Project Flip Jump

Group 17

Mikael Ahlberg
Daniel Ericsson
Axel Stenkula
Johannes Svensson
Fredrik Vretblad

Who are the users and what problem does the system solve for them?
Our game idea is a 2D side-scrolling game which is based on a very simple idea. The player needs to survive by jumping between blocks and avoid falling down. To make it a little bit more interesting, the screen is moving and forcing the player to move.

Since the game is based on a very simple idea and has no attribute that excludes men or women, it has the potential to attract both. These simple games are usually spread through community sites aimed on a young audience. Because of that, our game is likely to be played by youngsters spending a lot of time on the Internet. We're expecting these young people to be in the age of 15 to 20.

To play this game no earlier game experience is required, but it's likely that our users will have some kind of gaming experience, because these kind of games are usually shared between friends who plays a lot.

We'd like to believe that our game is played in a short period of time. We're not expecting people to be playing this game for longer periods since other more complex games tend to capture players interest in a way that our game is not expected to do.
Since the game is focused on a short play-time, the game itself needs to be quick-started and throw the player straight into the action. Also, the player is probably already sitting by his or her computer when he or she decides to start playing our game.

The game itself doesn't solve any major problems but is a good way of making the time pass by quickly and also entertain the player. It will probably be played when you feel like relaxing and forget about your daily problems.

The main uses of the system
The game should be fast to start up and shut down and it will not be possible to save or pause a game. It should also be fun to play and challenging, therefore we will offer different difficulty levels.

Usage narrative one:
Dieter, a typical young German kid, is sitting in front of his computer and doing some serious homework when suddenly his friend Wolfgang sends him a link to a game through facebook (a community site). Dieter opens up the Internet-page, downloads the game, and starts it. A new window on his computer pops up where he can choose difficulty level. Dieter chooses easy since it's his first time playing this game. A quick tutorial/text is displayed on the screen telling him to jump on the space-button and move around with the arrow keys. He presses the s-button to begin the game. He starts to jump between the blocks in the game, and notices how the screen forces him to progress forward at a higher and higher speed. Suddenly the screen flips 90 degrees so he's heading upwards instead of forward. He was quite surprised and almost missed the next block. A bit later he missed the blocks and falls down outside the screen (due to the speed of the screen was too high, he wasn't fast enough). The screen “game over” turns up, followed by a new high score! He enters his name in the high score list and presses enter. When the game asks him ”do you want to play again?” he answers no and the game shuts down. He then returns to his serious homework.

Usage narrative two:
Kalle, who is a 20 years old Swede, is currently reading a quite hard technical text and really needs to take a break. He then remembers a fun game he plays from time to time (our game), and starts it up. He chooses the hard difficulty level, of course, since he wants to have some challenge. He begins to play directly (after pressing the s-button to start the game) and when the screen flips 90

degrees for the first time he's not surprised. Of course he can play a bit longer than Dieter, but since he chose a harder difficulty the screen moves a lot faster. When he gets “game over” he gets a new high score. After writing in his name and quitting the program he must let all his friends know about his new high score!!

The context/environment in which the system is to be used

The system will be used in a home environment for private use, because it's a game that we believe people will play during their off-time. Since most people today own a computer, this is the platform that we will focus on for our game. We will use Java to build our game because of it's portability on the computer platform. Because of this, users who would like to play our game must have some Java-runtime software installed on their system. Except this, the game itself will not require a high-end system for the game to be playable since it's a very simple game (construction wise). Because of the simplicity (size) of the game, it will also be a lot easier to distribute and for friends to share the game between themselves.

The scope of the system

Since our time is limited during this project, we have to have this in mind when we're planning and building our system. But as you can see in our scope-list bellow, we still have a reasonably amount of functions to be included in our game to make it fun and playable in our limited time.
The scope-list is structured in four different topics: graphics, sound, game rules and misc. We've done this because it becomes much clearer this way and because it's easier to structure our comprehensive work later on.

With “accelerated difficulty level”, we mean that the rate of which the screen is moving will increase during play time and the frequency of screen flips will also escalate with time.
Graphics scalability is the ability to decrease or increase the quality of the graphics in order to avoid performance drops and have a greater game experience, which is a functionality we won't implement.

In/Out list:

Topic	In	Out
Graphics		
- 3D graphics		X
- 2D graphics	X	
- Graphics scalability		X
- Resize-able window	X	
- In-game menu system	X	
Sound		
- In-game sound effects	X	
- In-game music	X	
- 5.1 3D sound		X
Game rules		
- Multiplayer		X

- Single player	X	
- Cooperative		X
- Save game		X
- Pause game		X
- Changeable difficulty level	X	
- Accelerated difficulty level	X	
- Changeable controls		X
Misc		
- High-score list	X	
- Tutorial text	X	

The main factors that need to be taken in to account when designing and building the system
When talking about games in general we always have the factor that people don't know how to play the game, a factor that's common for most computer based games. To solve this problem, most game developers sends an manual with the game to help the user get started. We will have a short tutorial in our game and an manual that comes with the game. This manual will not only explain the game and it's settings, it will also include a FAQ-list containing common problems that can occur during play, and how to solve them. To minimize problems and user frustration, we're planning to introduce the game to a number of test persons before release. Those persons will then give us important quality feedback so that we can adjust the system and manual according to these results.

Another important thing that we need to take into account is system requirements to run this game, with a reasonable frame-rate and game experience. We need to list both minimum hardware requirements and software requirements and what versions of software that's needed (Java version etc.).

We also need to document our code very well so the system can be modified and extended in the future. This is an important thing for us, because we need to be able to understand our system, both during the development stage and especially afterwards.

Technologies and Risks
The game will be developed for the Java-platform because of it's portability and our existent knowledge of the Java language. We're planning to use LWJGL, an open-source library that provides OpenGL (graphics) and OpenAL-support (sound) to Java in a simple and straight forward API. It's released under the BSD-license model and have reached a stable form. We've chosen to use this library during our game development because Java itself doesn't provide any good hardware acceleration for graphic intensive software.

Our problem here might be that our group has very limited knowledge about OpenGL- and OpenAL-programming, though our Java-experience is at a high level. This can lead to problems during the development stage but we are reasonable sure that we can handle this since we've studied courses that will make our learning progress of this material fast paced.

Another risk could be the closing of LWJGL or Java software that we are using to build the game. This is however very unlikely to happen, because Java is a major language today and is actually opening up their underlying source code as we speak. Since LWJGL is available under the BDS-

license it's already out in the open and everyone is free to redistribute the code with the license attached.

OpenAL have recently been widely accepted and is used in many new big commercial games, because it's free and open. It's therefor highly unlikely that they will close access to this software-library in the foreseeable future.

Other risks include changes of product specification and losing a member of the team, but both of them have a low probability occurring.