Weekly log

Michal Staniaszek

Last updated February 15, 2011

1 [2011-02-01 Tue]-[2011-02-07 Mon]

1.1 [2011-02-01 Tue]

- Worked with Jere and David to discuss the class structure. Also created the top level abstract classes that we will use to create the game.

- Spent 30 mins trying to work out how JFrames work and how they get components added to them. Seems strange. Also thought about how best to do GUI elements. Would be good to have frames and panels in the same package, just because it makes sense. We could use multiple panels for the game, menus and so on, and add them to the main frame or sub objects when appropriate.

- Wrote a basic class for the main frame and tried to work out how the hell to test it, but didn't come up with anything.

- Started writing a panel on which to draw the game screen. This will need to have some sort of way of getting all of the objects that we need to draw from the game object. The game object should have references to all objects that are part of the game. Perhaps two arrays, ships and projectiles, will be good, but not sure if that's a good idea. Seems ok though. Another idea is to have everything that is part of the game as a subclass of some top level object (gameObject?), so that everything can be stored in one array. Again, not sure. Some experimentation will be necessary.

- Thought about how to do the mouse and key listeners. Most likely the best way is to have the control schemes subclassing their respective listeners so that we can pass those to the addlistener methods, which will supposedly make things easier. It might make it difficult to pass around parameters, but i guess since we're subclassing the listeners it doesn't really matter too much, but that's still to be seen. Otherwise, we just implement the mouselistener and keylistener in the main frame class. The problem of referencing can be solved by having references to the listeners in the frame, and having them chuck back values when something happens. This brings up another problem, though - how do we know when things have changed - we need another listener, which is silly. Might be best to do things the simple way first, and then try to abstract it later. Have no real idea how to solve this at the moment.

- THERE IS WAY TOO MUCH THAT CAN GO WRONG/BE HARD TO DO/BE ANNOYING! AAAAAAAGH!

1.2 [2011-02-03 Thu]

- Started work on the game class. This must integrate data from all over the place and then redistribute it to whichever class wants it. Will have to have a lot of methods to do updates on arrays, collision detection. Not really sure of the complete layout yet, since a lot of the rest of the code is still incomplete.

1.3 [2011-02-04 Fri]

- Thought about ways to solve the problem of fire rate of the vehicle. It shouldn't be too difficult to change the way that the player fires. There is an easy (hacky) way, where you use a count variable to check how many times you've entered the paintcomponent, and then only fire the weapon, say, every 20th time you enter.

- Mouse sensitivity may also be an issue. We may need some way to change the speed of the mouse's motion. A simple solution which looks promising is to take a percentage of the distance the mouse has moved and then modify player location to that value rather than the actual mouse location. This does mean that the mouse would have to be locked to the window, and also the player would run out of space, since you'd stop receiving new mouse input after a while because you reach the edge of the screen. You'd only be able to use a percentage of the window (the percentage of the mouse movement that you were using.)

2 [2011-02-07 Mon]-[2011-02-13 Sun]

2.1 [2011-02-11 Fri]

- Worked on pruning methods for the game class to remove units from the arrays when they are no longer needed. We should probably add some sort of flag to units if we want them to be able to move off screen and reappear somewhere else. This should require some sort of interface with the panel class, otherwise the game will have no knowledge of frame bounds. Could be done with an array pruning method in the game class which takes a width and height parameter, and removes anything from outside that box.
- 3 hours

2.2 [2011-02-12 Sat]

- Wrote JUnit tests for the Projectile class, and for the Game class methods that I wrote. Also wrote classes for use in testing - they take simplified parameters, just locations, so that things to do with location are easier to test rather than having to pass all the necessary parts of the constructor.
- 2 hours

2.3 [2011-02-15 Tue]

- Tried to work out how to do projectiles with Dave.
- Fixed having to create a new object each time you wanted to move the player around.
- Worked on drawing simple projectiles, by implementing a domove method to be called in the paintcomponent class to update the projectile's location, and then draw the updated thing.
- 3 hours