**Introduction**

I am going to be doing a plant vs zombie-based game however, the plants are going to be human artillery. This game will be a fixed map where you use soldiers and cogs as currency and fight the undead at night to stop your leader from dying. It is giving the

**Problem identification**

The game will start with a 6 by 9 square grid with grass over the top of the patches. The game starts by every 15 seconds the map spawns zombies randomly in each of the lanes. When you kill the zombies with the soldiers in the grid, your currency value will increase. This will then be recorded as a certain amount of currency in your bank which will be displayed as a number at the top of the screen. There will be a list of constructions and soldiers that are displayed. When you click on the soldier and you have enough currency to use it, you will be able to place the soldier in that space. This will be done by a drag and a drop onto the map. During the construction of your army, zombies will start walking towards your base across the 6 rows. There will be a list of different zombies that can attack you. Whilst all of this happens, there will be a score in the top bar of the screen along with a pause button. The timer will count down and when it reaches zero, zombies spawn a wave. Once the wave is dead, the timer starts again and the number of zombies will be increased. There is only 1 mode in my game called survival. It is basically a mode where you keep going until you lose. If a zombie goes past the last line, it will kill your leader and you will lose.

The reason that this game is good for a computational approach is that it is tailored to the gaming community which enjoy coded games. This game will approve of their desire for a new game.

The programming language that I will use will be python as although python has a limited graphical user interface, python is an object oriented software. I have only coded in python for a long time so the logic programming skills I have learnt on python will be applicable to the game.

| Feature | Explanation | Justification - |
| --- | --- | --- |
| Map | The map will be a 6 by 9 grid in the centre of the window. It will have squares that will have a fixed size and are able to place a square block inside it representing its weapon being used | The computer program would be able to generate a map in which the player is able to use in order to play the game. Since the map will be the same the whole time. The map is able to be clicked on after a click on the Hotbar to place something on the square. This can be done by setting an on-click listener as |
| Hotbar | The Hotbar will have a 1 by 10 of smaller squares that will be on the top of the map. The Hotbar will be able to interact with the player. It will be clicked on to choose a character. | Set an on-click listener to the Hotbar in order to click on the characters. This allows the player to build his army to win the game. |
| Score | The score will be displayed and incremented whenever you kill a zombie | This will then act as the score in which you high score is based on and will be stored in the user's file. |
| Pause button | This pause button ceases all actions. | All subroutines are paused and the game's timer is paused so as to not make the player lose. |
| The soldiers | There will be a range of soldiers but the basis of them is that they fire down the lane that they are in. the more they cost, the more damage they do. They can only shoot the zombie at the front of the line. | You can store all the information about the soldiers on a file where the stats like damage, currency and health can be imported. When they are placed down, they take their value away from the currency. |
| The zombies | The zombies will also vary in type. They will have 3 types of zombie: sprinter, normal and a tanker. Sprinters are fast with low hp, normal is average all around and the tank has low speed and high hp. The zombies are only allowed to go in single file so they will pile up next to each other. | This unlike the soldiers will be hardcoded into the game. This is because there are only 3 zombies whereas there are multiple soldiers |
| Login | My game will have a simple login with a username and password so as to save data. | The information will be stored on a file which will be unique for every player. |
| Controls | All controls are click based, including the pause button. | This will mean that the player is not able to input an invalid data type into the system and can only play the game and not break it |
| Currency | During the game, you start with 100 pounds. A unit in the game is able to give you currency. It is also impossible to remove this unit. As the game goes on, the unit earns you 25 pounds per piece | As you gain currency, the variable will add currency |

**Research**

During my time making the idea of the game, I thought of a few similar games that are like this. These include the “Plants Vs Zombies” franchise and a few others

1. Plants Vs Zombies (Original game)

Plants Vs Zombies is a single player game where zombies spawn to defeat your avatar. My game is heavily based on this and has similar if not identical traits. It was Published by Popcap in May 2009.

| Key features | Explanation |
| --- | --- |
| 1A) map | The map in PvZ is a fixed 6 by 9 map where zombies move down. The map is a 3D high angle view so the user can see and predict where the enemies will come from. This appeals to the strategy market as you are having to look at everything at 1 time. |
| 1B) combat | During the start of the game, the screen shows the zombies in which you will fight. This then allows the user to pick his team depending on the variety of the zombies. Once you choose your team, you start the strategy phase. When you begin, random drops of currency drop from the sky. This looks like there is at least a random aspect to the game along with the strategy. Then the player has to start by placing down units in order to do certain unique things to that specific unit. For instance, a sunflower makes more currency and the pea shooter shoots peas to kill the zombies. This is your input to the combat. During the game you are able to remove units in order to place a different unit in the original place. The plants are able to fire a shot straight down their lane so as to damage the zombie and fight them off |
| 1C) currency | When currency is picked up, the currency counter in the top corner increases by 25 points. When the game buys a plant, the currency variable goes down by the units value for instance a sunflower is 50 points and the pea shooter is worth 100 points. This happens for the whole time |
| 1D) enemies | After a certain amount of time, zombies come down the fixed lanes and head to your base which you are protecting. This means that the zombie wave has started to come down. Once that starts, the zombies walk down all of the lanes. This in turn makes the pea shooters shoot at the zombies. This then keeps going until the zombie dies and the whole system starts again. If they reach the end of the map, then the user loses |
| 1E) Hotbar | The hotbar is composed of the plants that you picked at the start of the game. It is a fixed length so if the user doesn’t fill the spaces it still takes up the required space. The hotbar is a drag and drop hotbar so the user can choose which unit that they want to use |

Weak points

Some weak points in the game are as follows: there is an achievements page which doesn’t serve a purpose to the game itself. The plants only shoot at the front zombie in the queue of attacking zombies, the whole idea of plants being able to kill zombies is unrealistic and there is limited graphics for the time of 2009.

Required hardware

The following hardware requirements are crucial to run PvZ: OS: Windows XP/Vista/7, 1.2GHz+ processor, 1GB of RAM, 128MB of memory, 16-bit or 32-bit color quality and 65+MB of free hard drive space.

Summary

Overall I like most aspects from Plants vs zombies. Its fixed sized map along with its fixed character placement makes the game more strategic than luck based. The way that the units only shoot down 1 lane at a time is also good as it means that the other lanes are independent.

1. Air Patriots



Air Patriots is a single player strategy game which uses pre-selected units in order to fight an onslaught of tanks. It is made by Lemon Games in 2012

| Key features | Explanation |
| --- | --- |
| 2A) Weapons | The weapons used in the game are realistic and are used in actual warfare which makes it a suitable choice for the theme for my game. You are able to drag your finger on the planes and they follow the track that you make. When they fly near the tanks, they use their ability to shoot at the tanks. It uses physics with the ricochet off of the wall and shoots the tanks to destroy them. |
| 2B) currency | The currency is made by destroying the tanks. After you destroy the tanks, it will be added to the currency variable and then is ready to be spent on the pre-selected weapons. When you click on the units to buy them, the plane will spawn and the value of the plane will be subtracted from the currency variable. |
| 2C) lives | If a tank gets through the attack of the plane and hits the player's avatar, it isn’t the end. The player has lived as to how many tanks it can let through. This is then counted at the end of the game to give a rating out of 3 stars. |
| 2D) Extra features | The game also has 2 buttons. One of which is a pause button which makes the game freeze and one button is a fast forward button. This speeds up the game exponentially. Also the game has a war type feeling because of the features and images of the weapons and other stuff like dialogue and images |

Weak points

The map is fixed but it is a winding road where the tanks can out maneuver the tanks. I think this is a disadvantage because the ships need to have their own radius on them when they fly over the tanks in order to shoot them instead of having a fixed position they can go. Also the ships are able to go wherever they want. This means you can have an unlimited amount of ships and the strategy part of the game is taken away.

Hardware requirements

Must have a device with either the play store or the app store. 80mb of memory available and a reasonable amount of RAM.

**Stakeholders**

Throughout my project, I will be communicating with my team of people that I am relying on as my stakeholders. These people are as follows:

1. Zak Serroukh - This student is a game fanatic, who is the ideal target audience for my game. He is old enough to know other games that mine is based off and yet young enough to have an interest in video games
2. Amandeep Sahota and Rhys Roberts - These computer science students will advise and critique my efficiency and my methods of creating my game making them an essential stakeholder in my project
3. Patrick O’Grady - This Student is an Art student and will be criticising my art and the layout of my game. Like Zak Serrouk, he is also a fanatic for games and is important for them to advise me on the gameplay and suggest different ideas.

**Proposed solution**

**Essential features**

From the research I have made, I am going to be doing a single player build and boom game. The game will consist of 1 gamemode and will have fixed positions with all of the weapons and other features. My research helped back up my ideas for a fixed map and hotbar size and a singular mode of play only. All seperate parts of the program will be stored as a subroutine

| Feature | Proposed Computational approach | Research reference | Stakeholders comments |
| --- | --- | --- | --- |
| Map | The map will be a fixed size and colour so I can use the map for every time the game is loaded to save time creating unique maps. Since it is a fixed size, it is able to be stored as a 2d array. | 1A | “Great choice for a computing standpoint. It will be easy and can be programmed concurrently” - Amandeep Sahota |
| Hotbar | The hotbar will be a fixed size with fixed units. this can also be stored as an array. This make the information easier to use and access | 1E |  |
| Timer | The timer will be a built in module which we can use in the code. |  | “Makes the game more enjoyable as you can see your progress” - Rhys Roberts |
| Delete button | This is a new feature presented to me by my stakeholders. As you progress through the game you need a way to remove old units so there will be a .pop on the current 2d Array | 1B |  |
| Pause button | This will be an interrupt where all subroutines will have an if statement where if called on, will play the subroutine of the pause button. The subroutine will be a while loop where if the |  | “ This allows you to go and do other things like get a drink or go somewhere else” - Patrick O’Grady |
| The soldiers | The soldiers are going to be stored inside a file which can be accessed for the hotbar. Then the soldiers will be stored in a 2d array when | 2A | “ Cool theme for the game” - Zak Serroukh |
| The zombies | These can use the random.randint to select a zombie out of 3 |  | “Good idea to use actual well known enemies for your project” |
| Login and sign up | The login will use a unique sign up method where all usernames will be unique and the username will be the name of their file. All usernames and passwords are going to be stored on a file |  | “Nice idea to keep other people's data safe |
| Controls | The controls are just a mouse click so we can use a program to wait for an “on click “ subroutine | 1B | “I would prefer this to another control scheme as you only need to press 1 button or \*click\*” Zak Serroukh |
| Currency | The currency will be increased over time and it will be stored as a global variable where the game can add and deduct values for each purchase of the weapons | 1C, 2B |  |

**Limitations**

One limitation to my idea is that there won't be physical projectiles that the player can see. This would merely be an add on in which the user would prefer whereas the game would work just as well without it. This prevents wasting time on an abstract idea.The second idea is that there will be a restricted number of playable weapons, this is because the game would require a system where you can change and select your units. This is because the hotbar where all of the units are stored is a fixed length, inturn making the size of the project too large a workload. The third idea to limit would be level based gameplay. This is not needed as the levels would just be a limited timed event identical to the survival mode. Another limitation is the software coding language python. Even though I chose a language that I understand, it is a language that has limited options when it comes to the user interface. Therefore this restricts me by having better graphics.

**Project requirements**

The computer must have access to the software for java and therefore needs to be connected to the internet. It must have sufficient storage in order to store all of the files (along with the python software) and it must have a keyboard and a usable mouse.

**Success criteria**

ZS - Zak Serroukh

AS - Amandeep Sahota

RR- Rhys Roberts

PG- Patrick O’Grady

| Spec Reference | Success Criteria | Justification | Research Reference point | Stakeholder comments |
| --- | --- | --- | --- | --- |
| 1a) | When the game loads, it must be able to display all of the different weapons on the hotbar. | This is so when selecting the weapons, the user can click on the hotbar to select them. | 1E | This is great as you can actually play the game - AS |
| 1b) | When the user clicks on a soldier and drags it onto the map, it should spawn a soldier down. | This allows the user to defend themselves against the zombies at all times . | 1B | You can now play the game by fighting the zombies - RR |
| 1c) | The soldiers must be able to fire at advancing zombies. | This allows the user to advance during the game and gain points and currency. | 1B | This is an essential part of the game in which the player needs to fight the zombies.- RR |
| 1d) | The soldiers must be able to take damage if a zombie is attacking it . | This makes it so that the user must think strategically as to where they place their soldiers. |  | Again this gives the game a play or die kind of feeling. -RR |
| 1e) | The soldiers must be able to fire in a certain range designated beforehand . | This allows a difference in games to PVZ but also gives the weapons more variety and allows the guns to be more realistic. | 2A | This gives some diversity for the soldiers and makes the user feel that they are in a realistic war zone -PG |
| 1f) | The soldiers need to be deleted when they are called back with the walkie talkie. | This is so the user can customize how they want to layout their soldiers. |  |  |
| 1g) | If the soldiers health is less than or equal to 0 then remove them from the map | This is to make sure that the user isn’t invincible and can’t lose health. | 1B |  |
| 1h) | The soldier only fires at the front most zombies. | This gives the user again another challenge and also doesn’t allow for multi kills. |  | ZS - this makes the game more interesting and realistic as in real life you wouldn’t be able to shoot more that 1 at a time. |
| 1i) | When the soldiers fire, a BANG will appear above their heads. | This is to indicate to the user that the unit is firing at the zombie |  | PG- I like this as it gives a visual description of what your soldiers are doing. |
| 2a) | At a certain time, the zombies must spawn into the map. | This is so that the user has an enemy to face and also to earn more currency. | 1D |  |
| 2b) | At a certain time, the zombies must be able to statically move forward. | This is so the user has a chance of losing otherwise there is no point to the game. | 1D |  |
| 2c) | When adjacent to a soldier, the zombie must damage the soldier. | This is so that the user is given a challenge and also so that the zombies aren’t just wandering down their lane without any way to deal damage. | 1D | PG- Good |
| 2d) | If a zombie reaches the other side of the map, the game must display “game over”. | This is the zombies objective and therefore needs to be able to end the game once it has completed that objective. | 1B | ZS- This has to happen otherwise there is no point to the game. |
| 2e) | If the zombies health is less than or equal to 0 then remove them from the map. | This is the same as the soldiers. If they don’t have health then you can’t kill them so you will eventually lose. | 1E |  |
| 2f) | When spawning in a zombie, it must be randomly selected as 1 of 3 types. | This is to give the game variation and to test the structure and integrity of the user's soldiers and design. | 1D |  |
| 2g) | The zombies slowly increase in spawn rate. | This is so that it makes the game more and more difficult as it goes on. | 1B |  |
| 2h) | The zombie must deal more damage if the difficulty is hard. | This is so the variation of easy, medium and hard is distinguishable. |  |  |
| 2i) | When the zombies are getting shot at, they should display a “UUR” above their heads | This indicates to the user which Zombie is being shot at. |  |  |
| 2j) | When the zombies deal damage to the soldiers, the words “BRAINS” appear above their heads | This is to indicate that they are dealing damage to the soldiers |  |  |
| 2h) | As the game progresses, the zombies must deal more damage and move quicker. | This makes sure that the game doesn’t ust |  |  |
| 3a) | The currency must increase when a zombie dies. | This is so you don’t run out of currency when playing and also to add it onto your current money. | 2B |  |
| 3b) | The currency must decrease when a soldier is created. | This is so the user can’t keep putting soldiers down as they would have infinite currency. | 2B |  |
| 3c) | The score must increase with every kill made. | This is so the score to the user is different to the currency meaning it starts at 0 and increases as you kill zombies. | 2B |  |
| 3d) | The pause button must be able to pause the program until clicked again. | This will give the user a break and to maybe strategize about what to do. | 2D |  |
| 3e) | The program must check if the score they have got is a high score | This is because if it is a high score then save it to the user's file. |  |  |
| 3f) | The program must display the “game over” text box along with the user's score. | This is to indicate the end of the game for the user. | 1E |  |
| 3g) | The game must display all of the text for the cost of the soldiers and the currency. | This is so that the user knows the cost of a unit and the amount of current currency it has. |  |  |
| 4a) | The user must load into the main menu screen once either they have lost the game or they have logged in. | This is so the game doesn’t just finish and gives the user a chance to do different options like to play again. |  |  |
| 4b) | The user must be able to choose whether they want to play a new game. | This will be if the user wants to play the game and then takes them to the next page. |  |  |
| 4c) | The user must select a difficulty to play the game. | This is to be set as the damage multiplier and allows the game to run differently to the other games. | 1B |  |
| 4d) | The user must be able to view their own high score | This is a button which when pressed will output the highscore saved in their file. |  |  |
| 4e) | The user must be able to log out | This is a button when pressed will take them back to the login page. |  |  |
| 4f) | The program must display a picture of a war behind the main menu page. | This sets the theme for the game along with some visual effects. | 2D |  |
| 5a) | The program must first load into the login page and display. | This is to find out which of the user's is playing the game before continuing. |  |  |
| 5b) | The user must enter their username and password. | This is verification as to if the user is the user . |  |  |
| 5c) | The user must be able to click on the sign up button. | This is in case the user is a non pre existing user. |  |  |
| 5d) | The user must be able to enter all of their information to create a new account | This is so the program can access the data further on in the code. |  |  |
| 5e) | The user must be able to exit at any time. | This is done by pressing the X in the top right of the window. |  |  |

**Input**

| Ref | Success Criteria | Justification |
| --- | --- | --- |
| 1b) | When the user clicks and drags a soldier. | So the soldier spawning can be identified . |
| 1f) | When the walkie talkie is selected by the user. | So the program knows to delete the soldier. |
| 3d) | When the pause button is pressed by the user. | Pause the action and stop all functions. |
| 4b) | When the new game button is pressed by the user. | Load into the “select difficulty” page. |
| 4c) | When the player selects the difficulty it wants. | Load into the main game carrying the damage multiplier. |
| 4d) | When the highscore button is pressed. | Display the user's high score. |
| 4e) | When the logout button is pressed. | Load and display the login page. |
| 5b) | When the user inputs their username and password | Check the passwords and usernames are correct and load the main menu page. |
| 5c) | If the sign up button is pressed | Display the sign up page. |
| 5d) | If all of the information is pressed and the user clicks create account | Update the user file and create the file. |
| 5e) | If the X in the window is pressed | Close the game or any operation. |

**Output**

| Ref | Success Criteria | Justification |
| --- | --- | --- |
| 1a) | The game must display the screen at the start of the game | This is so that the user can see what is happening on screen. |
| 1b) | The game should spawn the soldier | This is to show that the soldier is there and it’s cost has been taken away from the currency. |
| 1f) 1g) | the soldier is deleted from the screen | This is because the soldier has either lost all of its health or has been called by the walkie talkie . |
| 1i) | The soldier displays “BANG” above their head | This is to indicate which soldier is shooting |
| 2a) | The zombies spawn in | This is because the zombies have to actually start walking down their lanes. |
| 2b) | The zombies move statically | This is because it is easier to store the zombies in a 2D array otherwise. |
| 2d)3f) | The game over sign is displayed. | This is because the zombie has reached the other side and the game has finished. |
| 2e) | The zombies have been removed | This is due to them running out of hp. |
| 2i) | The Soldiers display “”ARR” | This indicates when the zombie is attacking which soldier. |
| 2j) | The Zombies display “BRAINS” | This indicates when the zombie is attacking |
| 3a) | The currency increases on screen | This is due to a zombie dying. |
| 3b) | The currency decreases on screen | This is because it has been spent on a soldier. |
| 3c) | The score increases | This is due to a zombie dying. |
| 3g) | The text is displayed at the start of the game | This is to show the cost of soldiers and current currency . |
| 4a) | The main menu screen is displayed | This is because either a game has been lost or the user has logged in. |
| 4d) | The high score is viewed | This is because the high score view has been selected. |
| 4e) 5a) | The login page is displayed | Either because the program has been run or the user has logged out. |
| 5e) | The window has been cancelled | The user doesn’t want to play at that moment. |

**Aesthetics**

| Ref | Success Criteria | Justification |
| --- | --- | --- |
| 6a) | Each of the weapons are a different color | This is to distinguish which soldier is which |
| 6b) | Each zombie is a different color and a different shape than the weapons | This is to clearly distinguish between the zombies and the weapons |
| 6c) | The map is coloured in dark shades of green and light shades of green like a chess board | This is to clearly show which space the zombies and soldiers are on |
| 4f) | The main menu screen is an image of a war/zombie like setting from World War Z | This is to get the user's understanding of what the game is about. |

**Processes**

| Ref | Success Criteria | Justification |
| --- | --- | --- |
| 1a) | The game loads with all of the correct functionality | This is to make sure all of the components of the map are working |
| 1b) | The game finds the position of the click down and the click up and spawns it in here. | This is to figure out which soldier is spawning |
| 1c) | The game scans the map and runs through a lot of if statements. If the zombie is in range then it will take its damage away from its health. | This is so the zombies' health is able to be taken away. |
| 1d)2c) | The game scans the map and if the zombie is next to a soldier, it damages it. | This is to also ensure that the zombie is damaged. |
| 1e) | The soldiers must be able to fire in a certain range designated | This gives the weapons more variety and allows the guns to be more realistic |
| 1g) | The game scans the map and If the soldiers health is less than or equal to 0 delete them | This is to make sure that the user can lose health. |
| 1h) | The soldiers fire at the front most zombie | This is to ensure a balance in the game as opposed to multi kills |
| 2a) | At a random time, the zombies spawn in | This gives the user a target and a way to increase their score |
| 2b) | The game scans the map and if the zombie is supposed to move, its x is changed from a multiple of the width. | This makes the zombies a lot more menacing and also so the zombies come down 1 by 1 |
| 2d) | The game scans the map and looks if there is a zombie in the last slot of the map. | This is to signal the game over flag |
| 2f) | The game randomly spawns in different types of zombies | This is to give the game some diversity. |
| 2g) | The game slowly increases the spawn rate using a variable that changes with a slow exponential. | This makes the game increasingly harder and therefore makes the user have to try harder |
| 2h) | The game carries a variable over from the main menu page and uses that as a damage multiplier if the game has a hard setting | This is to make the games more or less difficult. |
| 3a)3b) | The currency increases and decreases when a zombie dies and a unit is spawned | This is so that the user doesn’t have unlimited amounts of money |
| 3c) | The score for the user must increase if a zombie has been removed/died | This is to give the element of earning in the game |
| 3e) | The game checks if the score is greater than the high score stored inside the user's file. | This is to make the switch of the high score and the score if it is needed |
| 4b) | When the user chooses to play a new game, the game must load all of its aspects. | This is so the user doesn’t have to wait for the computer to load straight away |
| 5d) | The user must be able to enter all of their information to create a new account | This is so the program can access the data further on in the code. |