COMPSYS 302 Group One Project Plan

Group Background:

Sylvain is in his third year at the University of Auckland, working towards his Computer Systems Engineering degree. From COMPSYS 202, he is familiar with object orientated programming, Eclipse, and C++. Sylvain worked on Chatbot development as part of his practical work experience throughout last summer, actively coding in C# and following object orientated programming principles. He uses Git as a version control system regularly, whether to maintain the websites he created or for organising code-related, university group projects. Sylvain also has a personal interest in game immersion and the factors which impact on and result in a better (or worse!) game immersion.

Mark is studying a conjoint Computer Systems Engineering and Computer Science degree at the University of Auckland. Earlier this year he worked on a Microsoft Imagine Cup project in which his team created a Windows Forms application and a Xamarin cross platform app, both coded in XAML and C#. His team's extension features for the ELECTENG 209 project last year incorporated a real time graphing Android app, coded in Java. Both projects incorporated Git and object oriented programming. Last summer, Mark undertook a summer studentship for the University of Auckland's Department of Electrical and Computer Engineering, supervised by Dr Nasser Giacaman. The project involved designing and creating a selection of educational videos, a website, and lab exercises to help future engineering students familiarise themselves with software such as Ubuntu, C++, and Eclipse.

System Outline:

Our game will be based on the classic arcade game Warlords. The game will be able to be played by any number of one to four players. Each player will have a stationary base in a corner of the screen, protected by a block wall around it. There will also be a ball on the game screen, which will be continuously moving and bouncing off the sides of the screen and off the block walls. However, when the ball bounces off a block in the wall, then that block will be destroyed.

The aim of the game will be to defend your base, using two keys to control a paddle. The paddle will be able to deflect the ball away from your block wall and towards another player's wall or a side of the screen. If the ball fully penetrates a player's block wall and hits their base, then that player is defeated. A player wins the game if all other players are defeated within a certain time frame, or if they have the most blocks in their wall when the time frame expires.

A sample diagram of what each game screen will look like is included in the appendix to this plan. Our game screen will be square, with circular bases and paddle movement. Next to each base will be a timer and that player's score.

Also, included in the appendix is a plan of the menu layouts. The menu will allow the players to select game modes, additional features, options, and more. We are aiming for a simplistic and intuitive user interface, with only 3-4 keys required, and no mouse, for navigating the menu and playing the game.

Every instance of the game will always have four bases, block walls, a ball, and paddles. If less than four individuals are playing the game, then an AI player will be able to control the remaining bases. This AI will have multiple difficulty levels.

The basic game that we will implement will be a multiplayer arcade game. In addition to the basic game specifications as stated above, we will also include the following extra game features as shown on the next page.

Additional Features:

We want to add the follow additional game features:

- Single player / campaign mode with story, levels, and bosses
- Players can choose different classes with different powerups and abilities
- Game option to change ball speed
- Game option to change paddle speed
- Game option to make ball speed up on bounce
- Game option to have multiple balls
- Game option to enable paddle reverse physics
- Game option to have fire balls (penetrate further)
- Game option to have huge balls (does damage to multiple blocks)
- Game option to have temporary reverse controls
- Game option to make balls randomly bounce (and delay on bounce)
- Game option to make ball multiply on hit (makes single bounce bullets on paddle hit)
- Game option to enable chaotic mode (any selection of the above, or all the above)

Provisional Schedule:

- Friday Week 4 - Basic mechanics implementation:
    - Moving paddle
    - Moving ball
    - Bouncing ball
    - Wall destruction
- Friday Week 5 - Advanced mechanics and game menus:
    - Classes and players selection
    - Timer and Scoreboard
    - Win conditions
- Friday Week 6 - Additional game features:
    - Custom game options
    - Single player campaign
    - Music and graphics

Foreseen Challenges:

There are many foreseeable challenges which may be encountered. Some major challenges include the AI implementation in the game, closely following object orientated programming principles, and the creation of an original storyline which is appropriate to the game.

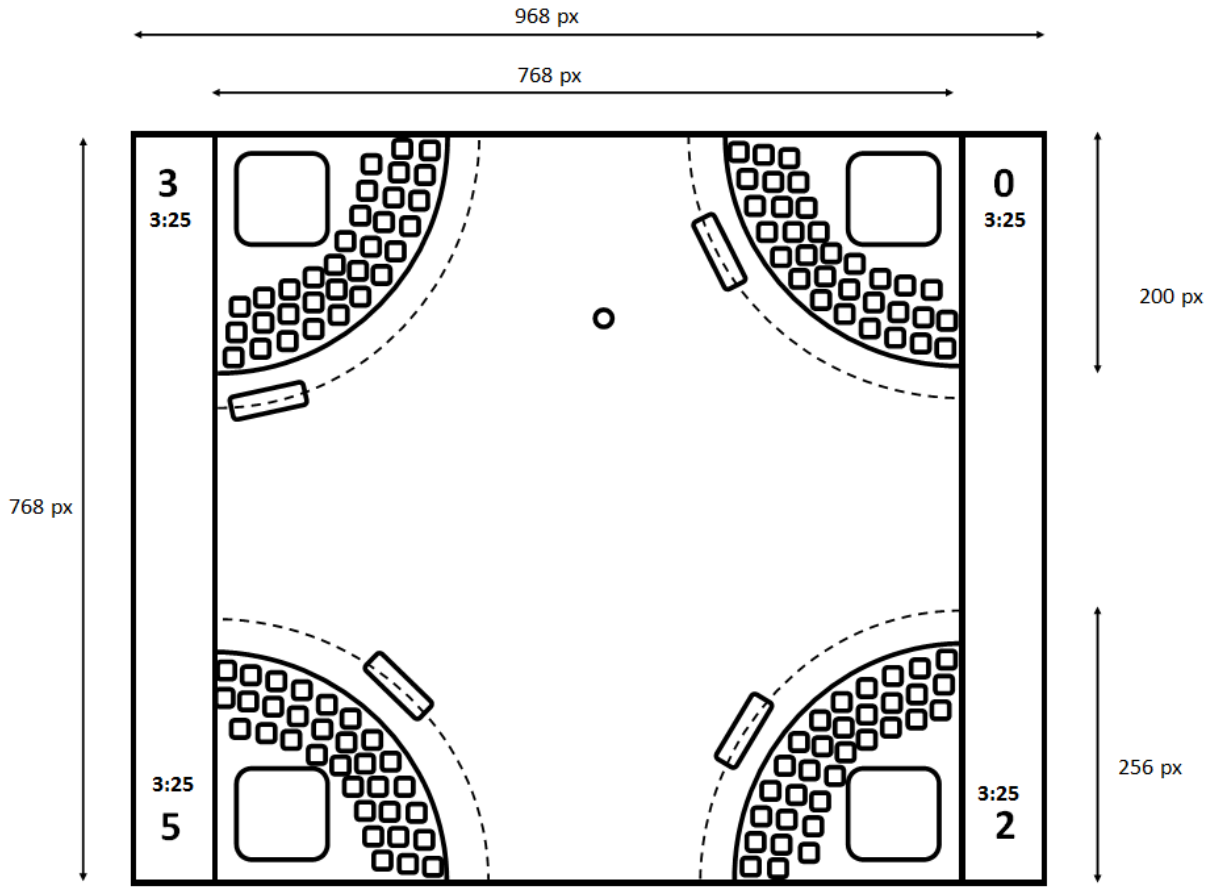
AI implementation may get complicated if we want it to be challenging for players of all skill levels, and not simply a basic AI following the ball moving on the screen. We want to implement separate AI for different difficulty levels.

Following object orientated programming principles will be more complex for students than it appears. This is because basic programming papers encourage learning the language and the syntax, rarely focusing on code efficiency and programming standards such as object orientated principles. Thus, students have had very few opportunities to practice following these coding principles, which explains why it may be one of the major challenges which will be encountered.

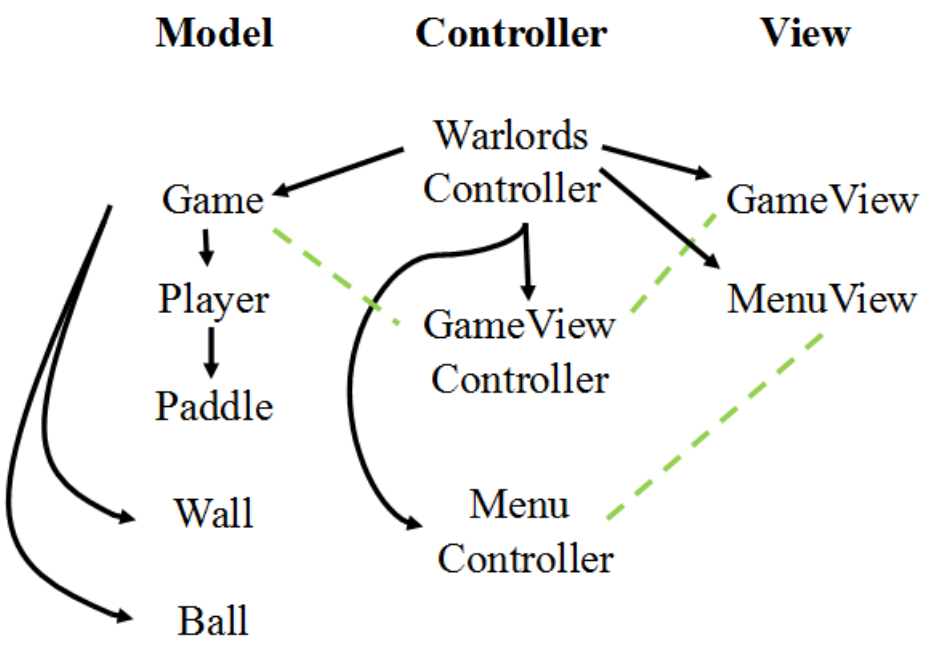
Finally, the creation of an original storyline relevant to the game may prove to be an issue. The game is very specific when it comes to the player number and game mechanics, and the environment never significantly changes. These are obstacles to creating a credible story which must encompass all these details. This will be especially difficult if the story is to be original and captivating to the player, while not reusing too many clichés or the storyline from the original game.

Appendix

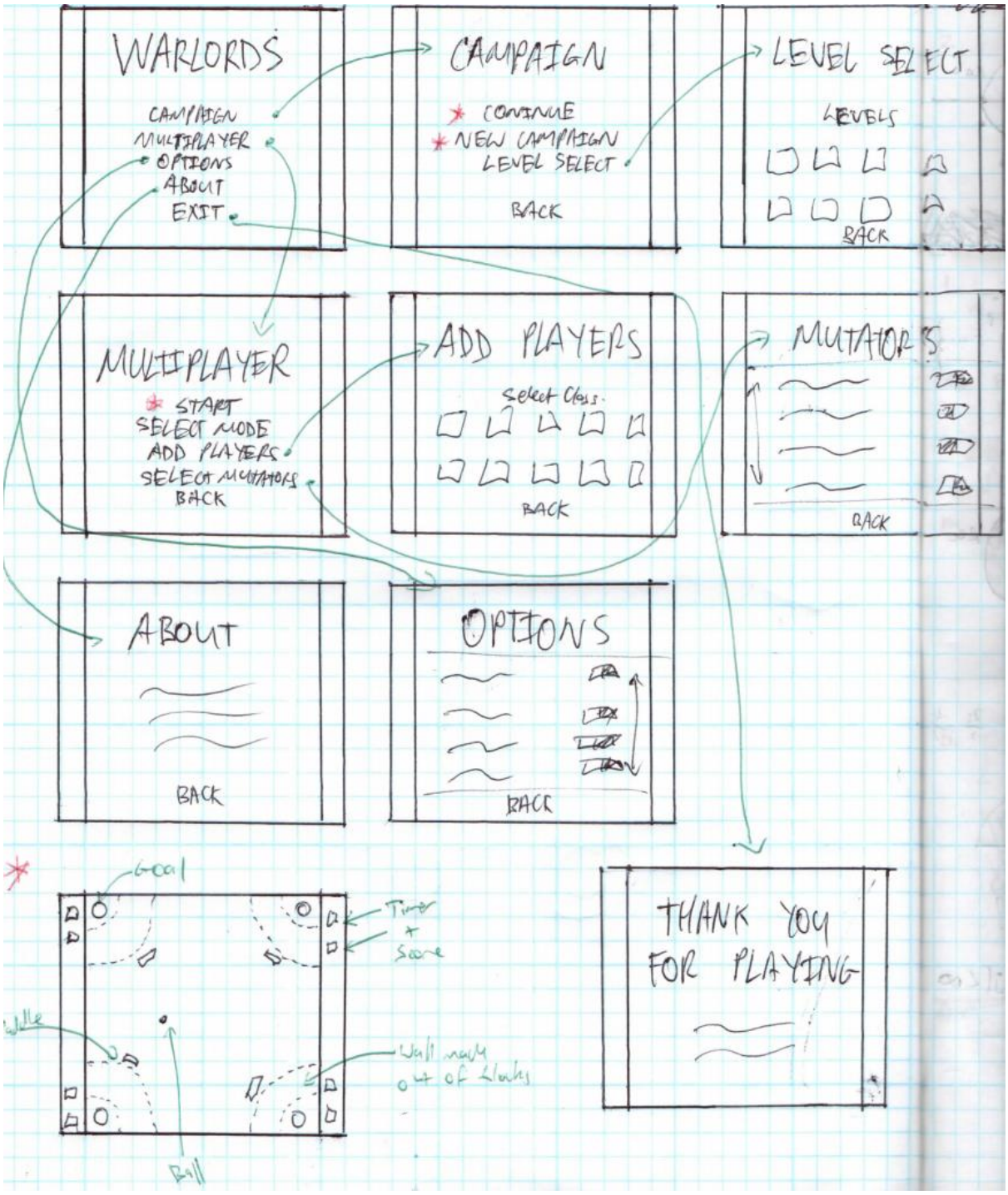
Game Screen Planning



Software Class Planning



Menu Planning



Player Classes and Abilities

Commander	Nation	Ability	Ability Type
Sylvain	France	Paddle can move anywhere	Passive
Shia	USA	Shoot an extra ball that doesn't bounce	Cooldown
Nigel	Britain	Immunity for a short period of time	Single Use
Andrew	New Zealand	Has an extra sheep paddle that wanders randomly	Passive
Mark	China	Extra layer of wall	Passive
Tony	Australia	Shoot poison spiders that slows other players down	Cooldown
Partha	India	Two paddles, slightly smaller, follow one controller	Passive
Vladimir	Russia	Steal a random outer block from other players walls	Cooldown
Muhammad	Egypt	Paddle turns into a large but stationary pyramid	Cooldown
Neymar	Brazil	Paddle can momentarily catch the ball	Passive