Design for Joint Project 1

(Graphics and Programming)

This project requires you to make a game demonstrating most of the skills you have learned so far. You are encouraged to design your own game. Your project (game) **must implement the functionality (skills) asked for in this specification**. You should **talk** to your lecturer about your ideas for a new game in order to make sure it is suitable.

There is a **sample student game** in your common drive - Programming\Projects\Joint Project 1\Student Demo Project**.** Run the JointProject Application file. We would like you todesign a version of the example game or a game like it.

# It is absolutely prohibited to use anyone else’s code in your project. You can ask for help with a particular problem from friends, colleagues, the lecturers in the lab etc but you must code the fix to the problem yourself.

## Game Description

You don’t need to have lots of sprites in this game. For example you could have one robot, one bullet, one hostage and a few of each type of enemy. **3-6 sprites** on the screen at once is fine. You should have two enemy classes which represent two different types of enemy. One enemy class has different behaviour to the other enemy class. For example one enemy could follow the player and the other could move in a straight line, then change direction every 5 seconds for example. You could create a few enemies objects (sprites) of each type of enemy.

If you want more fire power in the game, you could implement a bomb and/or a missile and/or a beam. Your game should be able to regenerate the bullet and hostage and enemy. For example there should be only one player bullet on the screen at once and the enemy could come back again once killed.

Your game should get quicker/harder as the game progresses. Another feature your game should have is the ability to start a new game when the player is killed.

You must use use **Classes and Objects** in your Project.

**Extra functionality** is worth **30%** in the programming and graphics subjects.

## Submission

Sign the Plagiarism Declaration (in the project folder of your common drive) and attach it to your design document.

Design document due: **Wednesday the 16th of December (Hand up in your lab class).**

**Very Important:**You will be asked a few technical questions based on your design. You need to be available in the lab to answer these questions. If you are not **NO marks can be allocated** for your design. You should be able to answer these questions in a clear and concise manner. Failure to answer our questions adequately will result in further questions being asked and could result in low or no marks being allocated to your design.

This project design is worth approximately 25% of the overall Project mark.

**Your game should contain demonstrations of the following skills:**

## Sprite movement and animation

* move up/down/left/right
* sprite should point in same direction as it’s travelling
* fires weapon

**Example:**Player can move in four directions and is able to fire a bullet/beam; the player disintegrates when the player is killed.

## Collision detection

* Collision detection between game entities.

**Example:**- Player dies when it comes in contact with an enemy robot  
- Hostage destroyed when comes in contact with an enemy robot  
- Enemy robot dies if hit with player weapon

## Autonomous game entities

* entities that move in a straight line, then change direction every 6 seconds.
* entities move up/down/left/right by themselves towards another entity
* entities die when hit by player weapon

**Example:**-Hostage moves at a constant speed for 6 seconds, then changes direction.  
-Enemy robot moves towards Hostage.  
-Enemy robot destroyed when hit by player’s weapon

## Display GUI

* show lives left
* show score

**Example:**Show player’s lives left on top left, and score is shown on top right.

## Gameplay

* Must have a goal/goals and rules that make it fun (difficulty must be balanced).
* Your game should get quicker/harder as the game progresses.
* Your game should have is the ability to start a new game when the player is killed.

**Example:**  
**Rules:**  
Game over if you lose three lives  
Win game if you kill all the enemies  
Increase score if you kill an enemy  
Decrease score if you lose a hostage  
**Goal:**  
Kill all the enemies  
**Balance:**   
the speed of the enemies and the bullet makes it possible to win but at the same time it isn’t easy.

## Design

Print and hand up your design during your lab class on **Wednesday** **the 16th of December.**

**Part 1:**

1. **Game Description:** Write a clear description of what your game is and how you play it. You should describe the game as experienced by the user. It should not contain any implementation details (class names or algorithms etc). Use storyboards wherever possible to help your descriptions.You can hand draw the storyboards. It should be clear from your description and storyboard diagrams to someone who has never played your game before how it works. This part should be no more that 2 pages. See the below web link for information on what a storyboard is:

<http://www.dummies.com/how-to/content/designing-video-games.html>

1. **Game Genre:** State what game genre it is. Compare briefly to another well-known game. Obviously your game will be a lot simplier.
2. **Goals:** Describe what the player is trying to achieve (kill a certain number, survive, find exit, score a set number of points?). How does the player fail?
3. **Player:** How does the player move? How can the player kill enemies?
4. **Enemies:** How do they move, how quickly compared to the player. Move randomly or follow path or head to player. Do they fire bullets, how much damage do they cause, how much damage can player cause to them?
5. **Progression:** How does the game keep the user involved, how does it get more challenging?

**Part 2:** Also provide the class diagrams to show the classes and the relationships between them. You should show the **instance (class level) variables** and the **method headers only (no code)** in each of your classes. **Note**: You should give the **data type** of the instance variables too. Write the names of all the methods you think you are going to use in the class. It is **very important** that you give the **Game class** also. **Important:** You should **comment briefly** to say what the instance variables and methods are used for.

**Part 3:** Research how to do a part of the project which you **find complex and are unsure how to do it**. State clearly what part of the project this is. You should describe your solution in relation to this part of the project in a concise but clear manner. You should then give the pseudocode for the solution to this part of the project. The pseudocode could be for a single method or sequence of methods. State clearly what class each method belongs to and how your pseudocode links in with the rest of the project. You should state clearly for example how your pseudocode method (s) are called from the Game class and how they relate to the other objects in the project.