hand-in. Contacted partners. Contacted school.	16mH
29/01-13	Serious games.	2mH
30/01-13	Game Planning. First brainstorm of game ideas.	16mH

01/02-13	Game idea brainstorm. Prism idea. Space collector.	3mH
04/02-13	Vitenskap senter visit. Theme of game brainstorm. Crime-lab act. Gravity, momentum transfer, and car race.	32mH
05/02-13	Meeting with Rune. Serious gaming lecture. Game idea discussion. Made a list of games that correspond to the curriculum. Voted on ideas.	17mH
06/02-13	Made descriptions for each game idea.	16mH
07/02-13	Bachelor group homepage. Design the framework homepage.	9mH
08/02-13	Meeting with Simon. Discussed every game idea. Decided on Magnets.	4mH
11/02-13	Design document creation. Framework site design. Design document for Magnets.	15mH
12/02-13	Keep working on design documents.	12mH
13/02-13	Keep working on design documents	12mH
14/02-13	Started the project. Made the folder structure. Set up the repository.	4mH
15/02-13	Framework discussion. Updated JIRA.	6mH
20/02-13	Started on game Level design.	9mH
21/02-13	More game level design. Made a one pager for the team of designers.	6mH
24/02-13	Keep designing game levels.	6mH
05/03-13	Back to game level designing. Tutorial levels done.	8mH
07/03-13	More game level design. Re-wrote most of the design document.	8mH
09/04-13	First meeting in a while. Specific tasks assigned.	22mH
15/04-13	First day of Game Jam. Short meeting. Pure development.	64mH
16/04-13	Game Jam. Non stop programming.	60mH
25/04-13	Development session.	48mH
29/04-13	More Programming.	40mH
30/04-13	Final rapport planning meeting.	16mH
06/05-13	General bug fixing. Programming. Development.	12mH
08/05-13	New assets. More programming and development.	24mH
/-13	even	mH

B Email Logs

This appendix contains logs of emails throughout the project.

B.1 Aquiring ImpactJS

```
Christian Holt <ymabob@gmail.com>
Jan 29
to dominic.szable.

Hello,

I am 1 out of 4 programmers (1-2 artists may also be involved during the semester,
we're not quite sure yet) doing a bachelor thesis on
Gaming for Science Education,
meaning we're looking to create two games aiming to teach different
science subjects.

The reason for me contacting you is that we are considering using impactJS in
our project, but due to no economic support, we are not able
to afford 4-6 licenses. We are therefore wondering if you would be so kind as
to grant us a good discount, or perhaps allow us to use it
free of charge. Crediting the use of impact will of course be done.

Our plan is to develop these games open source, so it will be available on Github
for everyone when we are finished. (We will need to discuss this
further with our supervisor and employer, but this is currently
the case).


Thanks in advance,
Christian Holt.



Dominic Szablewski <dominic.szablewski@googlemail.com>
Jan 29
to me

Hi Christian,

I'll send you a license key in a minute. You can share this key
within your group and use it for non-commercial purposes.
```

```
If you later want to publish your game commercially (e.g. financed
with ads, or licensed to a gaming site), please consider
buying some licenses :)

When publishing your code, remember that Impact is a commercial product.
You are free to put your code on Github, but please exclude the
code for the engine itself (the /lib/impact and /lib/weltmeister folders).

Cheers
Dominic \\
```

B.2 Dragonbox

B.3 Artist

jb May 9 (6 days ago)

to me

at the moment 4 developers, 1level designer, one graphic designer. sound and illustrations are consultants.
development has not changed the concept. Key controls and ideas were defined before coding. but small details and refining have been implemented continuously. we still have ideas that can implemented, meaning we have a very solid backbone.
hiding math is natural since math the way is taught is suboptimal to represent math ideas to beginners. We have started two new games, and u can t see the math at first.
mkting thru pr, but we will change that now with virality in the game.
target is parents
general advice be serious and passionate about what u re doing.

00 47 91632296
jean-baptiste huynh

Den May 8, 2013 kl. 10:28 PM skrev Christian Stokkebekk <gfzilter@gmail.com>:

...

> Hello again
>
> It was nice talking to you the other day.
>
> Here are the questions that we wanted to ask you:
>
>
>
> How is your your developer team composed? How many programmers, designers, artists etc do you have on your team?
>
>
>
> Has development and testing caused the concept to chang much from the initial idea?
>
>
>
> Are you actively trying to hide the math from the player to trick him into doing math, or is that a kind of byproduct of abstracting the numbers and letters used in maths?
>
>
>
> How have you been marketing the game?
>
>
>
> Who is the target costumer? Kids usually don't have money to pay for apps. Is this a "homework app"?
>
>
>
> Do you have any advice for students trying to make apps in the same category?
>
>
>
> Thank you for you attention!
>
> Christian Stokkebekk

Figure 47: Email 1 from Dragonbox

Jean-Baptiste Huynh Apr 25

to me

jbhkb1

On Thu, Apr 25, 2013 at 3:05 PM, Christian Stokkebekk

...

<gfzilter@gmail.com> wrote:
> Hi
>
> Sure, I can do that. What is you name on skype?
>
> In the meantime, I will check out you new game.
>
>
> On Wed, Apr 24, 2013 at 10:53 PM, Jean-Baptiste Huynh <jb@wewanttoknow.com>
> wrote:
>>
>> hi,
>>
>> thanks for your interest.
>> we have just released our new game, and it s a bit hectic now. Try to
>> skype me middle of next week!
>>
>> Jean
>>
>> On Mon, Apr 22, 2013 at 4:06 PM, Christian Stokkebekk
>> <gfzilter@gmail.com> wrote:
>> > Hello
>> >
>> > I represent a small group of students who are currently writing our
>> > bachelor's degree in Game Programming at Høgskolen i Gjøvik.
>> >
>> > We have decided to do our project in Serious Games, and Games for
>> > Education.
>> > We are writing a browser-based game which seeks to introduce children to
>> > the
>> > broad concepts of magnetism through play and puzzle-solving.
>> >
>> > Our research into Games for Education naturally led to Dragonbox. It
>> > surely
>> > is a great example of how games can be utilized to teach and simplify
>> > traditionally difficult concepts.
>> >
>> > We would like to ask for an interview regarding Dragonbox and it's
>> > develpment, and then discuss that interview in a chapter of our paper.
>> > We
>> > can come to you in Oslo for the interview, or we can do it through
>> > Skype,
>> > whichever you prefer.
>> >
>> >
>> > Hope to hear from you!
>> >
>> > Christian Stokkebekk
>> >
>> >
>> >
>> > PS: We speak both Norwegian and English.
>
>

Figure 48: Email 2 from Dragonbox

Eivind Eriksrud Apr 30

to me

Norwegian > English Translate message Turn off for: Norwegian

Da har jeg lagt deg til. Contact request meldingen er litt subtil i skype.

Den 16:11 29. april 2013 skrev Christian Stokkebekk <gfzilter@gmail.com> følgende:

...

> Jeg har lagt deg til på skype, det hadde vært fint om du godtok. Hadde vært nice å fått inn litt mer grafikk, og det er litt lettere å fortelle hva vi trenger over skype enn over mail.
>
> 2013/4/25 Eivind Eriksrud <erikseiv@gmail.com>
> > Eivind Eriksrud heter jeg på skype. Profilnavnet er vel eivind_tha_king
> >
> > Den 15:02 25. april 2013 skrev Christian Stokkebekk <gfzilter@gmail.com> følgende:
> >
> > > Flott med alt du gjør, men vi trenger noen små justeringer på et par ting. Har du skype, sånn at det går an å kommunisere litt raskere enn over mail?
> > >
> > > 2013/4/22 Eivind Eriksrud <erikseiv@gmail.com>
> > > > Ja fint dere liker det. Skal ordne med plattformer og bakgrunn neste.
> > > >
> > > > Den 15:24 22. april 2013 skrev Christian Stokkebekk <gfzilter@gmail.com> følgende:
> > > >
> > > > > Supert!
> > > > >
> > > > > Det hadde vært aller kulest om det neste du la opp var et bakgrunnsbilde til levelene.
> > > > > Vi tenkte kanskje det kunne se litt ut som en slags blurry bakvegg i en lab eller noe sånt. Hvis det gir mening. Litt viktig at det ikke tar fokus vekk fra puzzlet man driver med da.
> > > > >
> > > > > En annen kul ting hadde vært å fått på plass platformene som ballen ruller på.
> > > > >
> > > > > Skal bli gøy å legge til ordentlig grafikk, plutselig ser spillet nesten bra ut :P
> > > > >
> > > > > On Mon, Apr 22, 2013 at 2:49 PM, Eivind Eriksrud (Google Drive) <erikseiv@gmail.com> wrote:
> > > > > > I've shared an item with you.
> > > > > > Hei, jeg legger opp ting her.
> > > > > > Magnets
> > > > > > Google Drive: create, share, and keep all your stuff in one place. Google

Figure 49: Email from artist

C Hand drawn images

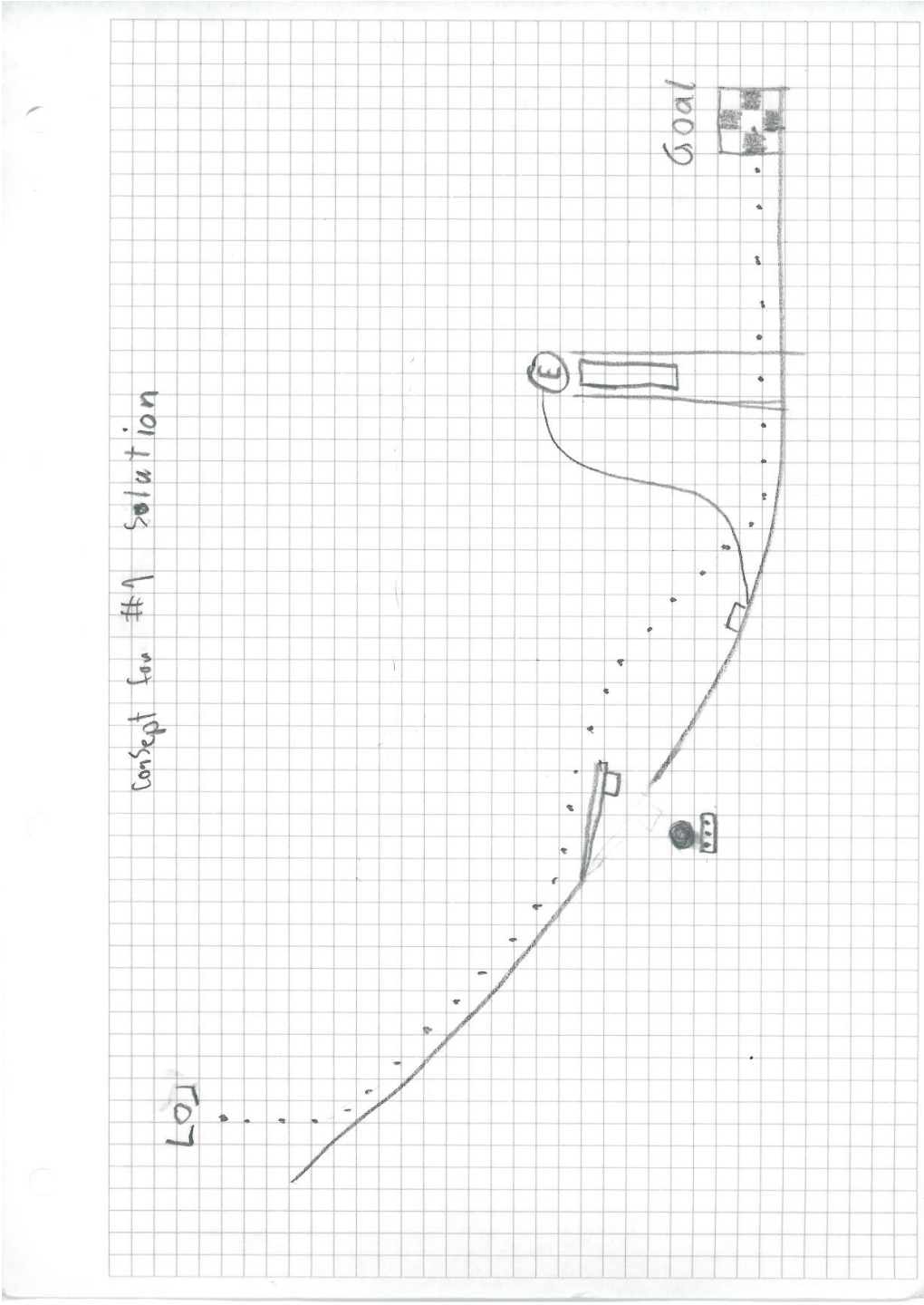


Figure 50: hand drawn image nr1

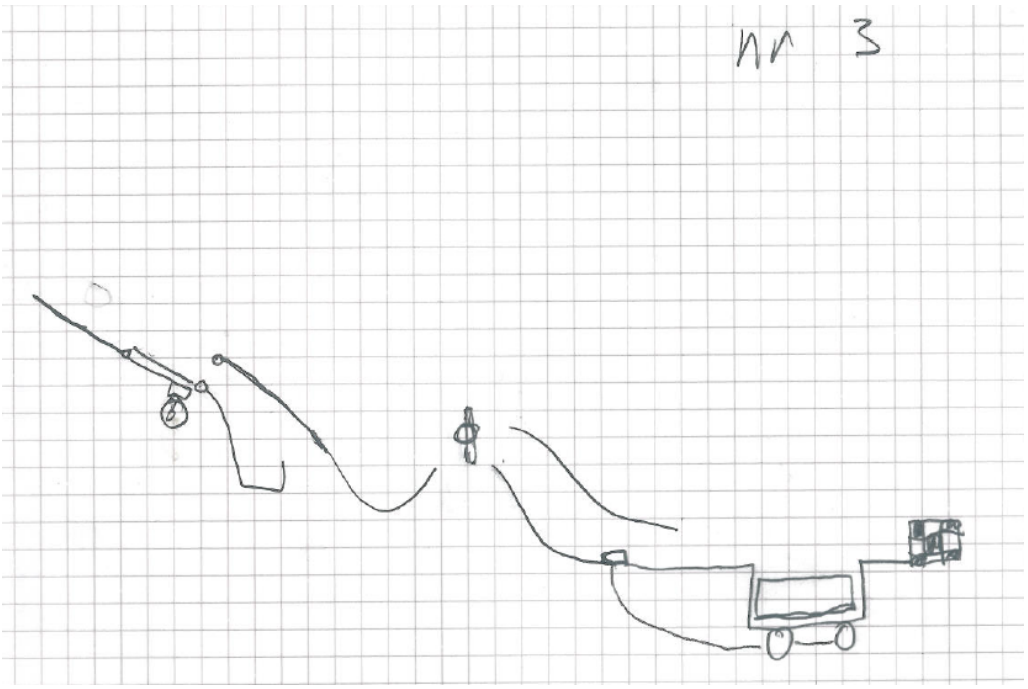


Figure 51: hand drawn image nr2

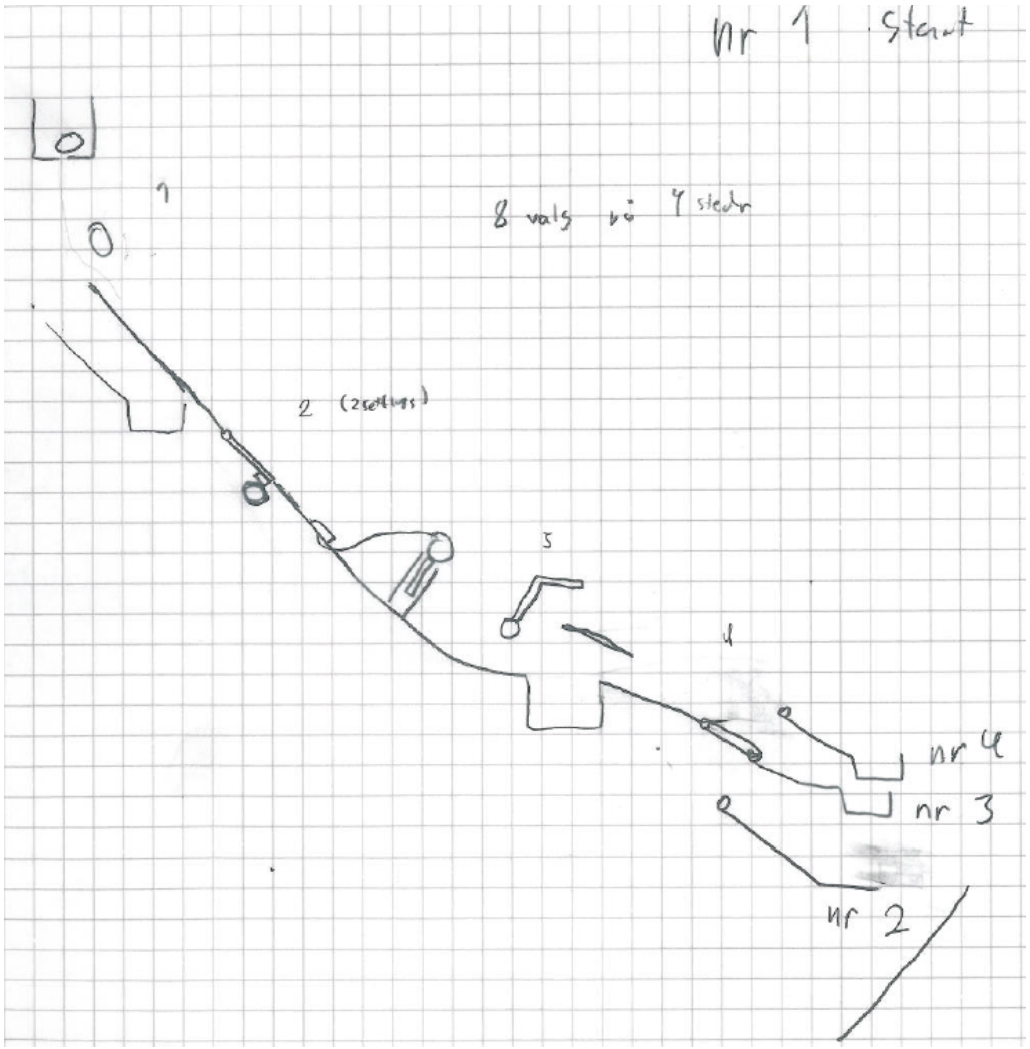


Figure 52: hand drawn image nr3

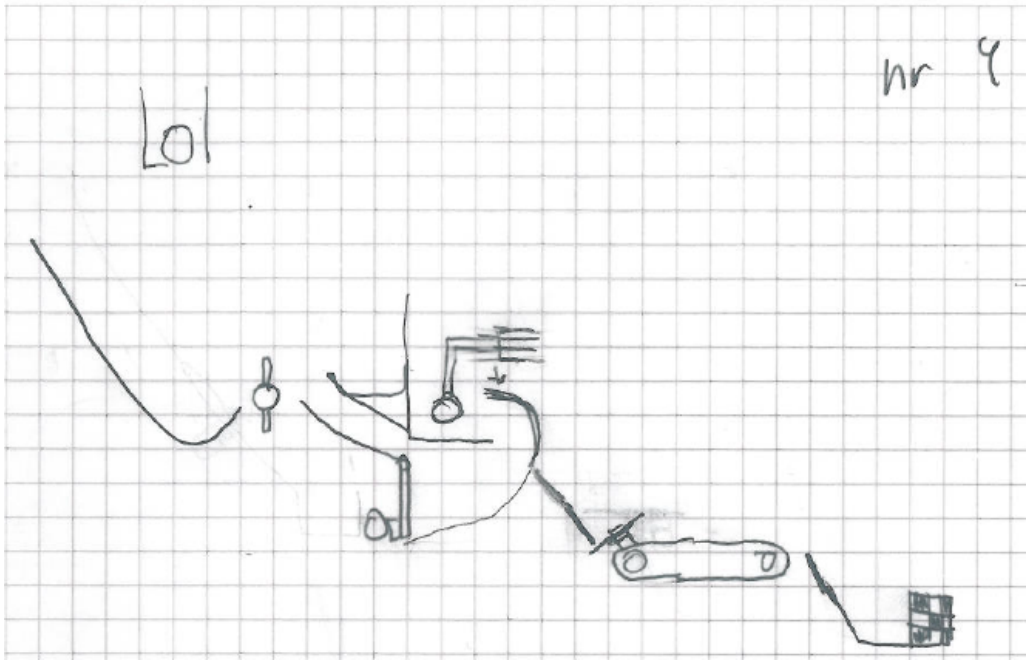


Figure 53: hand drawn image nr4

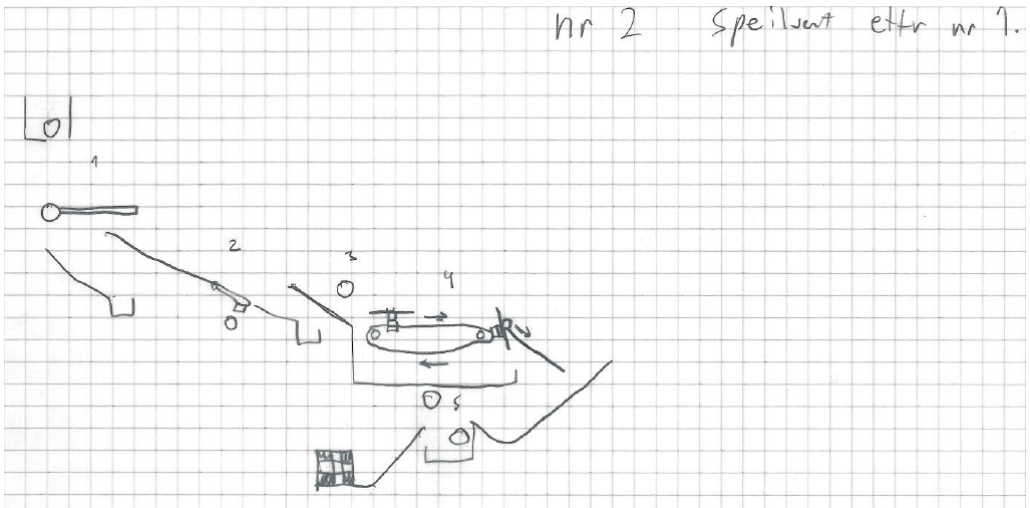


Figure 54: hand drawn image nr5

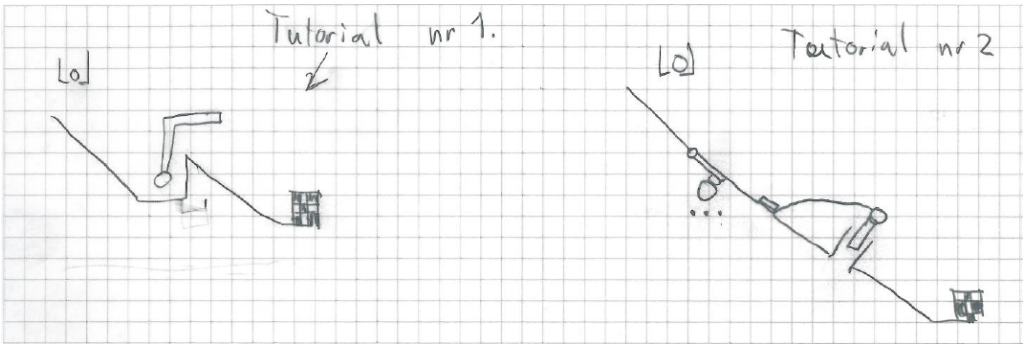


Figure 55: hand drawn image nr6

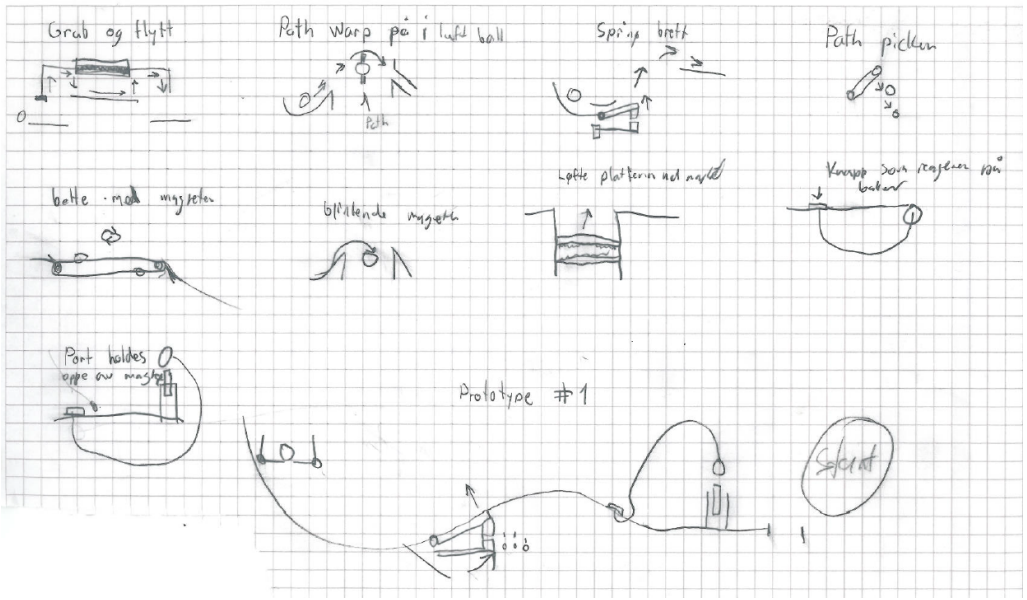


Figure 56: hand drawn image nr7

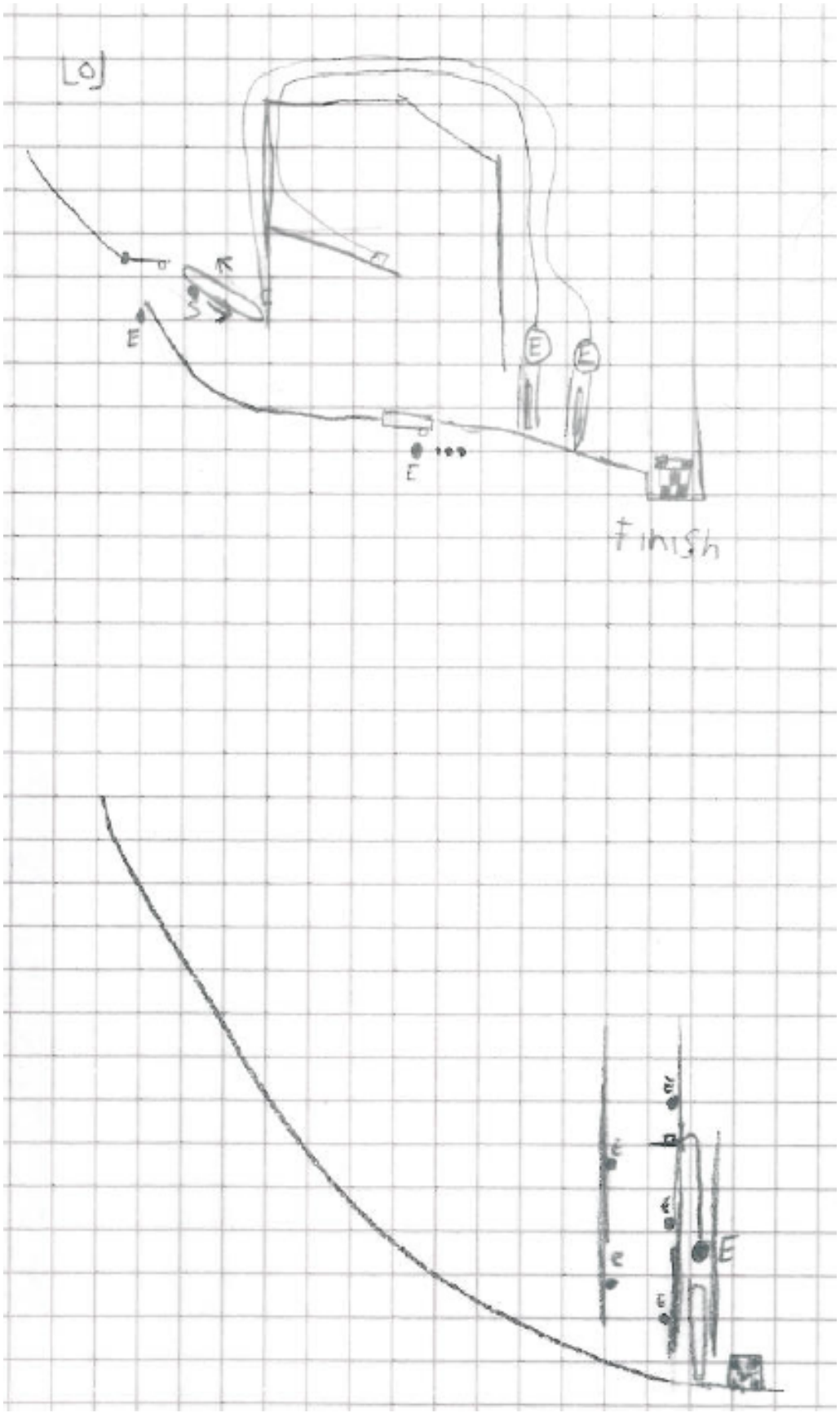


Figure 57: hand drawn image nr8

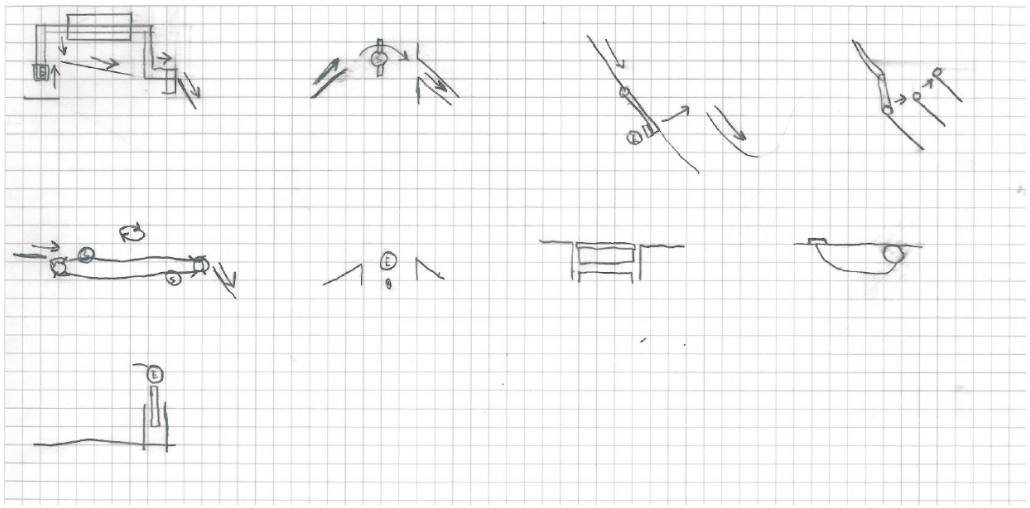


Figure 58: hand drawn image nr9

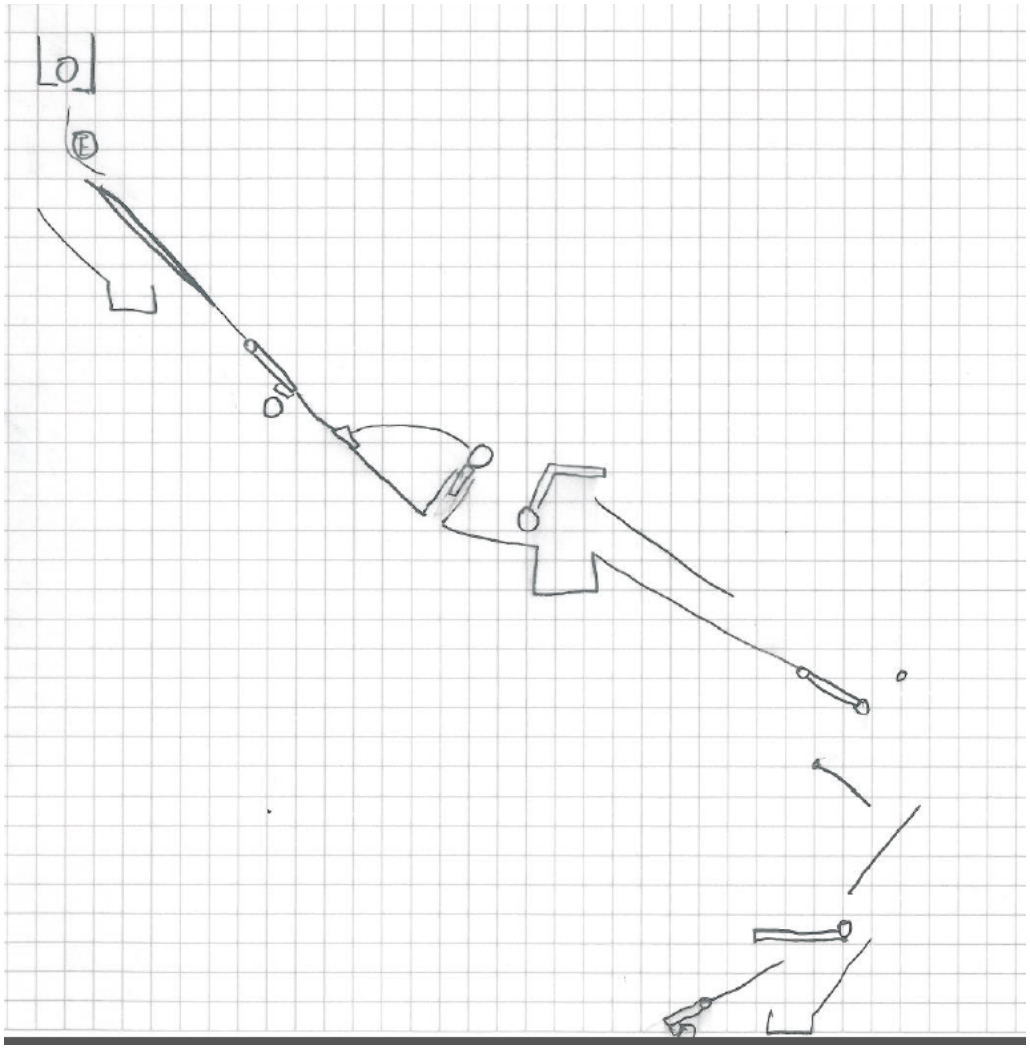


Figure 59: hand drawn image nr10

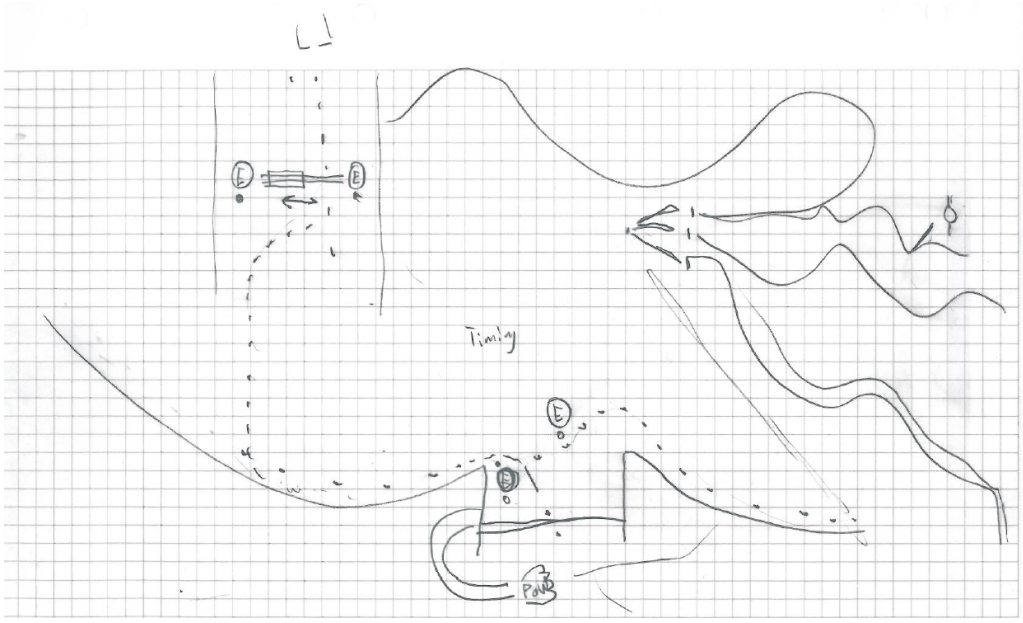


Figure 60: hand drawn image nr11

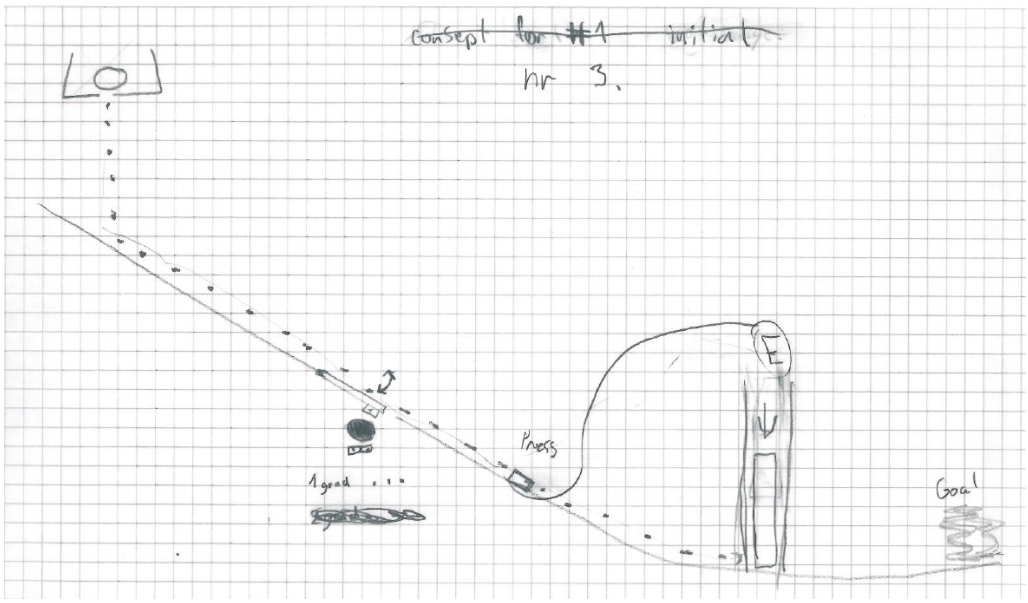


Figure 61: hand drawn image nr12

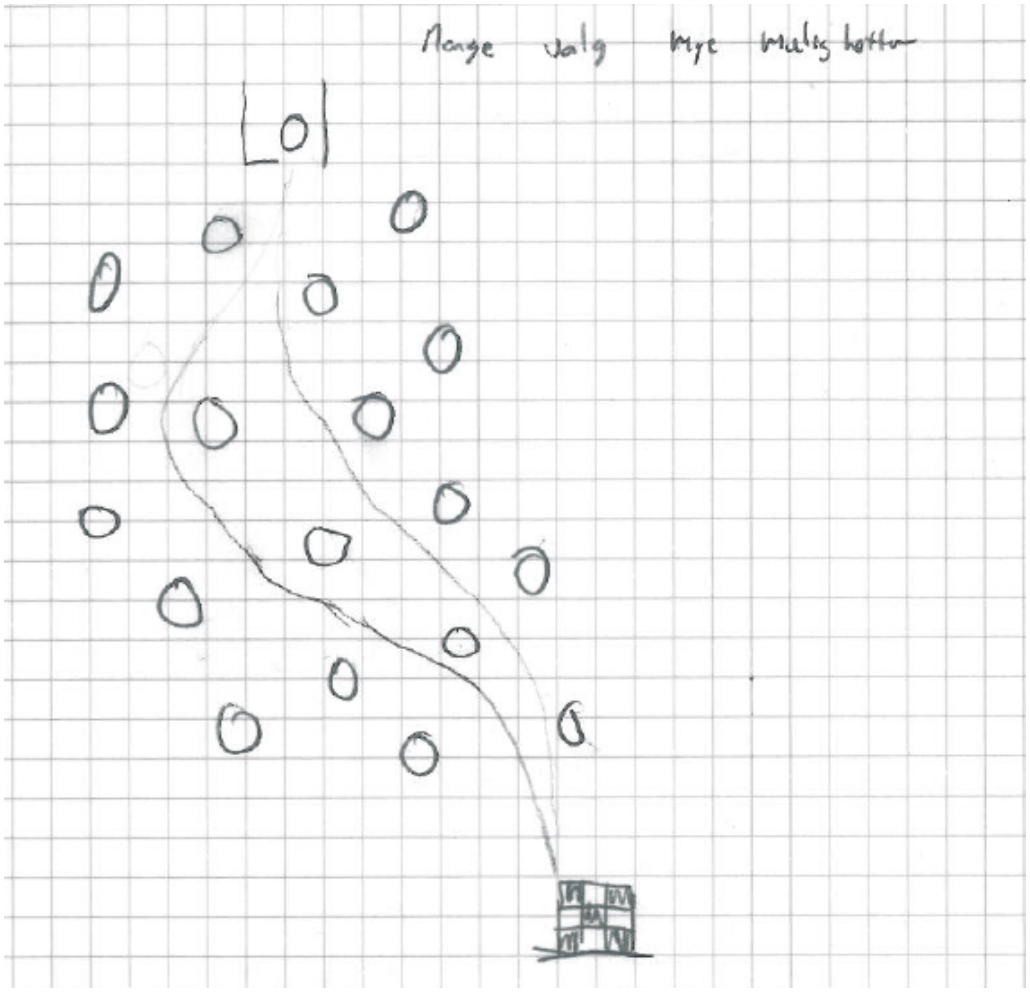


Figure 62: hand drawn image nr13

D Commit logs

Author	Commit	Message	Date
Øyvind Nordli	9a38f90	new button highlight sound	5 hours ago
BoMann	4e95689	made the changes titten wanted	7 hours ago
BoMann	2f813ad	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	16 hours ago
BoMann	caa32a7	fixed level 10	16 hours ago
zilter	a25bd4c	flipped the goal flag	yesterday
Øyvind Nordli	490e424	new background music	yesterday
Christian Holt	f5d06d4	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2 days ago
Christian Holt	b4b64d3	Added new menu music files	2 days ago
zilter	e62bbed	Some new art	2 days ago
zilter	6ac1579	Some new art	2 days ago
BoMann	dd2ba2c	Fixed rails with new 50x50 look	3 days ago
Øyvind Nordli	bb6708c	php stuff	3 days ago
BoMann	570994c	fixed rail reset for reals i mean it this time	4 days ago
BoMann	7e5db2f	fixed rail reset for reals	4 days ago
BoMann	98ec834	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	4 days ago
BoMann	5412379	added resett and changable icons to rail, also made rail bigger so all levels had to be redone	4 days ago
Christian Holt	d4e4524	Fixed a bug where the mute button would reset when returning to the main menu, causing it t	4 days ago
Christian Holt	d251cf3	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	4 days ago
Christian Holt	b3d12c6	Added .mp3 versions of all sound files for browser compatability	4 days ago
Øyvind Nordli	ab9e647	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	5 days ago
Øyvind Nordli	b129c53	locks on locked levels	5 days ago
zilter	8e27597	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	5 days ago
zilter	8ef6955	Plenty of rebalancing levels	5 days ago
Christian Holt	9cc029e	Removed some magic numbers in magnet.js	5 days ago
Øyvind Nordli	f7659c4	ui update	5 days ago
Øyvind Nordli	c019f80	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	5 days ago
Øyvind Nordli	5561ff6	fixed bug in particles	5 days ago
zilter	72da31f	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	5 days ago
zilter	d42f1e4	Lots of artwork	5 days ago
Øyvind Nordli	e9409d3	fixed particle color	5 days ago

Figure 63: Commit log page 1

Author	Commit	Message	Date
zilter	5b139a6	backgrounds	5 days ago
zilter	81d22fb	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	5 days ago
zilter	7e5e0e6	tilesheet testing	5 days ago
Christian Holt	24299a9	added another background for testing, fixed zindex for restart and menu buttons. Added som	2013-05-09
Øyvind Nordli	c86d728	files...	2013-05-09
Øyvind Nordli	352a7e3	updated background and tilesheet	2013-05-09
Christian Holt	62dca0b	Added an unused folder for entities that were created, but never used	2013-05-09
Christian Holt	af2f2d3	Went over every file and updated comments. Added the MD5.js file into the local filesystem.	2013-05-09
Christian Holt	0341347	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-05-07
BoMann	189c492	Done stuff	2013-05-07
BoMann	0ea63f6	new sound files merge	2013-05-07
Christian Holt	55c2b1d	fixed menu ordering	2013-05-07
BoMann	53cffa6	randoms tuff i level 1	2013-05-07
Christian Holt	9797f21	Fixed up some language derps as well as fixed the background size of credits	2013-05-07
Christian Holt	fd34bb3	Fixed up creditsScreen and a gamebreaking bug where you could create infinitely powerful	2013-05-07
BoMann	bfbb36b	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-05-07
BoMann	de9cbb5	new sound files	2013-05-07
Christian Holt	3db84eb	fixed level1 crash	2013-05-06
Christian Holt	7217475	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-05-06
Christian Holt	0463531	Added credits button, credits screen, fixed a couple of small bugs, edited away a lot of spellin	2013-05-06
BoMann	faf70a7	added alot of assets and added level 15	2013-05-06
Christian Holt	df17dbd	Fixed a bug where the player would not be affected by magnetism by electromagnets after	2013-05-03
BoMann	1dc02aa	added information on all the information screens	2013-05-03
zilter	11e73c8	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-30
zilter	a675a52	merge	2013-04-30
Christian Holt	05e0fc8	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-30
Christian Holt	562ae97	Added images for info and end screens	2013-04-30
Øyvind Nordli	097b301	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-30
Øyvind Nordli	ba43a5f	fixed particles	2013-04-30
Christian Holt	f79fcde	added win sound on level end, added level 14 and fixed up XML paths in info and end screen	2013-04-30

Figure 64: Commit log page 2

Author	Commit	Message	Date
zilter	27f53be	initial level 13	2013-04-30
BoMann	127140c	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-30
BoMann	c031a60	mange ting blir gjort og fiksa	2013-04-30
Christian Holt	914f986	Added some more sound effects, fixed a bug in lolgate where it caused the restart button to	2013-04-30
Øyvind Nordli	ee3e549	merge	2013-04-30
Øyvind Nordli	22ab607	particles	2013-04-30
Christian Holt	9db6870	Fixed a bug with playing of the resize of magnetic field sound effect. It should now play	2013-04-30
Christian Holt	e5a0a12	Added the rest of the temp levels	2013-04-30
Christian Holt	b5f4dd1	added level 10	2013-04-29
Christian Holt	60b058c	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-29
Christian Holt	96a1ed1	Added in info and level end screens for several levels, added the strings into the XML files	2013-04-29
zilter	ff59d4c	Level 11	2013-04-29
zilter	0576a45	I cant even remember	2013-04-29
zilter	45fcfed	fixing file naming error	2013-04-29
zilter	cd22f34	fixing file naming error	2013-04-29
zilter	4b7467a	merge	2013-04-29
zilter	6149e86	added lolgates to levels	2013-04-29
Christian Holt	22596aa	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-29
Christian Holt	578d870	Some bug fixes, other small fixes, sped up rails, added some sounds	2013-04-29
BoMann	0e2e5d5	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-29
BoMann	501ecfc	level 8 is a now a made a	2013-04-29
zilter	39c0038	lolgates	2013-04-29
zilter	4a7e15f	merge conflict	2013-04-29
zilter	a67770f	lolgates, and some mumbo in springboard	2013-04-29
BoMann	366c999	end and info screens now work for the first 7 levels	2013-04-28
BoMann	861cbe6	info screen testing og end screen finnished + last sec change	2013-04-27
BoMann	7738927	info screen testing og end screen finnished	2013-04-27
Christian Holt	a9f5ed6	Added back in some test levels for showing off some functionality	2013-04-27
Christian Holt	3279e8f	Fixed several bugs in rail.js, the magnet can now be turned on and off	2013-04-26
Christian Holt	7abf001	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-26

Figure 65: Commit log page 3

Author	Commit	Message	Date
Christian Holt	5a16019	Created and added a bunch of levels, also merge.	2013-04-26
Øyvind Nordli	05a391d	invisible ring on magnet when not targeted	2013-04-25
Øyvind Nordli	9456ca7	merge conflict	2013-04-25
Øyvind Nordli	5b1f276	arrows show on scrolling	2013-04-25
BoMann	e53e54e	Rails are now perfect	2013-04-25
BoMann	8944376	stuff for rail	2013-04-25
zilter	83e4b76	magnets can be set as not interactive	2013-04-19
BoMann	003c815	good stuff and more level 3	2013-04-18
BoMann	093053d	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
BoMann	44732c9	Level 3 gon be made	2013-04-18
Øyvind Nordli	f1c3488	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
Øyvind Nordli	b0a1595	started on arcade level and created level6	2013-04-18
zilter	a24aee0	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
zilter	9e63089	Level 4 finsihed-ish	2013-04-18
Øyvind Nordli	7922acf	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
Øyvind Nordli	52e229e	level5 fix	2013-04-18
zilter	569657d	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
zilter	785840c	switches can't be spammed	2013-04-18
Øyvind Nordli	8448290	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
Øyvind Nordli	6863450	created lvl5 and fixed electromagnet targeting	2013-04-18
Christian Holt	0b8b98b	Another small fix	2013-04-18
Christian Holt	c585e11	Fixed a bug where magnets would increase in power both when increasing and decreasing th	2013-04-18
BoMann	f8e91f5	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
BoMann	5281034	pendelum motor working	2013-04-18
zilter	3ce91de	Championship-level coding skills	2013-04-18
zilter	42eac34	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
zilter	137cc69	oops, this is javascript	2013-04-18
Øyvind Nordli	4e3f20d	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-18
Øyvind Nordli	b93327a	start button now working	2013-04-18
zilter	eb41753	Fixed polarity, work on Level 4	2013-04-18

Figure 66: Commit log page 4

Author	Commit	Message	Date
Øyvind Nordli	378bc90	created the first two levels	2013-04-17
Christian Holt	5e86c0c	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-17
Christian Holt	0819c25	Fixed another bug on the magnet fields, they now start affecting properly	2013-04-17
Christian Holt	aaeb908	Fixed another massive bug with magnetism breaking	2013-04-17
zilter	82f0990	merge	2013-04-17
zilter	de430fa	merge	2013-04-17
BoMann	cbee91e	pendulum merge	2013-04-17
zilter	9ca29e4	switchsounds, polarity magnetism and tutoral level 2	2013-04-17
BoMann	87aa3a7	adda pendulum	2013-04-17
Christian Holt	db03782	Added vertical screen scrolling	2013-04-17
Øyvind Nordli	767583c	fixed pop-up background position	2013-04-17
Christian Holt	f791b36	Fixed a major bug with magnetism when the screen was moving	2013-04-17
Øyvind Nordli	74a43ca	created camera	2013-04-17
zilter	3e41125	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-17
BoMann	af3069a	lotsa thingys now work, magnets and boards and stuff merge	2013-04-17
BoMann	8669e41	lotsa thingys now work, magnets and boards and stuff	2013-04-17
zilter	8d60126	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-17
Øyvind Nordli	d6ed6ef	camera fix	2013-04-17
zilter	7d79b78	merge	2013-04-17
zilter	618137e	Working with polarity	2013-04-17
BoMann	c7be41c	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-17
Øyvind Nordli	7a12e98	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-17
Øyvind Nordli	60938ed	fixed magnet targeting when jumping to a new level	2013-04-17
Christian Holt	d9a6a71	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-17
Christian Holt	5f39bf2	Bugfix and static UI fixed	2013-04-17
BoMann	bbdc835	more changes to board	2013-04-17
Øyvind Nordli	a073afe	fix	2013-04-17
BoMann	1628aa3	ekstra magnets merge	2013-04-17
BoMann	4bb7766	ekstra magnets	2013-04-17
zilter	0f0caef	Merge	2013-04-17

Figure 67: Commit log page 5

Author	Commit	Message	Date
zilter	e512339	ball-triggered switches	2013-04-17
Øyvind Nordli	9204f50	fixed level1	2013-04-17
BoMann	2fc9edc	board now have its magnet merged	2013-04-17
BoMann	f85a5e2	board now have its magnet	2013-04-17
Øyvind Nordli	aeaca15	small fix on goal in level1	2013-04-17
Øyvind Nordli	a85123e	tutorial level1 finished	2013-04-17
BoMann	2568fd0	board changes angain	2013-04-17
BoMann	c299923	board changes	2013-04-17
Christian Holt	953d749	Changes to level1	2013-04-17
BoMann	ad5a1f4	added a comment in main and code in board	2013-04-17
BoMann	d4d36a2	added a comment in main and code in baard	2013-04-17
Christian Holt	83f3937	Added restart button properly!	2013-04-17
Christian Holt	db040a9	Debug changes	2013-04-17
BoMann	e3a3604	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-17
BoMann	1874bb3	bigger balls	2013-04-17
Christian Holt	08fbda7	mergeBOB	2013-04-17
Christian Holt	8fab104	reset buttons and epic magnet battles!	2013-04-17
BoMann	f237481	Pendel working atm	2013-04-17
BoMann	dfb7332	merging	2013-04-17
BoMann	8b30095	merging	2013-04-17
Christian Holt	7b698df	merge	2013-04-16
Christian Holt	dccc697	Restart soon working	2013-04-16
zilter	45c8d69	aaand a couple of files	2013-04-16
zilter	cf5946e	Electromagnets done	2013-04-16
zilter	66c3e7b	electromagnet merge conflict	2013-04-16
zilter	788c059	electromagnet with some merge	2013-04-16
Christian Holt	259605e	merge	2013-04-16
Christian Holt	3ace02f	Made some bug fixes, and started work on restart button	2013-04-16
Øyvind Nordli	4010de4	fixed some ready functions instead of ugly code in update	2013-04-16
Øyvind Nordli	bf4f9f4	menu button fixed	2013-04-16

Figure 68: Commit log page 6

Author	Commit	Message	Date
Øyvind Nordli	2761773	added small button sprite	2013-04-16
BoMann	23ae8ac	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-16
BoMann	641d1eb	started on the spring board	2013-04-16
Øyvind Nordli	7852000	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-16
Øyvind Nordli	8da946f	fixed callback for ending matric session	2013-04-16
Christian Holt	5b5b837	Resizing of magnets and such should be done	2013-04-16
Øyvind Nordli	5ad2754	fixed creation of save on first login	2013-04-16
Christian Holt	9b56181	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-15
Christian Holt	b461f47	Added preliminary support for 'click & drag' magnet strength and radius control	2013-04-15
Øyvind Nordli	eb687df	small fix	2013-04-15
Øyvind Nordli	f06bc28	added md5 file	2013-04-15
Øyvind Nordli	42542d5	metric session functions done	2013-04-15
Christian Holt	bd89565	Support for multiple sliders added	2013-04-15
Øyvind Nordli	72b8b7b	save is finished	2013-04-14
Christian Holt	4c43e3b	Added some new sound effects for shield and language buttons, added slider and sliding of	2013-04-12
Øyvind Nordli	aa443bb	small fix in save	2013-04-12
Øyvind Nordli	8f0acf1	fixed so the game works without logging in	2013-04-12
Øyvind Nordli	f22a545	now uses the uid of logged in user instead of hardcoded 1	2013-04-12
Øyvind Nordli	5965910	created a php-files to use web tics. Now loads save from database and not from xml	2013-04-12
Øyvind Nordli	522ff2b	changed salt and sitekey in the user1.class.php	2013-04-11
Øyvind Nordli	906310b	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-11
Øyvind Nordli	490b1f1	a few small fixes	2013-04-11
BoMann	d339fe4	Goal and Basket is done with message	2013-04-11
BoMann	22409c7	Goal and basket kill message is beeing worked on	2013-04-11
Christian Holt	027657b	Merge	2013-04-11
Christian Holt	f4ef651	Added a mute button on the main menu	2013-04-11
BoMann	1ac5544	Goal and Basket is done	2013-04-11
BoMann	215ce07	gate works atm	2013-04-11
Christian Holt	7c42264	fixed an issue where we didn't check for unlocked levels when using keyboard	2013-04-11
Christian Holt	c73ec75	Added sound effects on button highlight and button presses, both for unlocked and locked	2013-04-11

Figure 69: Commit log page 7

Author	Commit	Message	Date
BoMann	d3fd56d	abit more chnages	2013-04-11
Christian Holt	9fda0a9	Added some functionality to not restart a song when swapping screens if it is the same song	2013-04-10
Christian Holt	ec13ebe	Added music support	2013-04-10
Christian Holt	6c66747	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-10
Christian Holt	667bfcc	Fixed an issue where magnets would apply magnetic force the themselves	2013-04-10
BoMann	03d77b7	random fix	2013-04-09
Christian Holt	96c7661	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-04-08
Christian Holt	b910f90	Affecting magnets with magnetism and such, also, merge.	2013-04-08
BoMann	baf6e51	Addet totorial 2 for reals and an updated spring board.png	2013-04-08
BoMann	cd9f148	added totorial level2 og ny tegna spring board tile	2013-04-08
Øyvind Nordli	a839c98	small fix in the meny with zIndex and id so the buttons are correctly numbered and the shield	2013-04-04
BoMann	37c4fbc	added totorial level 1	2013-03-31
Øyvind Nordli	eb0e641	created a web page with login for the game(s), need to be done more on this when/if we	2013-03-23
Øyvind Nordli	d240e9c	added gitignore file and bootstrap	2013-03-23
Christian Holt	8439f90	Gravity, magnetism, and slopes are now working correctly, and together with each other.	2013-03-22
Øyvind Nordli	6d6e5b2	slope fix in box2d game.js	2013-03-22
Christian Holt	fc2ede4	Added the Box2D physics engine, and merged this into the already existing framework. 'real'	2013-03-21
Christian Holt	ce4d69d	Magnetism is implemented and working, some tweaking may be needed for the drop off rate	2013-03-14
Øyvind Nordli	1939bf7	started wokring on a save.xml file, but this needs to be done another way, since	2013-03-14
Øyvind Nordli	b0f53c5	created a pop-up message with an extend on the infoScreen, also updated to another font	2013-03-14
Christian Holt	4e201cc	Made progress with the magnetism simulation	2013-03-13
Christian Holt	01326e5	Working on magnetism implementation	2013-03-13
Øyvind Nordli	424bad2	fixed an error in the infoScreen and created an example for the first info screen	2013-03-12
Øyvind Nordli	efb2de6	created a info screen, uses it on the splash screen and removed the old entity	2013-03-09
BoMann	4cf764f	Added the leveldesign media files	2013-03-07
Christian Holt	ca1a7c9	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-03-04
Christian Holt	badeea9	merge	2013-03-04
Øyvind Nordli	44ea8d8	merge error	2013-02-27
Øyvind Nordli	631ae34	merge	2013-02-27
Øyvind Nordli	6958bf2	finished the main menu, all buttons should be working, and flags now change language with	2013-02-27

Prev 1 2 3 4 5 6 7 **8** 9 Next

Showing 211–240 of 262

Figure 70: Commit log page 8

Author	Commit	Message	Date
Christian Holt	47754a1	Merge...	2013-02-26
Christian Holt	24a02ad	merge	2013-02-26
Øyvind Nordli	d315cae	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-02-25
Øyvind Nordli	fa94beb	button selection is no longer bugged in chrome and the text labels in the menu is now loaded	2013-02-25
Christian Holt	62641c6	Added gravity, and set gravityFactor for buttons. Collisions with test level added so the	2013-02-21
Øyvind Nordli	99273fb	button highlighting now works with mouse as well as arrows, mouse overrides buttons	2013-02-21
Øyvind Nordli	e59fe2e	can click buttons to go to next level	2013-02-21
Christian Holt	b53ab02	Started working on implementing the decay function and field effect radius drawing	2013-02-21
Øyvind Nordli	686f239	main menu up and running, move with arrows and select level with enter-key	2013-02-21
Christian Holt	cbcaedd	Added magnet entity, fix'd another entity bug, modified the loadLevel method, set up working	2013-02-20
Øyvind Nordli	e92cfe5	fixed coding convension in some files	2013-02-20
Øyvind Nordli	b03dcaa	splash screen done, started on the menu buttons	2013-02-18
Christian Holt	bb3874d	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-02-14
Christian Holt	e2140b8	Added entities for player, button and trigger	2013-02-14
BoMann	35f1069	added .php index	2013-02-14
Christian Holt	a8f8f58	initial commit of Impact.JS	2013-02-14
BoMann	1638f57	herp test commit	2013-01-30
zilter	ee8873a	testing and stuff	2013-01-30
Øyvind Nordli	b641e45	test	2013-01-30
Christian Holt	7bf9850	Merge branch 'master' of https://bitbucket.org/ymabob/2013-games-for-science-bachelor	2013-01-30
Christian Holt	3d0f8cc	Reinitialized repo and added the basic webpage	2013-01-30
Christian Holt	ed580f6	Readme added	2013-01-24

Figure 71: Commit log page 9

E Depth Jam Report

Depth Jam

Group:

Christian Stokkebekk
Christian Holt
Morten M. Andersen
Øyvind Nordli

Time and place:

This years depth jam was housed at Høgskolen i gjøvik, a school in Norway.
It started on Tuesday at 12:00am the 16th of April, and ended thursday 18, at 12:00am.

Theme:

The theme for depth jam was: Pendulums.

Purpose:

The purpose of a depth-jam is to select some feature(s) and spend a set amount of time, in our case 48 hours, developing this feature. Because this jam was done for the rapid prototyping and innovation course we chose to create something new.

Report:

We got a tremendous amount of work done in the 48 hours we were going to work. Not only did we finish the pendulum, but we created a lot of other parts necessary for the game to run properly.

Most of what was done are:

- Electromagnet
- Menu button
- Reset button
- Switch
- Camera panning / screen scrolling

- Pendulum
- Springboard
- Level 1-6
- Started on the Arcade level
- Magnet polarity
- Smaller walls for collisions and such.
- Finished the click and drag for manipulating magnetic field radius and field strength

Best:

Focused work:
One of the best aspects of a depth-jam is having set out 48 hours to do work on the project and nothing else. It lets you focus on the task at hand without worrying about other projects going on, or procrastinating as easily on the internet. This enables us to get a lot of work done quickly as opposed to dragging it out over a longer timeframe. In essence it works as a mid-project boost/burst.

Working at the same place:
Having all 4 group members sitting around the same table worked very well. If any one of us had a question we could just shoot it out and get answers or further help if needed. As some members of the group had a lot less experience programming with JavaScript, ImpactJS and Box2D than others, this was very helpful.

All the small things:
Since one of our goals on this jam was to create at least 2 levels, we were forced to create all the small things needed for a level to work properly. Like a back to menu button, a reset for the ball, a goal that sent the player to the next level etc. It was really nice to see all those get into place and actually have some functional and running levels in the end. Made it easier for us to see how far we had come in the project so far.

Worst:

Box2d and Impactjs:
box2d in impactjs is not updated to the newest version, this causes some problems. Mostly concerning gravity and the placing of objects. It is not possible to place a static object at one point, and later have it affected by gravity. This is possible in later versions of box2d, but the plugin version created for impact runs on 2.0.2 instead of 2.1 or newer. We tested a version of 2.1 to see if it worked, but big parts of our earlier work bugged when we tried this.

Camera Panning:
There was no problem at all moving the camera around in the game, but it caused a lot of problems concerning ui elements we had created earlier. We had stupidly created those to fit on the screen, but without thought of a moving camera. So 2 members ended up spending 5 hours to fix it. It was an easy fix on most of the elements, but the pop-up message for example seemed to live a life of its own. We made it work after a while, but it felt like a lot of wasted time.

Personal notes

Øyvind Nordli:
This depth jam was really helpful to our project. It gave the project a nice boost, and since we wanted to have some levels ready at the end of the jam. We were forced to create all the small things binding our project together. Even though these were a lot of small things, it was a huge help in the bigger picture, so we could start testing and creating more levels.

Christian Stokkebekk:

In accordance with what the rest of this document states, I consider the depth jam to be a good burst of activity, that was very much needed for the workflow of our project. We had been slacking off quite a bit prior to the jam, and it helped kick productivity back to where it should have been all along.

Morten Møller Andersen:

One of the most help some things we have done all year. We had been wasting a lot of time on other things so we were behind, but thanks to the Jam we are now ahead.
Easier to work when you know you can get help instantly, and by being everyone together it's harder to procrastinate without feeling guilty.

Christian Holt:

Having a depth jam that focused on improving our bachelors project was extremely helpful. Not only did we get new features, but it also forced us to polish and finish up quite a few other quirks required for multiple levels, like the restart button and camera panning. The jam worked as a mid-project crunch for us, and we got a lot of stuff done. Overall doing jams is a fun and creative experience that lets us focus solely on what we're doing, with almost no distraction around. It is a lot harder to procrastinate and waste time on small stuff when you have your other group members sitting beside you, and it is a lot easier to get help, or additional input, if you need it.

Compared to my first game jam we decided to actually sleep more than 3-4 hours a night, because being extremely tired can be more of a hindrance than one may think. As we used the same tools for this jam as I did on my first game jam, I really felt the difference in knowledge of the tool when I was writing code. The speed of which we pushed stuff through the door was a lot higher. If you’re doing a game jam for the purpose of learning a new tool, that works pretty well, but if you’re doing a game jam to create a complete product, it is definitely a major plus to work with something you already know, and already have an existing code base for.

F Pre-Project plan

In the section that follows is the pre-project plan of the project.

Games for Science

Pre-project Plan

Christian Stokkebekk
Christian Holt
Morten Møller Andersen
Øyvind Nordli

Table of contents

- Table of contents
- 1. Goals and limitations
  - 1.1 Background
  - 1.2 Project Goals:
    - 1.2.1 End product goals:
    - 1.2.2 Future effect goals:
  - 1.3 Limitations
- 2. Scope
  - 2.1 Detailed Project Description
  - 2.2 Out of Scope
- 3. System Development Model
- 4. Risk Analysis
- 5. Quality Assurance
  - 5.1 Development Environment
  - 5.2 Standards for coding
  - 5.3 Documentation
  - 5.4 Configuration management
  - 5.5 Group rules
- 6. Gantt
- Appendix
  - A: References & Attributions:
  - B: Coding conventions
  - C: Group rules for G4S

1. Goals and limitations

1.1 Background

We are four students who study game programming at Gjøvik University College. We are all passionate about science in general, and wanted to create a game to serve as a gateway into the wonders of the scientific world. Instead of simply making a game, we have embarked upon the task of creating a shared framework for multiple educational games dedicated to teaching science to the younger generation. We're going to use Vitensenteret [1] as a user group to get feedback and help in determining how we should portray our game for our target audience.

The framework will be a common platform for sharing results and achievements from within the educational games.

The idea for the framework comes from Simon McCallum, he will function as our employer.

Supervisor:
Simon McCallum

Advisors:
Kurt Squire (external)
Jayson Mackie

User Group:
Vitensenteret Innlandet [1]

Group Members:
Christian Stokkebekk, Christian Holt, Morten Møller Andersen, and Øyvind Nordli.

1.2 Project Goals:

The goal of the project is to have a usable framework and 2 games (at least 1 full) integrated with the framework.

1.2.1 End product goals:

Framework Goals:
At the end of the project the framework should be:

- Compatible with further development (more games) by third parties.
- Able to properly communicate with the "Webtics" software.
- Allowing for integration with learning management systems such as MOODLE.
- Able to keep track of a common user for all the games.
- Able to handle scoring (High scores, personal records etc) and achievements.
- Multilingual, with the primary language as Norwegian.

Game Goals:
At the end of the project the games should be:

- Running a stable version.
- Educationally purposeful.
- Communicating with the framework.
- Able to track and log all important events.
- offering a degree of difficulty suitable for the user group, both game mechanically and educationally.
- Able to offer an interactive and entertaining learning experience.
- Multilingual, with the primary language as Norwegian.

1.2.2 Future effect goals:

We would like the project to have the following effects:

- Satisfied schools and learning environments.
- Games more accepted as a learning tool.
- Make it easier for others to develop games for education and science.
- Give children the possibility to learn at their own pace, since not every child learns at the same rate.

1.3 Limitations

All user data will be stored and saved in compliance with Norwegian laws regarding privacy policies. Because our target audience is beneath the age of consent, these laws are even more strictly enforced.

There will not be any economic gain from this project, it's goal is to educate, not profit. There will be no salaries involved, and no foreseeable expenses other than traveling to meetings and interviews with external resources, i.e companies within the genre we're aiming at.

The project will consist of different phases. The alpha version of the framework and the games should be done by the 5th of April.
We are targeting the week after easter for our first user group testing of one or more of the game. Thanks to Norwegian laws, we can't log information or ask questions about the game on the usergroup without permission. Because of this we will give the children the games and then see if they enjoy it and if they get stuck while we're observing from the same room.

The transition from alpha to release should primarily be done by the 5th of May. With some room for emergency bug fixing. The written report is due the 15th of May, and the verbal presentation is ~5th of June.

The development team will consists of four programmers, at least one graphics artist and one sound composer. Everyone in this bachelor group will act as programmers, and the graphics artist and the sound composer will be freelanced.

The software will be web based and accessible from all modern browsers. It is also important to take into account that computers in schools and other public facilities might

have below average hardware.

2. Scope

2.1 Detailed Project Description

This is our current project description, but it's not the final description so it might change during the design phase of the project.

Our Project is divided into two main parts; The common framework and the games made to communicate using the framework. In addition we would like to set up a website that works in similar ways as XBLA or Kongregate, in that you have a user spanning over multiple games with a user level, score, achievements and such. This website will access the logging system, "Webtics" [2], and read the data needed to display the information of the user. The website itself is not a part of the framework, but it will facilitate the idea of having all of the games made using the framework in one place. We would like to give the user an avatar, or a recurring character that is used throughout the games, hopefully giving more incentive to play the games.

Framework:
The framework will consist of the shared codebase used by all the games. User authentication and login, achievement tracking, scoring, event logging, and communication with "Webtics" are features that are going to be shared by most of the games. The framework will be built as we work on the first game, then reused for the second game.

Games:
We want to create at least 1, hopefully 2 smaller games that can interact with the framework. We will focus on making games for science education and not science research. We put some thought into which of these we wanted to do, and we chose education because we feel that we don't have the experience to do the proper research necessary to make a game providing scientific results. We're taking inspiration from games such as MinecraftEdu[3] and Dragonbox [4] when choosing education, in that we want the students to enjoy playing a game, and still learn at the same time. It is important that we try to avoid a Shavian Reversal (Papert, 1998)[5] when creating our game. When trying to take the best of both worlds, it is easy to end up with the worst of both instead, a boring, uninteresting game that doesn't teach anything well. To avoid this we want to interview the creators of Dragonbox, MinecraftEdu, Vitensenteret Innlandet in Gjøvik, and other companies within the games for education genre.

We want these games to teach a specific subject within a scientific theme. The reason for this is that we feel this makes it easier for a lecturer to include our games in their curriculum. We believe that a lecturer is more likely to lecture for a while, then include our game in the class, rather than have the student play a game that is, say, 4 hours long. In these short games we want to teach the students through the use of mechanics, rather than flow breaking popups and messages that tell you to do a task. Here we feel that Dragonbox is a great example in the way it teaches math. At first an unbeknownst player might not even see that he is in fact doing math, and only later discover that he is in fact doing algebraic equations.

Due to the fact that we want our games to be able to run both on computers and on phones & tablets, we chose a web based approach. It is important that we keep this in mind when

designing the controls and UI elements of the game. Things that work well on PCs, might not work so well on phones, and vice versa. Keeping this in mind, we're going to aim at games that don't require extremely precise, or fast actions.

2.2 Out of Scope

There are things that we consider out of the scope of our project. First and foremost we're going to steer away from doing a 3D environment or 3D graphics, as this will take more time, and may require 3D models. We would like to be able to finish both the games, therefore doing 2D is better for us due to time constraints and previous experience with 2D.

We are going to look into Learning Management Systems (LMS) and discuss the options and opportunities that come with integrating with LMSs such as Moodle. [6] We are however not going to actually integrate with any LMS because of the high amount of man hours this will eat out of our project.

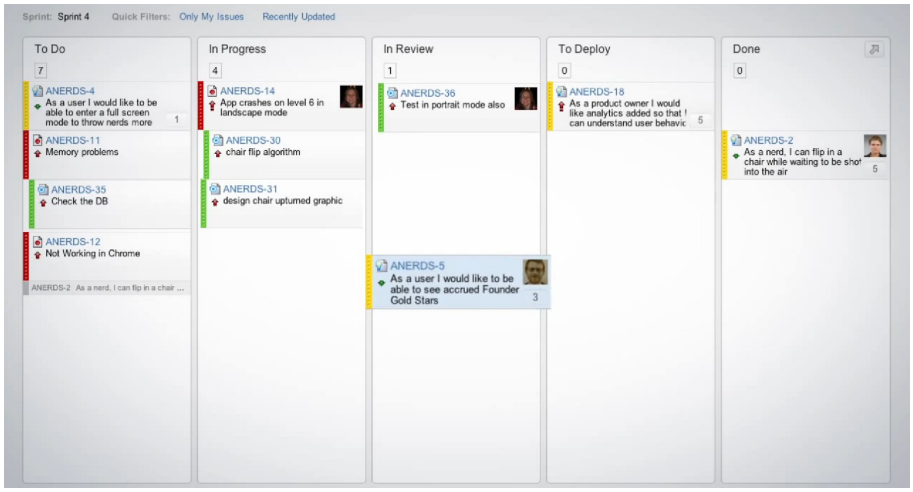
3. System Development Model

Planning the development of a relatively small project like this can be very difficult, especially as the available resources and manpower are very limited. Problems can and probably will arise during development, which very well might cause delays or setbacks. It is also very difficult to estimate the number of manhours required for a certain feature or code segment. In situations like these, the agile development models are better suited. Agile models allow far greater versatility, as new modules or upcoming sprints can be planned or altered based on user feedback or analysis of the previous sprint or module.

We have decided to pick Scrum [7] as the system development model of our choice. We will have sprints spanning two weeks, with a sprint meeting every monday. These meetings will alternate between planning the next sprint, and reviewing our progress in the ongoing sprint. It is very difficult to plan every sprint before starting development, which frankly is the idea of scrum.
Scrum is versatile enough to handle the evolution of a project and the changes we might end up making in the process of this project. In a project without clearly defined product goals and milestones all the way from the start, scrum is the way to go.

We have been given licences to a bundle of project management tools from Atlassian.[8] We will utilize JIRA[9], Greenhopper[10] and Bonfire[11] from their package.

JIRA will be used to keep track of issues and task needed to be done, and we will then use Greenhopper on top of that to maximize what we can get out of the Atlassian tools. JIRA is a good tool for setting up all the tasks and issues needed to be done in our project. We can also assign different tasks to one or more members of the team. Greenhopper is created to fit into a agile development model like scrum. It creates a list of all the issues we have in the project and then the team can estimate the value of the tasks for the upcoming sprint. You can then start a sprint and it will set up a board of the different tasks at hand.



Bonfire is created to easily do QA on the projects that are being developed. You can playtest the game through Bonfire and if you see a bug or something similar it it easy to submit a new issue to the project. It will automatically take a screenshot of the game and you can highlight with the editing tools where the bug is shown.

4. Risk Analysis

The risk we have to be most careful about is the privacy policy for kids. If this part is broken the framework and all the games built in it is more or less useless. So we have to be careful not to store information specific to any of the users.

We also have to be careful not to overscope, so we can finish the games according to the schedule. Then plan now is to make two smaller games with good mechanics over one big game with a lot of features.

Probability/consequence	Small	Moderate	Critical	Catastrophic
Unlikely	5			1, 2, 6
Probable	3	4		
Very likely				
Highly likely				

Priority	Description	Probability	Consequence	Impact	Measures
1	Privacy Policy for kids younger than 16	Unlikely	Catastrophic	Can't be played by our user group	All information about the users will be anonymous
2	A part of the framework is not finished before the time of delivery	Unlikely	Catastrophic	The games won't have a framework to run in and the assignment have more or less failed.	Always keep other members posted on what takes more time than originally anticipated
3	A part of any of the games is not finished	Probable	Small	The part of the game that is not finished will either be delivered as is or removed	When the delivery date get closer it's important to keep other members posted, so the most important parts are finished first
4	Webtics won't be finished in time	Probable	Moderate	Can slow down production while it is being finished or while we find another metrics system	Remind Simon and Jayson of when we need it to be finished. Write code as if it was working, then connect to it when it's up and going.
5	Vitensenteret pulls their support for our project	Unlikely	Small	Need to find another place to test out the game	Keep them updated on our progress and be professional

6	Loss of source code	Unlikely	Catastrophic	If a small part is gone we have to work extra hours to catch up. If the whole project is gone we have to contact the supervisor and sort it out from there.	Use bitbucket, then all members will have the project and it will be hosted online in a repository as well.

5. Quality Assurance

5.1 Development Environment

Framework:

The framework will be coded using javascript and html5.

Games:

For the game we have decided to use LimeJS. [12]

LimeJs is a Html5 framework for making games, specific game that needs to work on both tablets and phones, and in browser's.

5.2 Standards for coding

See appendix named "Coding Conventions".

5.3 Documentation

For documentation of the source code we will be using Doxygen[13] with an extension for JavaScript. The documentation will be automatically updated if we write all our comments and descriptions before the code block it is representing.

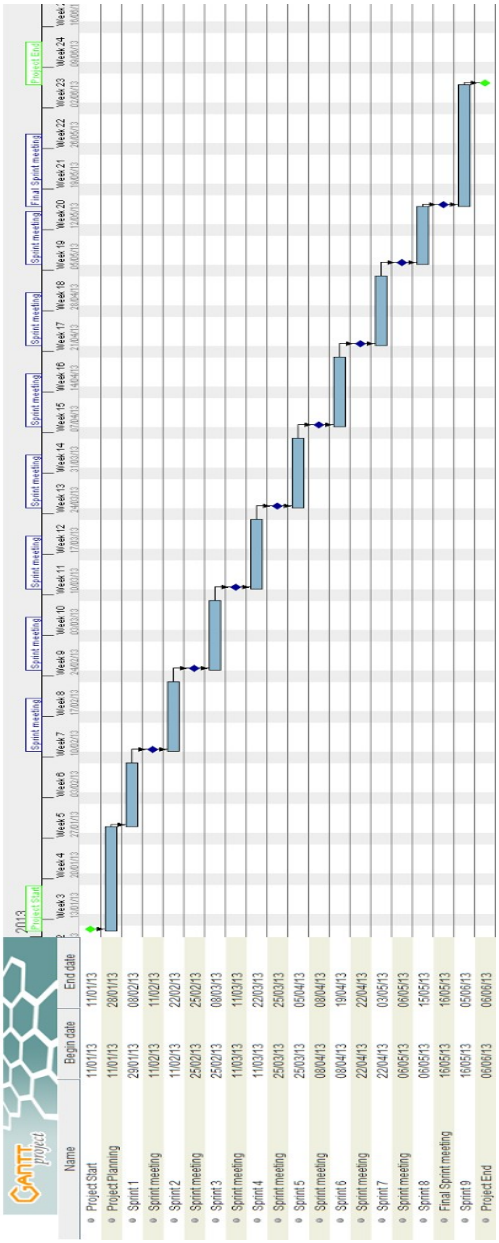
By forcing all the coders to write their comments before the code, we will minimize the possibility of having big parts of code with no comments at the end of the project.

5.4 Configuration management

We will be using Bitbucket[14] tool for configuration management, since we got the complete Jira development tools free from the school.

5.5 Group rules

See appendix "Group rules for G4S"



6. Gantt

The project started the 11th of Jan. when we had our first official meeting with the group. From that day and until 28th Jan. The biggest planning part of the project will take place.

After this the design phase of the first game will take place. This will be the first sprint. After this the content of the sprints will be filled in as the project goes on. Most of the sprints will be 2 weeks long, with some variations in the first and last sprints to hit the

deadlines. There will be a sprint meeting the first day of the sprint, this will usually be on a Monday. This is where the group set up the tasks for the next sprint.

The plan is to be done with all the coding before the end of the 7th sprint. Then the 8th sprint will be used to finish the report and the 9th to get the presentation. The presentation is scheduled to be performed 4-6th of June.

Appendix

A: References & Attributions:

1. Vitensenteret: www.vitensenteret.no/
2. WebTics Wiki https://github.com/SimonMcCallum/WebTics/wiki
3. MinecraftEdu http://minecraftedu.com/index.php
4. DragonBox http://dragonboxapp.com/
5. Seymour Papert http://www.papert.org/articles/Doeseasydoit.html
6. Moodle https://moodle.org/about/
7. Scrum http://www.scrum.org/Resources/What-is-Scrum
8. Atlassian Suite http://www.atlassian.com/
9. JIRA http://www.atlassian.com/software/jira/overview
10. GreenHopper http://www.atlassian.com/software/greenhopper/overview
11. Bonfire http://www.atlassian.com/software/bonfire/overview
12. LimeJS http://www.limejs.com/
13. Doxygen http://www.stack.nl/~dimitri/doxygen/
14. Bitbucket https://bitbucket.org/?page=overview

B: Coding conventions

In order to maintain an orderly and uniform code base, it is important that everyone conforms to a certain code standard.

Environment:
Aptana Studio 3. www.aptana.com
LimeJS
KineticJS

Documentation:
Doxygen with add on for Javascript.

Variables:
Variables should be named using lower camel case (camelBack), and should have longer informative names, rather than short ones.

examples:
playerHealth.
playerDamage.
playerRunSpeed.

Functions:
Function names should make it absolutely clear what the function is going to do.
In addition, we want to differentiate between *get* and *find* functions.

- *Get* functions should only return simple values.
- *Find* functions calculate values, and signifies that the function may spend a longer time returning the value.

```
function getPlayerHealth()
{
   return playerHealth;
}

function findPlayerPosition()
{
        playerPosition * x * y / z; // Random calculations as an example, usually more complex
                                    // than just these 3 simple operators. For example "Find path" for
                                    // an AI.
}
```

Parameters:
Parameter names should be completely clear, so when another member uses the function it shouldn't be confusing.

Correct: function(vector position);
Wrong: function(vector p);

Brackets:

We're going to use next-line brackets, as we feel this gives the code more "air".

Wrong:
```
while {
}
```

Correct:
```
while
{
}
```

Comments:

We're going to use Doxygen to generate documentation. Therefore comments for functions will have to be placed above the definition, e.g:

```
/// this is a doxygen comment
/// So is this, and it is above the sayHello function.
void sayHello()
{
  std::cout << "Hello" << std::endl;
}
```

File names:
Uppercase start letter that is then followed by lowercase letters in a logical way that when read forms words that describes what is inside that file.
We will use Upper camel case for the naming.

Resources will all have the "place" they are used as the first word of the filename (main menu, background etc.).

Folder structure:
Subject to minor changes if we find this structure doesn't work as intended.

- Project Folder
  - Web Page (HTML)
    - FrameWork(shared code base)
      - JavaScript
    - Game 1
      - HTML
      - JavaScript
      - Resources
        - Images
        - Sounds
        - Data
        - Libraries
    - Game 2
    - Game X

Indentation:
We will be using tab's for indentations.

C: Group rules for G4S

Attendance:
Mandatory attendance for all sprint meets, and at least 1 other meeting every week. This includes meetings within the group, and meetings with the supervisor.

All meetings with externals and information sources, i.e vitensenteret, are mandatory for all members.

Deviations:
If a member is ill or otherwise unable to attend, the rest of the group has to be informed, as soon as possible, preferably, at least 1 day in advance.

Attending meetings over skype can be viable if someone is unable to attend in person.

Illness:
For all illness, inform the group of your absence from meetings. If able to, attend meetings over skype.

Short term:
Unable to work for less than 2 weeks. Reallocate your current workload to the rest of the group. Some tasks for the current sprint that are of lesser importance may be postponed.

Long term:
Unable to work for longer than 2 weeks. Get official documents from the doctor that explains and verifies your absence. If someone is unable to work for more than 4 weeks, contact the school and explain the situation.

Workload:
The workload should be as equally divided as possible. The amount of work should consist of ~60 hours per sprint (2 weeks) per group member, meaning ~30 hours per week. If your allocated work for the sprint is finished in less time than expected, start working on another task, or helping the other group members.

After every work session, it is important to write down a short log of what has been done, and how long has been spent doing the task.

Minute and meeting recording:
Personal work logs will be kept with a lower time of 1 hour, meaning work will be rounded to the nearest hour. The logs should contain the date, time spent in man hours, and a brief description of what was done.

Meeting logs will contain a list of who was present, what was on the agenda, what was done and, maybe, what we're going to discuss further at the next meeting.

Shortcoming:
If you're not able to finish your workload for the current sprint, you need to tell the other members of the group as soon as possible. The work you won't be able to do is divided to other members or postponed to the next sprint.

If other members see that a member won't be able to finish his workload this will be addressed the same way as above.

Coding convention:

Following coding convention is mandatory. See other document for the coding convention.

Disputes:
Any disputes will first and foremost be solved by majority votes. If a majority vote is not reached, the vote will lean towards the group leader.

Every group member has a delay Veto, which means that we postpone the decision until the next meeting with the supervisor.

If any members in the group falls out, we will try and solve the situation peacefully, if we are unable to do this internally, we will ask the supervisor for advice and mediation.

Internal romantic relationships are not prohibited, but if it disrupts work for the group to a degree where the work starts to deteriorate, the relationships will have to be postponed until the bachelor is done.

Failure to comply with the group rules:
If the rest of the group unanimously agrees that the offending party member(s) aren't complying with the group rules, they will be given verbal warnings.

After repeated offenses, the group will schedule a meeting with the supervisor to address the problem.

If the problem still persists, another meeting should be scheduled to discuss expulsion from the group.

If an expulsion is necessary, the main project will keep all work, but the offending party may take with him the work that he has done throughout the project.