Software Workshop Team Java Proposal
'Giant Cow Games'

Joss Greenaway 0944297

Specification

As a team we have decided to work on a game which, for now, is being called 'Dark Matter', the basic concept of which being that both players will have control of a either a ball of 'Matter', spawned in different areas of a 'map' which contains randomly generated (Some levels less random with others with the inclusion of other elements for specific level challenges), the player then guides their 'Matter' by propelling out a small amount of it's mass, shrinking the matter and launching it in a specified direction, upon contact with other NPC matter or the enemy player matter, whichever of the two is the larger will 'absorb' the other, destroying them and taking on their mass to increase the player size, other pickups will be available around the map which will allow for the use of abilities which can slow down other matter, reduce other matter in size, increase speed, amongst other things.

In terms of Functionality there are a number of things to take into consideration, the first is that control will be done exclusively through the use of the Mouse, where you point and click to 'expel matter' which will propel you in the opposite direction from where you clicked, and the right mouse button will be used to active any power ups you happen to receive, being launched in the direction of the mouse, with the amount of matter expelled depending on the number of mouse clicks, and more matter expelled resulting in a greater velocity, but being a double edged sword as this will also reduce your overall mass, making you more vulnerable. Collision detection upon circle objects will play a major role, being that one of the key elements of the game is that when two matter objects collide, the larger will grow and the lesser will be annihilated. Depending on whether the player is in single player or multiplayer mode their objectives will differ, Single Player being a more relaxing experience where you want to absorb enough matter without dying to complete a level, with Multiplayer being competitive, where each player will be trying to grow in size quickly and collect power ups in order to take down their opponent, so they will be determined the winner.

In terms of GUI design we will be using Java SWING to implement our game, exploring ideas such as Double Buffering in order to give players as smooth an experience as possible, without having to dip into anything like OpenGL which would prove to be an obstacle. Players will play the game in full screen, with the objects on said screen scaled so that neither can see more of the map than the other, which could result in an unfair advantage to the player with the larger screen. A popup menu with options to Exit etc... will be available through use of the Escape key. There will be a colour based indication system where other matter will be tinted either Red or Blue depending on it's difference in size from the player's matter, being red if it is larger than the player, thus being a threat, but blue if it is smaller than the player, and therefore safe to absorb.

For our Release Plan we intend to keep it relatively simple with a new total version / package being presented on a week by week basis, we'll be fully utilizing Subversion to achieve this, ensuring that by a set day of every week we have a fully functional piece of code in our main directory, with the features we wanted to work on that week complete.

With Software Engineering and Testing as a group we looked at multiple different 'styles' of coding within a team, and decided upon the agile Extreme Programming method, focusing on three main ideas which were Pair Programming, which we have already timetabled, having at least an hour each week coding with everyone else in the Team, generally with the weaker programmer doing the actual programming whilst the stronger programmer explains and talks through the coding, which will allow everyone to be involved without anyone who may have a weaker grasp on certain concepts being left behind. A philosophy of testing as we code, rather than leaving it all until we have a large chunk of untested code and then running tests, this way we'll know all the bits and pieces function as they should right after they've been built, and we continue to test them as more code is integrated, guaranteeing that things are working together and we won't uncover some massive issue at a point where we're not sure where the error's coming from. We'll be working iteratively, deciding on a small amount of features we'll be working on individually / together, and throughout the week we'll work on these features, ensuring they're all complete and robust once the week is over, and this will ensure we're not working on lots of different things at the same time, ending weeks with lots of 'half done' features, so that we'll be able to build on solid foundations, rather than unfinished code. We'll be using Maven as a part of our collaborative effort as it has a great deal of very useful features which will really lubricate our coding process, such as writing jUnit tests for our code, compiling an API from all our Javadoc for us, generating a website from our project where we can view inconsistencies in style / errors that are present in the code / conflicts, etc... as well as a great deal of other handy features.

Networking is also something we've discussed, however we're tentative to make any outright decisions in regards to this due to the fact we haven't, yet, had any lectures on socket programming and haven't done enough research of our own to be sure what is or isn't achievable within the 11 weeks we have for this Team Java project. At the moment we're thinking that we'll be going with a Client - Server based networking system for our game, as from the sounds of it that won't be too much of a nightmare to implement, however as we learn more about networking from the lectures and further reading we've said that we may look at P2P networking as it would be an interesting route to pursue with some improvements for the players of our game, however whether or not this is or isn't a realistic pursuit we can't yet say.