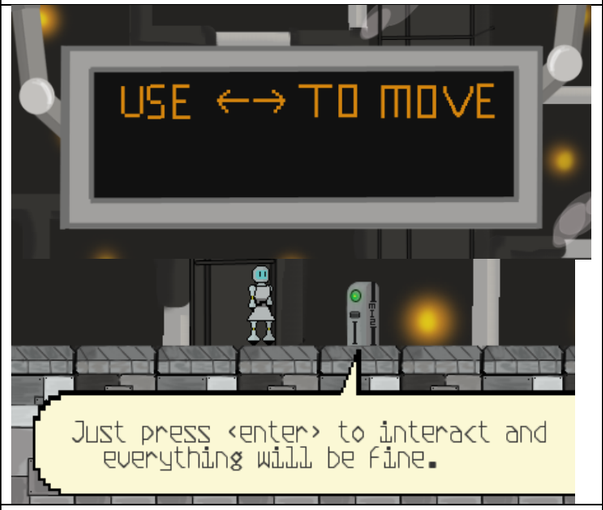
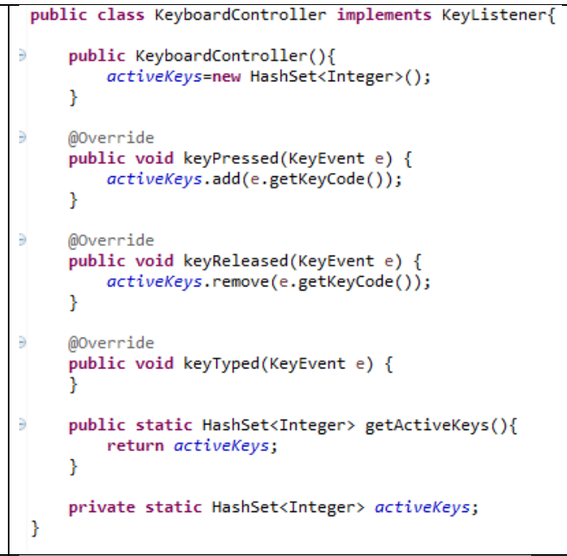
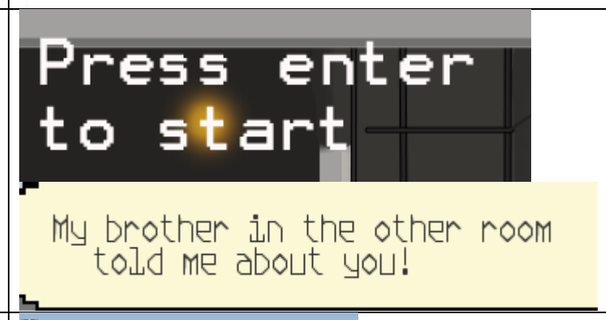
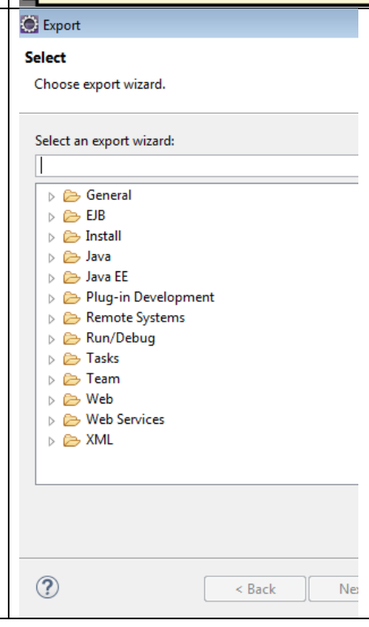
Describing and Evaluating the Usability Features of my Solution

"Usability means making products and systems easier to use, and matching them more closely to user needs and requirements.

Usability should not be confused with 'functionality', however, as this is purely concerned with the functions and features of the product and has no bearing on whether users are able to use them or not. Increased functionality does not mean improved usability!" - http://www.usabilitynet.org

Usability feature	Description	How well it is met	Evidence	Possible improvements
Instructions	This is the game's ability to ensure that the users are aware with the control scheme, mechanics and aims of the game.	The first four levels of the game are dedicated to a tutorial that is designed to show the controls of the game in different ways. The first stage has a big panel explaining horizontal movement and there are enter symbols floating above all the tutorial kiosks. The first kiosk gives the user some basic exposition; the second explains jumping, the third introduces batteries and the last one gives further information about interacting with objects. This should equip the player with enough information to reach the end of the game.		I have received one comment as feedback that one user did not interpret the symbols above the kiosks as the enter key. This would render the user unable to leave the first stage. While I have received this feedback and identified that this could a problem, I vouch for the design of the enter key symbol and will keep it as it is for now. Another possible improvement would be the potential to shorten the size of the tutorial by a level, as I feel that four dedicated levels are one too many, three would be an ideal size. The issue is that the tutorial is an important star of the game, meaning that it may have to be longer than I would have liked.
User Interactions	How the users are responding to the game. How long does it take users to accomplish a task? How are they performing these tasks? If they fail a task, how long will it take for them to recover?	As the game is designed as a method of exposing my target audience to my client's theme, it has been designed to no drastically slow down the user's exposure with difficult levels or gameplay. Each puzzle is simple, as it's what the objects represent and how they interact that is important, not their ability to slow down the player's progression. While most puzzles do not allow the user to fail, there are some points in the game that allow the user to fall into an endless pit. This will cause the player to respawn at the start of the level. As each level isn't too long, this will not set them too far back. This lightly punishing approach makes the game less stressful and more available to players of different skill levels.		As stated before, if a user attempts to interact with an object again, it will work, causing he user to get stuck in place. Ideally once an object has been interacted with it will lose every interaction. This bug doesn't cause any major problems but can cause issues with the smoothness of puzzle solving. While a part of me wishes that the puzzles were more complex and impressive, I do have to remember that the target audience is being exposed to concepts and ideas that they will be introduced to over the next four years- therefore it is important to keep the puzzles simple. On the other hand, my limited animation (and lack of delay in changing states in regard to dependency) does ruin some of the clarity of the puzzles. One of the major changes that would massively improve this game would be better animations, allowing for more detailed puzzles giving better explanations.

Control	How does the user interact with the system? How does the system react? Is it smooth or is there a delay?	The control scheme is very simple. There are only four accepted button inputs in the whole game to move the robot left and right, to allow it to jump and interact with selected objects. There is no delay in taking commands, meaning that the player will control the character in real time. This is important as some levels require a greater control of the robot in order to complete.		
Readability	Can the user read the text on the screen?	While most of the text is stylised (that being the font used in the speech of the robot), I have used large font sizes and contrasting backgrounds to make text as clear as possible in the game. So far, I have received no completes about the next used in the game, although I can understand that someone with dyslexia may struggle to read the spindly text used in speech.		
Accessibility	What devices does the game run on? Can users with disabilities play the game?	Right now the game can only run on a computer, but due to its simplicity, it can be easily ported to any other device (assuming it can handle the 200MB memory the game takes). As the game only requires four inputs, it can be compatible with most specialist peripheral that can be used in place of a keyboard, the game doesn't require the use of a mouse but may be hard to play if you cannot use both hands (although custom hardware can help that).		The application itself could be ported into a format capable of running on most devices. Software exists to allow this to happen, but at this point there is no reason for me to put it on any other devices. Maybe the biggest change I could make that would make the biggest difference in terms of user accessibility would be to have options to allow you assign custom keys to the inputs. This would allow anyone who has issues to play the game, as long as they have the specialist hardware. This would also be more accommodating towards users who are left handed as they can assign the 'wasd' keys to take the play of the arrow keys (a feature that some games have).

Speed	How fast does the game run? Is there any kind lag in the system when run?	When in development and opened from the IDE, the game has slight graphical lag for the first moments of the game, this passes and the rest of the game is smooth with no hint of lag.
		
		Currently, when opening the game from the runnable JAR file will play with some graphical lag. This does ruin the gameplay as it feels slow and slightly unresponsive.
			`//double buffering should improve animations`
			`this.setDoubleBuffered(true);`
			`}`
				There are currently two things that could be done in an attempt to counter this. The first would be to convert the JAR file into an application with all its source folders included, I believe that this will improve the speed of the game, but currently I am struggling to implement this change.
				
				The next idea, would be to increase the movement speed of the robot to compensate. While this is more of a 'quick-fix', it doesn't actually solve the problem, rather it looks for a solution around it. On paper this could work, by speeding up the game, you can compensate for issues on the system (the reverse of the artificial difficulty on space invaders). Although, as the speed is just fine on the IDE I am hesitant to change it.