Feasibility Study

Brett Raible
Maike Keune
Maren Sandner
Nicholas Paquette

Software

The software our team will be using for this project includes the Spring framework for Java, IDEA, an IDE created by IntelliJ, and the JavaFX libraries. All members of the group have downloaded and tested Java Spring and IDEA on their own machines to ensure that they function together appropriately. One group member has had limited experience with JavaFX, while the others have never used it before.

We will use Maven to build our project and include external libraries such as Java Spring. Spring will help us with the program structure by providing an easy to use application context. We have tested if it can be used for our project (see code given in github: the app class will start the Spring application context and create a non-player character who says: “Hello, my name is Harry.”). The GUI of the project will be created with JavaFX (includes the control mechanics, sounds and 2D graphics).

Solutions

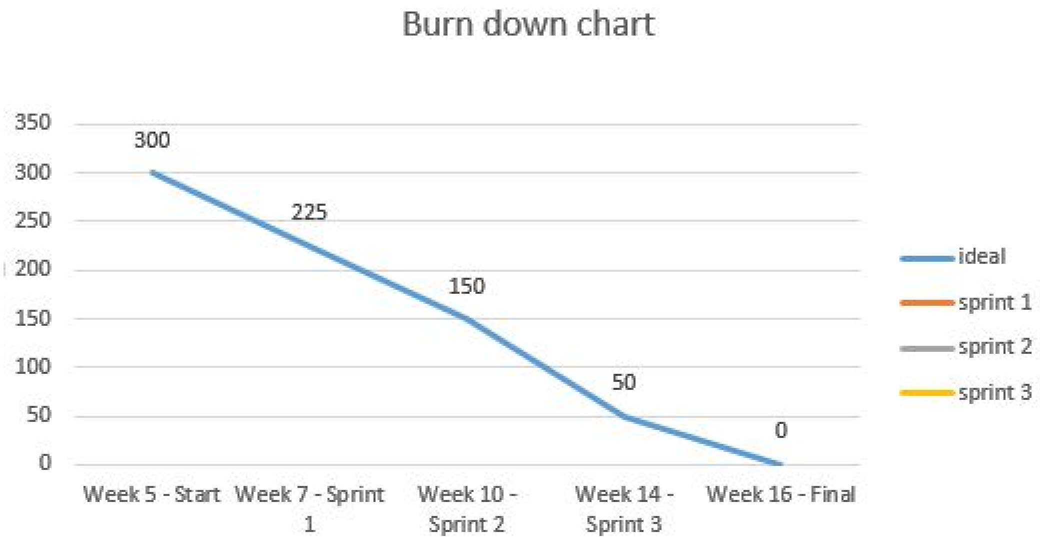
Our project presents several issues that our group will have to overcome, the randomly-generated overworld and dungeons being the biggest issue. One possible solution to this is to create small randomly-generated tiles that can be pieced together like a puzzle. These tiles will have to be flagged as accessible by the character to walk on or inaccessible because they represent a wall/forest/abyss. One tile represents one step of the character. In these tiles, it will also be determined if an enemy can be encountered on the tile (dependent on a percentage) and it will be randomly chosen what kind of enemy the character will have to face. We could also generate the random worlds through pre-made maps that get pieced together instead of just single tiles to reduce the odds of creating impractical worlds in which the character has no possible way to walk to an objective. Whether we use a tile by tile algorithm or a pre-made map algorithm will be decided while testing to see which solution has more potential.

The graphics and sounds will be provided from a program called “RPG Maker VX Ace” which both Brett and Nicholas own. This way, we don’t have to be bothered with creating them ourselves or searching elsewhere. We checked to see if it was legal to use the graphics and sounds even though we don’t use the software to code our game, and we determined that we could use these resources so long as we give credit where it is due.

Generally there will be three types of worlds. The overworld (which holds dungeon and town entrances), the dungeons and a town where the character can collect and complete quests (either through NPCs (non-playable characters) or a bulletin board) and shops to buy items with money. The overworld and dungeons will be randomly generated, while the town and shops will be static. The character will receive money for finishing quests and defeating enemies (amount dependent on the type). Buyable items will include weapons, armor, and health/mana potions, and possibly more.

The player will navigate one character through the game but when encountering an enemy, he or she will fight with at most 2 other characters in their party. All characters will have a class (mages, rogues or warrior) and can level through gaining experience from battles. Some or all items will be specifically for a class. For example, mages can use mana potions, staffs and light armor, while rogues and warriors can use stamina potions and either medium or heavy weapons/armor.

Burn Down Chart



The burn-down chart shows the ideal work in hours we should spend on this project. We're estimating a lot of work, so we calculated 300 hours in total to be on the safe side. There are 2 weeks between the start, the first sprint and the second sprint, where we will try to work 75 hours for each work period. Between the second and third sprint there are 3 weeks, so we will work around 100 hours. Between this and the final week we will probably have to work around 50 hours to get all bugs and small issues fixed.

For the first sprint we want to have all the actor classes with associated graphics and handling events and a static map for testing done. Also we should already make layouts for the dungeons and worlds. Parallel we will work on the world generator which should be done by the second sprint. The town (which should be static) and the file saving and loading system should be done as well. After the world generator works we can deal with usable items and the associated money system, the battle mechanics and possibly the story until the third sprint. After the third sprint, we should have everything major done and will work on all issues that we encountered while coding and testing.

Division of Labor

Work will be assigned to the person with the most interest in that specific topic and therefore thinks he/she is best suited for that specific topic.

Maren and Maike will work on the world-generator and the battle mechanics together as these require more time and they can work locally together. Furthermore Maike will work on handling events (for user interaction with the characters) and coding actor classes. Maren will also deal with creating quests as well as the experience system. Brett will work on everything regarding the GUI with help from Nicholas, who will also work on items with the associated money system and also the ability to save and load the game.

Topic	**who will be doing this?**
GUI (battle, main world, dungeons, shops, menus)	Brett & Nicholas
items	Nicholas
money-system	Nicholas
file saving and loading	Nicholas
handling event	Maike
actor classes	Maike
random-world-generator	Maren & Maike
battle mechanics	Maren & Maike
quests	Maren
experience system	Maren

Although we split the topics in this feasibility study, in reality everyone will try to work on every topic so we have a broad knowledge of how our project works. This is further detailed in our work policies.

Work Policies

The most crucial part in our work policy will be that everyone will receive weekly assignments that should be done at the end of the week or the week afterwards if a problem occurs. In weekly (or more often) skype meetings we will discuss our progress and if there are any new ideas, problems etc.

The created code will be uploaded in our repository and tested from everyone within 24 hours. This will help with fixing bugs as soon as possible. If someone wants to edit the code further (because they have a better solution or located a mistake) he or she should work with the creator to improve it.

We separated our Repository into two branches. For now there is:

- The master branch, which contains the source code, class diagrams and use cases, if needed.
- The documents branch, which contains the journals and icebreakers of each group member. It will also contain the feasibility study, sprint reports and the final report.

There will also be an experimental branch for testing new ideas before merging them with the existing project.

What is important is that we are all engaged in constant communication with each other and everybody understands the entire source code, otherwise this project will not be doable.