PART 6 – CONCLUSION

The following describes the last changes in Project Magma before its final incarnation as Magmageddon and our overall conclusions on the project.

THE PRODUCT

The final release contains all features from the desirable target, including HDR rendering (ReqG04) and the advanced camera (ReqG02) from the high target. As mentioned in the alpha release, some features, like island attraction (ReqP04), were implemented but later removed if they were too hard to use or didn't make sense gameplay wise. Likewise, the jetpack has been reduced to a safety measure, which can be used to stop one's fall when pushed off the island. Similarly, fog was introduced to get calmer overall look. Although it was sometimes hard to get completely rid off features or put already implemented features in the background, doing so was clearly a large improvement in the look and feel of the game, which we were quite happy with at in the end.

CHANGES

Since the alpha release the following features were added or improved:

- Power-up respawn (ReqI13) was improved in a way that power-ups now wait in a queue when no island is free to respawn on.
- Player selection (ReqUI04) has been simplified by removing the option to select colors and giving each controller slot a fixed player color.
- The dwarfs from the alpha release have been replaced by the final robot model (ReqP03), which fits much better into the game's landscape. The models feature animations for jumping, island repulsion, walking, running and hitting.
- A moving camera (ReqG02) has been implemented, which keeps the main action in focus: it zooms in on the players and all islands which they can jump too. This way, some islands may be out of the screen, but never the ones you can interact with.
- The movement of the islands (ReqI04) has been improved, so it is smoother and better reacts to collision events.
- The placeholder sounds used for all previous releases were replaced by better suiting and normalized ones.

FEATURES

- 4 robots fighting in death match mode
- 4 different maps, with their own look and playing style
- 3 attacks: ice spike (long range), flamethrower (medium range) and pushing (short range)
- 3 means to navigate between islands: jump (short range), repulsion (long range) and jetpack (backup)
- Beautifully designed islands hovering amongst pillars
- A cave full of realistically animated lava with snow dropping in from above
- Various sophisticated lighting and particle effects

THE FEEL

The game play has undergone some major changes since the first playable version and evolved from a simple prototype, where your main goal was not to fall down while navigating between the islands, to a polished product which gives players a range of choices how to defeat their opponents.

A round of Magmageddon starts out with up to four robots getting dropped into the cave onto hovering islands. Each player controls one such robot, distinguishable by its color. The slow spawn animation gives the players time to spot their robot on the field. When the last robot has landed on an island, the game starts.

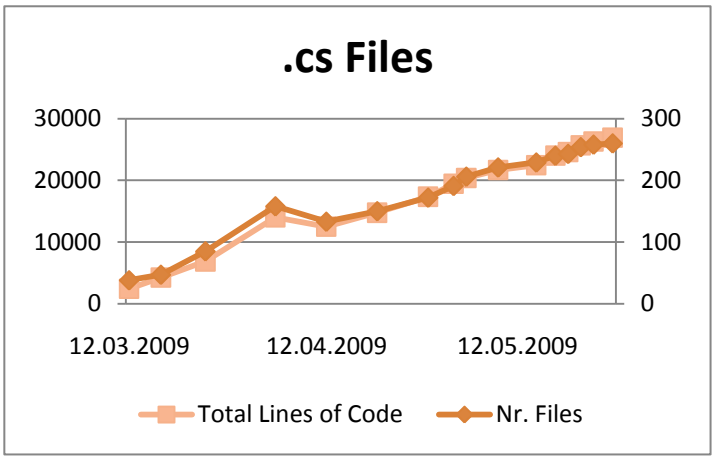
Now the players have the main choice between going mano-a-mano (or roboto-a-roboto?) with another player or us one of the ranged weapons (flamethrower or ice spike) to take him out. Whereas directly engaging another robot is free and can result in him getting pushed down the island, ranged attacks cost energy – and you cannot directly kill an opponent with your initial amount of energy. So when energy runs out you either have to wait for it to slowly recharge or find an energy power-up. Waiting, though, may not be the best idea as the other players will very probably not just sit back and relax. They'd rather directly charge at you and push you down your island. If you're skilled, though, you may still be able to activate your jetpack before falling into the lava and fly to safety. On the other hand, if your opponent is skilled to, he will throw an ice spike at you beforehand or while you're in air, which will deactivate your jetpack. Had he been the one pushed down, you couldn't have stopped him at all, being low on energy from your use of ranged chicken-weapons before. But then again, he may be low on energy too, because moving the islands to navigate through the cave needs energy too.

In the end, the essence of the game is to find the balance between not using energy at all and using it up too much so you don't have any left to navigate or throw ice spikes to stop an opponent's jetpack in time.

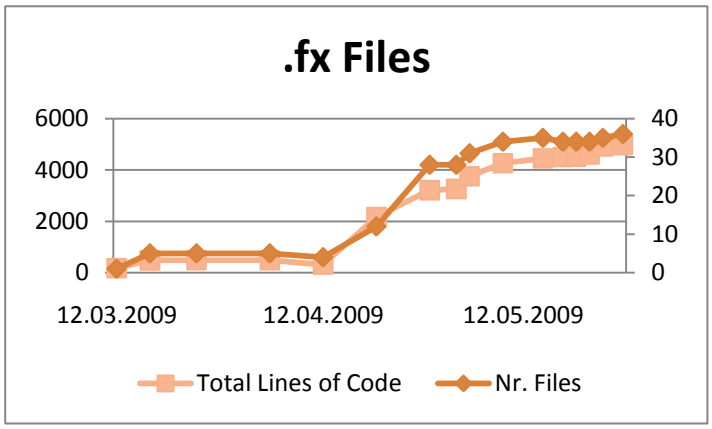
STATISTICS

In preparation to writing this document and also for the final presentation we sampled some statistics about the work that we have done. We checked out every 100th revision from the subversion repository and created an excel sheet containing the most important numbers for both c# code and effects code. We would like to point out some interesting facts that we discovered while analyzing the statistics:

- Only the first half of all subversion revisions (in total 1600) has been submitted until end of April. This means that we checked in more than 800 revisions in may, while we implemented testing feedback.



- Another indication that we have been fine tuning small details is that we wrote most of the code (nearly 20'000 lines of code) until the end of April. Only small parts have been changed or added during the final phase of the project.



- In general our code seems to be well-sized. Overall we had an average of about 100 lines of code per file. We think that this is a reasonable number and statistics confirm that we never drifted much from this ratio.
- Comparing the c# statistic with the effect statistic it is easy to see that the project started with a prototype concentrating on the core architecture and the game play. Graphics effects have been implemented after about the first half of the project.
- The loss of code between 04/02/2009 and 04/09/2009 seemed to be a bit weird first because we never deleted large parts of the code from our game and we also did not count in any external library. After a bit of research on the subversion log we discovered that this was due to the fact that the subversion plugin for visual studio which we used did not commit file deletes. So chunk files accumulated and were deleted during the questioned time period.

CONCLUSION

OVERALL

We think our project is quite a success.

First and foremost, our vision was to create a game with moving platforms, thus making the game interesting through an ever changing “playfield”. In addition we wanted the game to be a contrast between the hot Lava from below and the cold ice from above. We think that after a lot of testing we succeeded in bringing this experience to the player although it still needs some time to get accustomed to the controls.

Besides the actual game the project was also a success in other areas: The project involved a lot of teamwork, which we managed to achieve. At the beginning we needed some time to learn how the other team members think about different areas and problems within the game. But after that the communication was always straight and clear and it was clear to every member of the team how the others thought about the topics in question. Even though our discussions have been intense and heated sometimes, we think this is the better way than to always be polite and leave the other person in the dark on what you are really thinking.

Additionally, from a project management view, we were on schedule most of the time. Our aim was to meet all the milestones and to move features dynamically based on the experience gained by previous milestones reached. This means that we sometimes moved features to later milestones in favour of more important features that we needed earlier than expected. The milestones themselves have been delayed by at most one day which is, as we think, a good result.

Finally, the work with external artists, which was planned from the very beginning, paid off in the end. Of course a lot of time went into discussions with the artists on how to model different aspects of the game. It was also uncertain for quite some time if we would ever get a player model and we prepared some replacement models ourselves. In the end we received everything we needed and were able to show a game with high quality, professional models.

Considering the complexity of a project like Magmageddon and looking at the aspects mentioned above we think that the project overall went quite smoothly, with some hard, but solvable problems in between.

JANICK BERNET

My personal goal was to create a game which is fun to play with your colleagues and which contains some novel ideas. To achieve this, we had to experiment with a lot of different weapons and types of movements and had to tune hundreds of parameters to get the working features balanced right. I myself especially had many different approaches for the various collision-responses to experiment with, as depending on the objects involved and their movement a different type of collision-response may be appropriate. For instance for

an island colliding with a pillar, you can simply reflect the island's velocity at the collision-normal, while for a player using his jetpack this doesn't work. It was sometimes hard to get rid of an algorithm which doesn't create pleasing results, but was a huge effort to implement, or to even drop some features, such as island attraction, which may have had many hours of work dedicated to them, completely. But in the end it is a necessary to get a streamlined and balanced game play and I think we did a very good job at that.

Overall, working on a game was a great experience and definitely one of the high points of my ETH career. Although time consuming, having one's own game to show to people – and to watch them even enjoying it – is extremely rewarding. Still, I should warn future Student that it is not the best idea to also do Compiler Design I in the same semester. And it gets even harder if you manage to get your system disc to break down three weeks before the semester's end…

DOMINIK KÄSER

It has always been clear that the game laboratory will be one of my most challenging classes at ETH. Retrospectively, it was a big journey indeed. On one hand, I could fill my knowledge gaps in the areas of GPU programming, 3D modeling and software architecture using C#. On the other hand, though, it was important to get the experience of working full-time on a reasonably-sized project for multiple months with two team members. Since we all had quite strong (and not always agreeing) opinions about both design and implementation, this raised several challenges which we were confronted with. It turns out that meeting regularly in real-life is a crucial aspect to avoid misunderstanding and inefficient online chats. Another point is that having a bugtracking system is essential.

Our goal in "Magmageddon" was to obtain a visually appealing game which was fun or even addictive to play and running smoothly on the X-Box. As three very motivated people whose teamwork got gradually stronger with time, I think we successfully reached this goal and I hope that we can go beyond the current result at some point. First, though, a great deal of holidays will be needed!

CHRISTIAN OBERHOLZER

I always wanted to visit the game development laboratory from the very first moment on that I arrived at ETH. Because of that I also had a lot of expectations both in terms of the course itself and also in the end result that we had envisioned. And of course in the end it was hard to fulfil the expectations.

In terms of the project itself I think everything is nicely summarized in the overall conclusion above. But regarding the course itself I would like to mention some points.

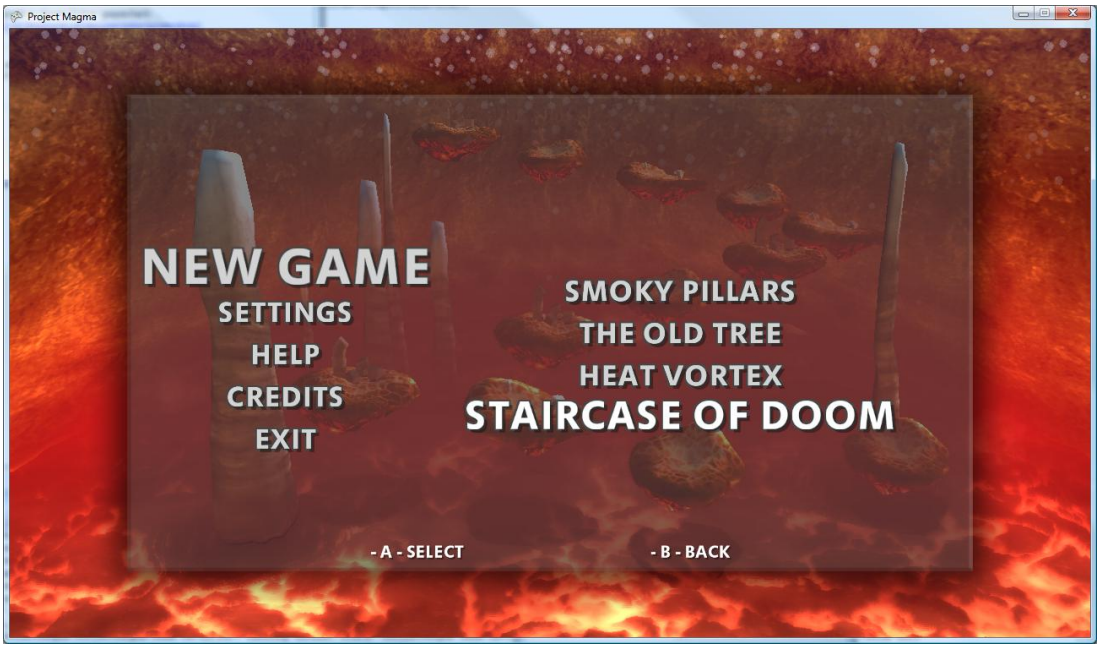
First of all I imagined the course to be quite different from what it actually was. This was mainly due to the advertisement presentations that have been hold at various occasions by different representatives of the computer graphics laboratory. For example I expected that there is a weekly meeting in the room with the large television to play all the games. But in fact this was never the case. It would be a cool idea to arrange such a meeting by the

organizers of the course. Even if it would not be mandatory I think a lot of the course participants would meet there, strengthening the cooperation in between teams.

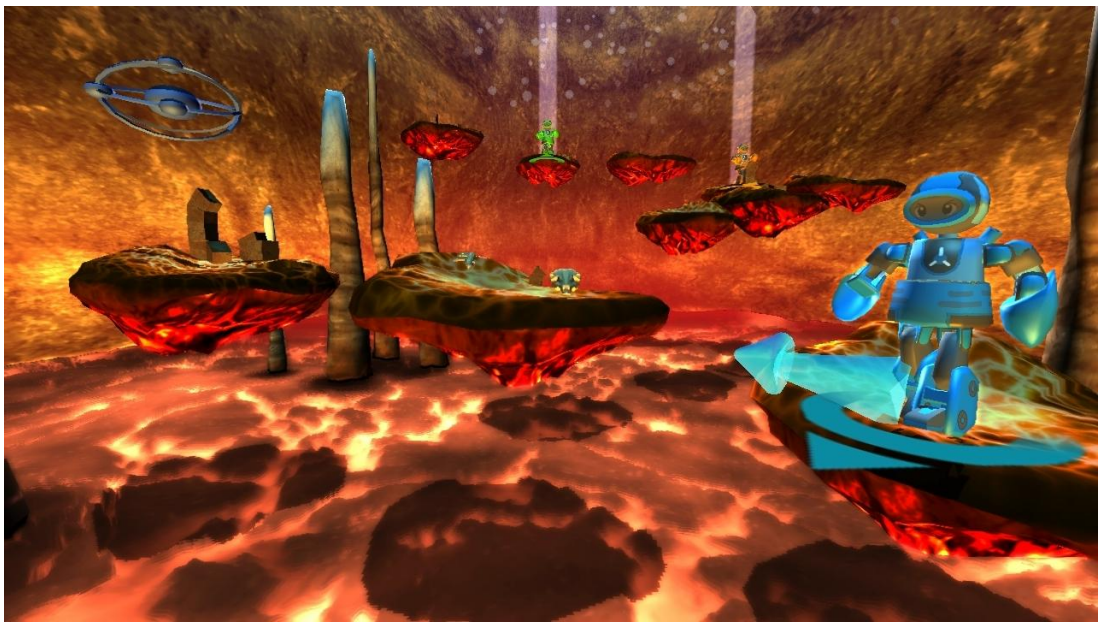
I also think that targeting the Xbox 360 using XNA is a double-edged sword. On the one hand it ensures that everyone starts off from a secure point. It is also quite motivating developing for the Xbox with a huge coolness factor. On the other hand it is also very restrictive. Some game genres like strategy games are nearly impossible to implement on a console. Personally I would have preferred to be open to choose the target platform depending on the game to be implemented.

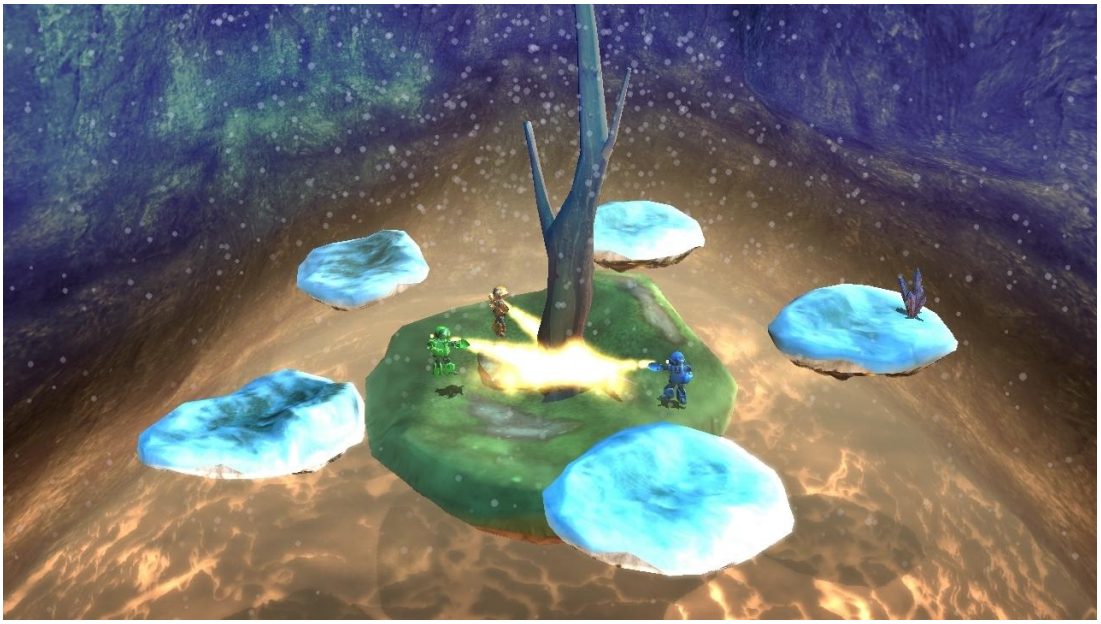
All in all we had sometimes an exhausting, but always exciting time developing the game. And most importantly: The result is really cool!

SCREENSHOTS

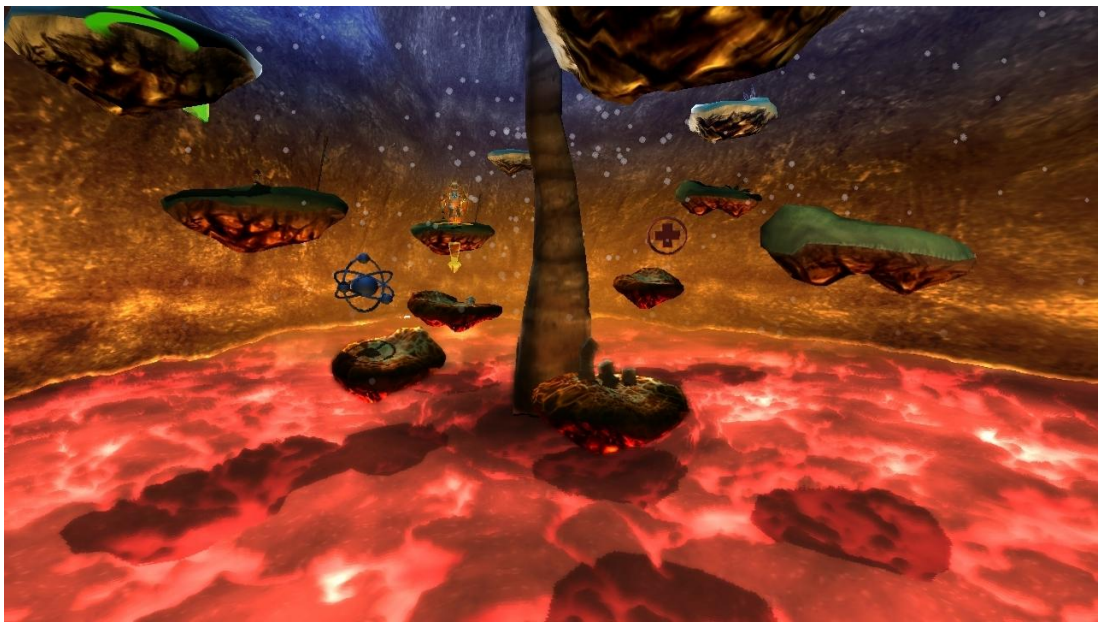


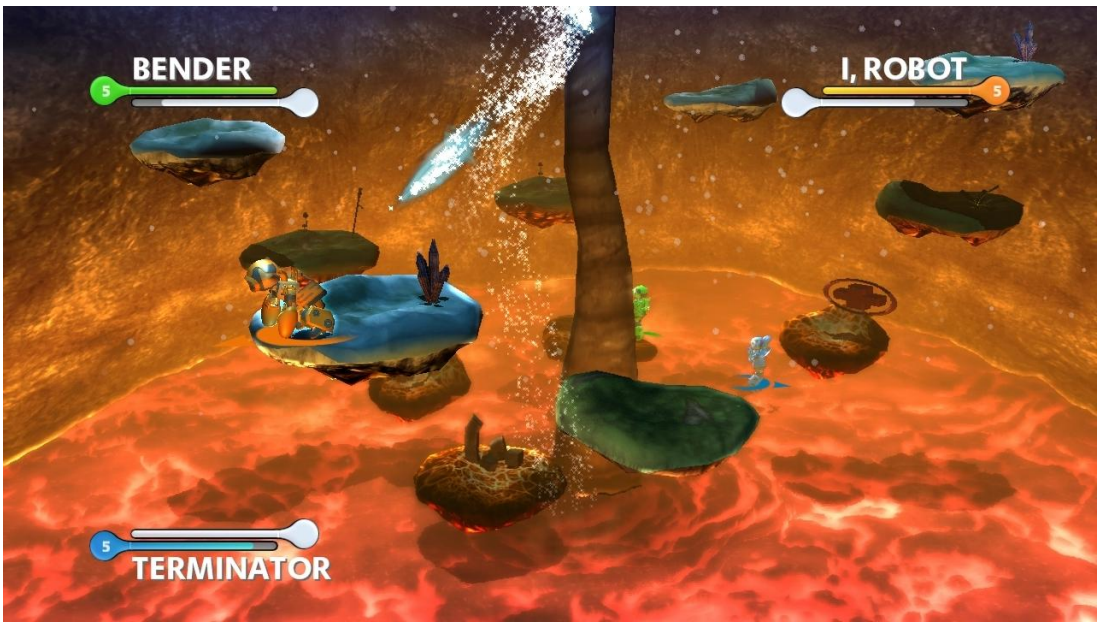

Project Magma
HIT START
TO JOIN
- A - SELECT
- B - BACK

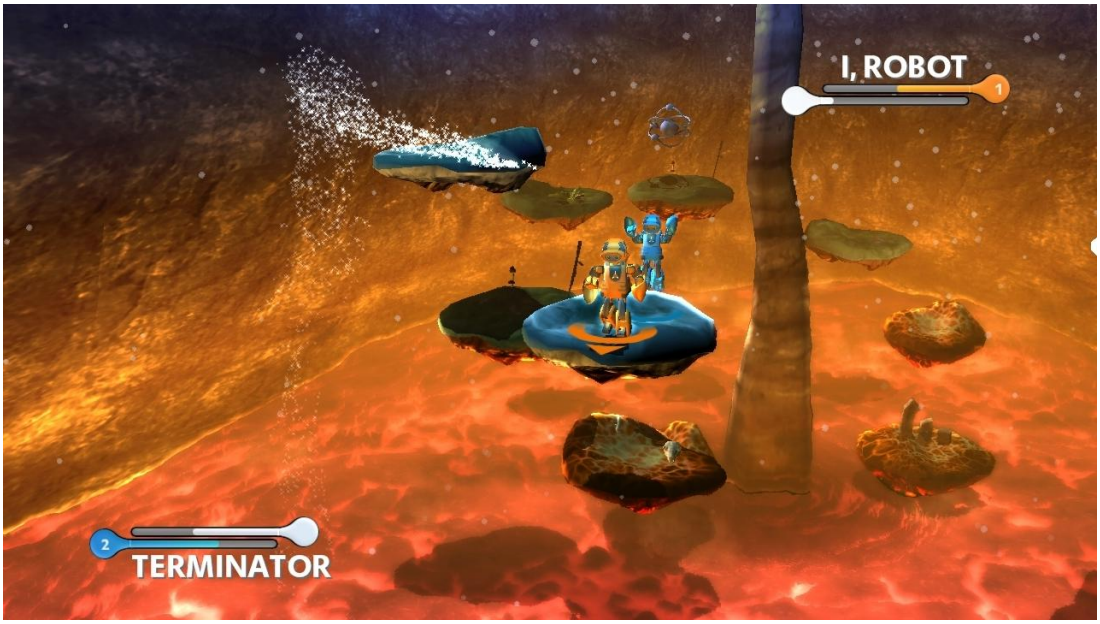





BENDER HAS WON!
CONGRATULATIONS!




BENDER
5
I, ROBOT
5
5
TERMINATOR


I, ROBOT
1
2
TERMINATOR