Get Ready For a Blast of a Game With...Hawaiian Punch

Senior Project Report

# 

|  |  |  |
| --- | --- | --- |
| Author(s): Adam Holan and Akshay Samuel |  | Date Started: 5/31/2019 |
| Course: ICS3U1 | Last Updated: 6/19/2019 | Date Delivered: 6/20/19 |

1. **Customer Requirements**

The client, Mr. G.Reid, is a computer science teacher at St. Francis Xavier Secondary School. His requirements for the project include 4 aspects: design, implementation, documentation and testing and deployment. In the design aspect of the project, the customer requires the use of HLD such as UMLs to plan out different classes needed for success and Stepwise Refinements to brainstorm different subprograms, and LLD for proper flowcharts of subprograms used. In the documentation component, he would like proper and effective use of commenting in the code to convey a clear message of what exactly each piece of code does, and he needs proper spacing of code. In the implementation aspect, Mr. Reid needs to see efficient use of code which utilizes everything we have learned thus far; selection, repetition, sequence, variables and there should not be any useless code. Finally, in the testing and deployment component of the project, the client needs the entire game to run without any errors. In addition to this, Mr Reid requests that all games be Hawaiian themed.

**Project Timeline**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Requirements** | **Design** | **Implementation** | **Testing** | **Deployment** |
| Proposed | May 31 - June 2 | June 3 - 6 | June 7 - 12 | June 13 - 15 | June 16 - 18 |
| Completed | May 31 - June 4 | June 4 - 8 | June 4 - 14 | June 14 - 15 | June 16 - 19 |

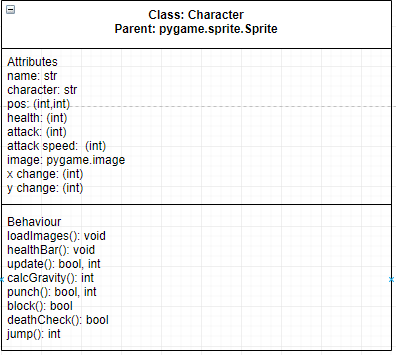
It is worth noting that design and implementation had a few days crossover. This is because in the early stages of implementation, we thought up an idea to use fruits to make it more “Hawaiian Themed”, and adds a fun new element to our game. Otherwise, the dates went more or less as planned, but of course personal events and errors in our code ate up a couple days and pushed back development of our Game.

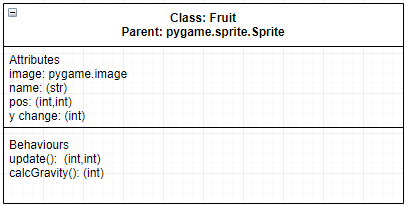
1. **Design Proposal**

The program will be created using the Pygame module which enables us to use features such as sprites, classes and objects, bitmapped graphics and collisions. In addition to this, we will be using pixlr; a photoshop software, to crop and resize certain images which will contribute to the aesthetic of the product. On top of this, we will be using the website programarcadegames.com to learn how to use the pygame module effectively.

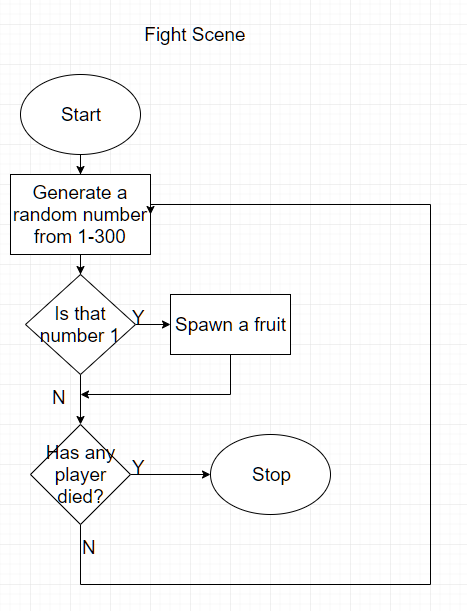
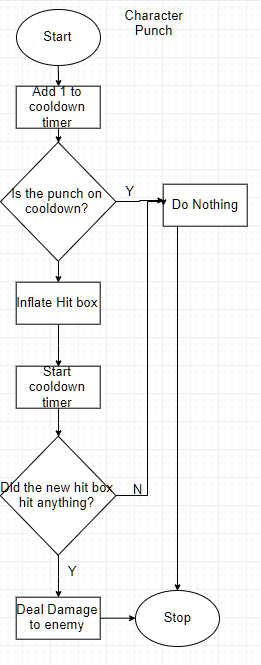
In our project, team GReid will use everything we have learned throughout Mr. Reid’s computer science course to efficiently program this game. Our program will be a remake of the game “Street Fighter,” which means we will try to include specific aspects of the game such as having characters punch, block, jump and shoot projectiles at each other. This will all be done through the use of classes.

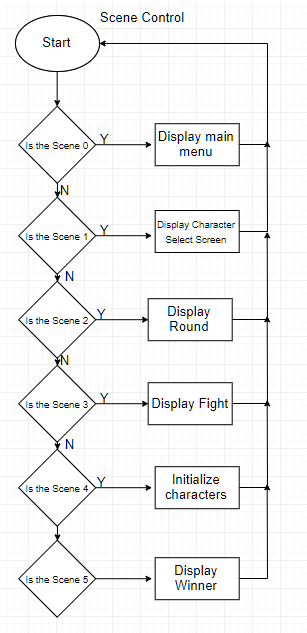
The UMLs below will show each specific class with the relating behaviours and attributes.





These flowcharts show the flow of some of the more complex logic:





1. **Implementation Details and Deliverables**

We have created our final product using everything learned in the course thus far. We have used object oriented programming thoroughly during our time forming the product. These classes allowed for us to define different objects and give them specific attributes and behaviours. Also, all of our classes inherited from the sprites class which made it easier to implement collision into our game which was vital because our game revolves around fighting other characters.

Inside the classes we used and outside, team GReid utilized a lot of repetition in the form of while and for loops. An example of this is when we used a myriad of for loops to check for collisions. Aside from this, we would not be able to complete our product without the use of selection. Specifically, we needed to use selection for many scenarios involving scene selection, boosts/health reductions due to collision and key movements. Also, we used lists and pygame.sprite.Group() function to draw all the objects in our game. Finally, the use of variables to store specific data was crucial in creating the product.

A method Team GReid employed to create a “character select screen” is something that many games do not do, which is an interactive “menu” of sorts where you choose a character with a base character. Our base characters were *Street Fighter’s* “Ryu” and “Ken” and these were originally meant to be placeholders; just sprites we would use for testing. However as development lead on and Team GReid realised it was not the greatest at creating graphics, we grew fond of these characters, and decided to keep them in along with Barack Obama, who hails from Hawaii. The way the “charblocks” (as-called in the code) work is the instant you make contact with one, the player loads the character associated with that block.

The Fighting consists of the characters being able to perform a few actions that any good fighting game should have. Player 1’s controls are AD to move, W to jump, E to punch, and Q to block, while player 2’s controls are JL to move, I to jump, O to block, and U to punch. The characters can of course punch to deal damage, and block to reduce damage taken on them by a considerable amount. In early development, Team GReid found that out of the players we selected to try out and give feedback on our game, all of them decided to run at each other and punch until one died. Initially, this spawned the idea of being able to block and jump to dodge, however no one utilized those since punching lead to a surefire victory. Ultimately, it allowed us to come up with the idea of fruits: power-ups that fit our hawaiian theme perfectly. Their attributes are simple: Mangoes give attack damage, bananas allow you to attack faster, and coconuts damage you, however they do not linger on the ground. It is also worth noting that characters switch directions when one jumps over the other, which can be quite difficult to keep track of if both characters are the same, however to a player this feels fluid and is hardly if ever a problem.

1. **Maintenance / Testing**

Over the duration of Team GReid’s project there were many moments of adversity which we overcame through diligent teamwork and communication between one another. Furthermore, We believe this was one of the many strong points of our group. One major challenge we overcame was the idea of keeping the version of our product up to date for both partners. To combat this issue, we utilized the online resource GitHub which allowed us to simultaneously work the project together without error. Aside from this, team GReid did an exceptional job of allocating and organizing different tasks. To avoid error, we assigned tasks by classes. For example, Adam would work on the character class, while Akshay would work on the fruits class. This led to a smoother process of coding the product.

A critical challenge we faced was with the design component of the product. We did very minor design which only included the use of UMLS for our classes. Additionally, we left our flowcharting till the end, and tried implementing before planning which inevitably led to a variety of errors in the program which include:

1. Fruits: This was the first glitch we ran into. When fruits were first added to the game, after the first couple fruits fell, there would be multiple glitch images of the fruit constantly popping up when it would not provide any power boosts. While Team GRied did some code optimization to mitigate this glitch, it still occurs should the game run on too long, however this minor bug can be considered a feature, as it appears as though the game has entered a “hell mode”, providing players with an anxiety-inducing flurry of fruits that is sure to make the last few seconds of any round heart-pacing.
2. Character movement: The second glitch we experienced was related to the movement of the character. The user would be able to move their character normally until this glitch arose where their character would only move in one direction. This was an extremely weird glitch which we did not know how to fix because it varied from computer to computer. One computer would experience it, while another would not. For this reason, we were not able to fix this glitch.
3. Character Jumping Off Screen: One niche and very easy to miss bug was if two characters were at the edge of the screen, it would clip both of them out of bounds and down into the eternal abyss below, where the only escape was to quit. This was easily solved in maybe 5 minutes by expanding the platform by about 100 pixels off screen on each side, just to be safe. This fix had the added benefit of clipping the characters back into position because of the way Team GReid wrote our code, which was a very nice surprise.

Overall, after reviewing all these glitches, one thing we could have improved on was better design planning and time management. With the use of flowcharts, some minor glitches could have been avoidable and time management would allow us to fix any outstanding errors. Aside from this, we believe team GReid has surpassed the client criteria with our product.

1. **Resource Allocations (for groups only)**

|  |  |  |
| --- | --- | --- |
| **Task** | **Responsibility** | **Description** |
| Mr. Reid | Project Template | Constructed a project report template for the students to use to complete final report. |
| Akshay, Adam | Report Management | Making updates on the report during the allotted time. |
| Akshay | Design of UMLs | Handled the design of UML elements to be implemented into the code by both members of Team GReid. |
| Adam | Co-level design and implementation | formulating design ideas for the project and implementing it |
| Akshay | Co - level design and minor implementation | formulating design ideas for the project and implementing smaller tasks |
| Adam | Design of Flowcharts | Handled all the flowcharting for the project |

1. **References**

You must clearly cite any references to material which does not represent your own work on this project. For example if sections of code were written by someone else (perhaps the teacher gave you some of the solution) you must clearly indicate what sections represent their work. There is nothing wrong with leveraging other people’s work, so long as they are given appropriate credit and you are no attempting to claim their work as your own. This applies to code segments, algorithms, diagrams, graphics/images and any written documentation.

Hawaii Bird’s Eye (Main Menu Background):

https://matadornetwork.com/destinations/north-america/united-states/hawaii/

Beach Background: https://www.feedbackcasino.com/games/spinions-beach-party

Sprites for Ryu and Ken:

https://www.spriters-resource.com/

Mango:

https://purepng.com/photo/7881/food-mango

Banana:

https://purepng.com/photo/14134/food-bananas

Coconut:

https://purepng.com/photo/14241/food-coconuts