

SpaceRescue: A communicative Game for Interactive Floors

Daniel Birnstiel, Patrick Kuhn, Fabian Paul, Lennard Wolf

Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
{ daniel.birnstiel, patrick.kuhn, fabian.paul, lennard.wolf }@student.hpi.de

ABSTRACT

After looking upon the possibilities of interactive floors and taking into account the immense demand for video games today, we developed *SpaceRescue*. *SpaceRescue* is designed as a fun prototype for communicative, interactive floor based games and demonstrates, how playing games at home can once again involve moving the entire body again - and not just your thumbs.

The two to three players' goal is the joint reaching of levels by performing tasks that are randomly given to each player. These tasks can then be carried out by the player himself or he can tell his partners to do it for him since they are closer to the task-subject.

Author Keywords

SpaceRescue; Communication; Interactive Floors; Body Movement

INTRODUCTION

With the advent of the *Internet of Things* and thus the rising digital interaction with everything around us, floors will soon become intelligent entities just like our phones are today.

But next to all the productive things we can now do with our devices, we also want to integrate them into our leisure time. Just like touch screens revolutionised the way we play games, interactive floors will again push the boundaries of the way we think about enjoying ourselves through games.

To understand what players want from a game, we interviewed Jörg Friedrich, a professional game designer at AAA game development studio *Yager*, Willy Scheibel, a game development lecturer at *HPI*, as well as Theresa Zobel, a representative of our target group, the casual gamer. From these interviews we learned that:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from one of the e-mail addresses listed on the top of this page.

HPI '15, Jan 2015, Potsdam, Germany
Copyright 2015 ACM #...\$ 15.00.

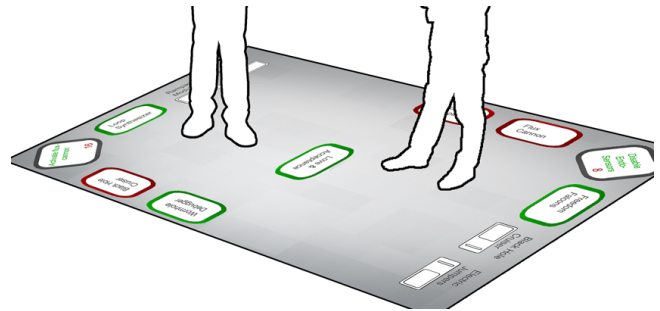


Figure 1: The Game in action.

- The game mechanics are the very most important part in the success of a game and should mostly be based on already proven concepts.
- Today's players don not want to read complex manuals and thus want to understand the game right away.
- Gamers always want to be challenged, but not to the point where they get frustrated.
- Next to the challenge, gamers also need a purpose for their actions. This might involve a story or the immersion in a virtual world.
- Pleasing graphics are nice to have but not overly important, especially not photorealism.
- According to sales stats, most gamers today like to play cooperative games rather than single player titles.

This is why we started out contemplating different mini games and at some point even considered creating a mini game collection. But what we saw as the main advantage of an interactive floor based game was the fact that the player might not be alone, but rather be in the same room with others and cooperate.

This is why we chose to concentrate on three main goals for our project: **Communication**, **Cooperation** and **Discoverability**.

The game is set on a spaceship, which is on the edge of destruction and the players on earth are the astronauts to save the ship. They have to get on the beaming platforms to beam onto the spaceship, where they get their tasks. These tasks involve flipping switches and pressing buttons, which make up the control bridge of the ship.

Since these widgets are not always part of the player's control stack, he will often have to tell his partner to do it for him instead. But a task is only active for a limited amount of time and if a task is not performed in time, it's game over. After a certain number of finished tasks, ladders will appear that lead the players to a new and harder level, where all controls will change and the game goes on. Winning as such is *not* possible, the motivation is rather staying alive for *as long as possible*.

WALKTHROUGH

In our sample scenario two users A and B want to play a game of *SpaceRescue* with the goal to reach the second Level. As the users enter the floor, they will start out in the *Beaming Area*, as shown in Figure 8.

The Beaming Area

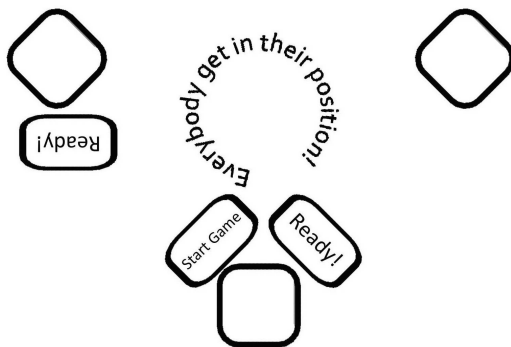


Figure 2: The Beaming Area from the bird's perspective.

1. Users walk to the *Beaming platforms* and want to start our game.
2. To start the game they have to enter one the circle while the label in mid is rotating *Ready?*-Buttons appear and will change the to a *Ready!* after tapping on them.

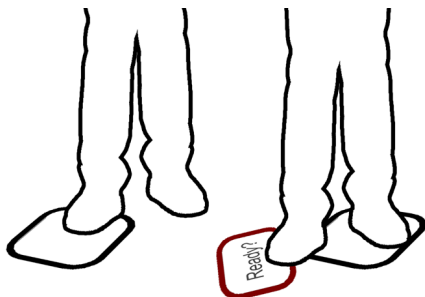


Figure 3: Steps 1 and 2a

3. The first player to step on a *Beaming platform* gets a *Start Game-Button* when everybody has tapped on their *Ready?-Button*.
4. That player can now start the Game by tapping on the *Start Game-Button*. After it the screen will change to our *Gaming Area*.

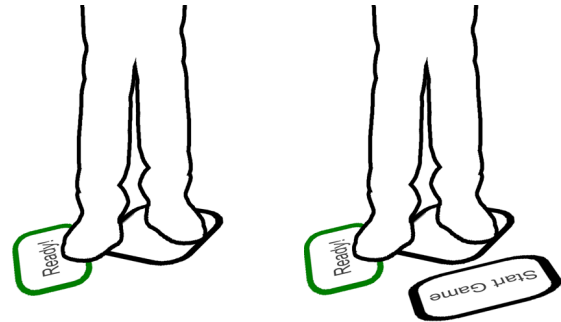


Figure 4: Steps 2b and 3

The Gaming Area

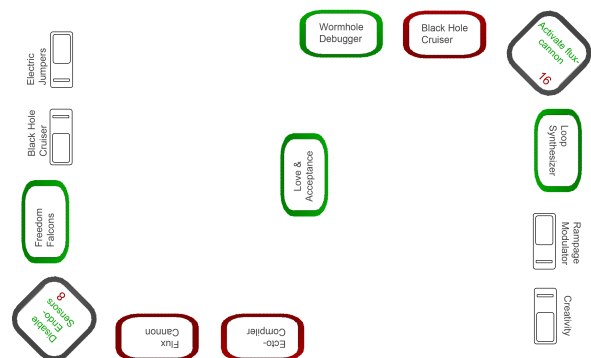


Figure 5: The Gaming Area from the bird's perspective.

1. Players A and B mention the two highlighted task panels behind them and turn around.
2. There, each player reads his instructions and sees their timer counting down.

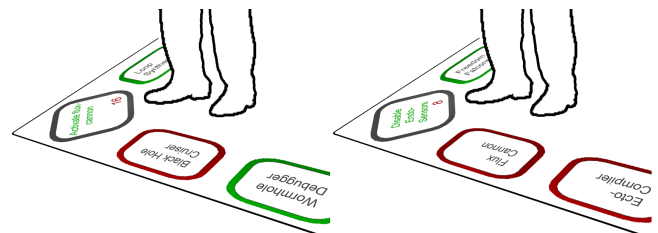


Figure 6: Steps 3 and 4

3. Player A gets task X (for example "Activate Flux Cannon" as shown in Figure 6).

- Player B gets task Y (for example "Disable Endo-Sensors" as shown in Figure 6).
- A and B try to perform their task by communicating or finding their widget.
- After a task has been performed before the corresponding timer has counted to zero, a new task is generated and the timer is reset.

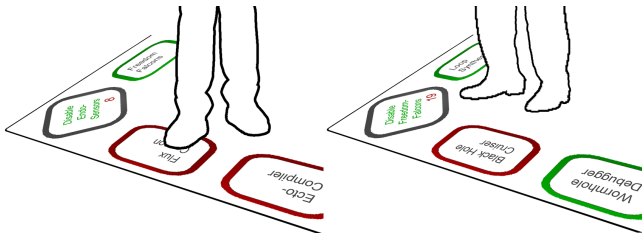


Figure 7: Steps 5 and 6

- Repeat steps 3. to 6. until a new Level is generated and the players have reached their goal.

DESIGN

In this section we will discuss the design choices we made to achieve our goals for **Communication**, **Cooperation** and **Discoverability**.

Cooperation as a Paradigm

Our first idea was to create an application which would consist of various *minigames*, which could be successively played against each other. The problem with this approach was, as Willy Scheibel pointed out in our contextual inquiry, that this approach would only allow users to interact with each other to certain degree. Playing by taking turns would not actually require the other player to be there, thus the whole concept of being together in a room vanishes.

Therefore we wanted to integrate cooperation as a paradigm for our game design, by assigning tasks to users, who cannot accomplish them.

Separating the floor into two areas to enforce communication

One of our basic game principles is communication between the users. To avoid the users from trying to accomplish the task by themselves rather than telling the other teammate what to do, we separated the floor with the graphic of rifts, where one can see the background of stars. When testing the design with users, we observed that some of them simply jumped over the rifts, because they perceived them as normal obstacles. To prevent this from happening, we added yellow "do not cross"-tapes on both sides of the rifts. This ensures, that each user only operates his own controls.



Figure 8: The "danger area" which separates the two control rooms

Avoiding roles as a Game Mechanic

TOOOOOOOOOODOOOOOOOOOOOO

Originally we came up with the idea of having different roles for each user, so that each would only be able to carry out certain tasks. As an example, the role of the *captain* was assigned to the first person entering a beaming platform. That user was then the one able to start the game and they would also have special rights to certain widgets and so on. In our paper prototyping process, most of the testers asked about the meaning of those roles and having to explain it to them in more than a sentence showed how this concept was hardly discoverable and would just *mode the user in*.

For these reasons we replaced the concept of roles with more specific tasks.

Explaining System Status with Conceptual Models

As we started designing our game, we used menus and text fields to let the user start a game or change a level. This approach seemed to work out quite well in paper prototyping, since the change of level for example would require some time to reconstruct the floor anyway. But as we started testing our application on the floor the change of status, especially the switching between two levels did not appear natural. As a result we had to change the core structure of our design to more modular areas: TOOOOOOOOOODOOOOOOOOOOOO

added areas with a ladder texture to explain the switch to the next level. Moreover we use beaming platforms as login mechanism instead of a introduction dialog to have a consistent appearance in our whole game

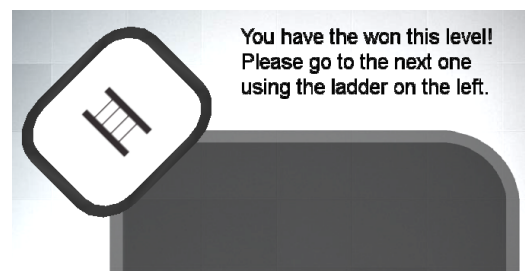


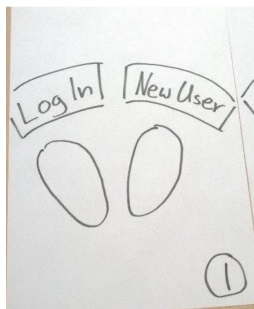
Figure 9: The message explaining the status with a ladder next to it

Standing in a defined Area as Login Mechanism

In typical PC or console games users log in by typing their user name and select start in the menu. Initially we thought about applying this concept to the floor by having the users register to the system with an on-floor keyboard and then letting them log in every time they enter the floor. During *paper prototyping* we encountered the issue that it is really tedious for the users to type their names, since tapping on small buttons requires precision and having the buttons spread makes them hard to use because it would be necessary to walk over them to get to the destination.

We decided to create areas in which the users have to stand to start the game. The paper prototyping and the heuristic evaluation showed that this was easily understandable, especially because of the conceptual model of the *beaming area* of spaceships from *Star Trek*.

Before:



After:

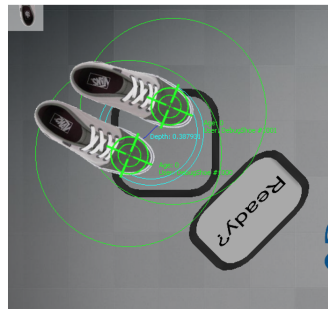


Figure 10: Two users logging in through the *Beaming Platforms*

Having a distinct Exit Button

When we faced the issue of how the game should be terminated, our initial solution was to just let the users walk off the floor and wait until the timer would run out. But during the heuristic evaluation, almost all of our testers asked us how they could end the game. This showed us that having no visual representation to use some functionality of the system is not at all discoverable.

To solve this problem we introduced a distinct exit button, which can be pressed to ensure a controlled ending of the game. This is important because now the user has full control over the system at any time.

Before:



After:



Figure 11: The *eject button* that pauses the game



Figure 12: After pushing the *eject button*, a siren starts off

CONCLUSION

Creating a game for a completely new Platform turned out to be a bigger challenge than first anticipated, since most classic gaming concepts were not applicable anymore. But talking to both gamers and the professionals in the field gave us great insights into what makes a good game. These insights made us create *SpaceRescue*, a game that we think of as really showing the possibilities for communication and involving of the whole body in Interactive Floor based games.