Progress Report
- Increment 1 -
Group #7

1) Team Members

Please write the name of all the team members, their FSU IDs, and GitHub IDs here.

-Mason Russo | mer21i | M-russ09

-Will Andrews | wja21 | TheWillAndrews

-Tori Lee | tel21b | torilee7935

-Parker Thompson | pjt21| PJThompson715

-Jacob Sonn | jrs20k | jsonn32

2) Project Title and Description

-Pixel Pummel is a 2-D retro-style street fighter combat game. Players will step into the shoes of our unique fighters, each with their own moves and style. Our fighters will battle through multiple environments facing off against other players of varying difficulty AI. It provides an action-packed experience for all players to enjoy.

3) Accomplishments and overall project status during this increment

Describe in detail what was accomplished during this increment and where your project stands overall compared to the initial scope and functionality proposed.

-In this increment we implemented the game window along with the backgrounds images that will be used throughout the duration of the game. The project is still in its first stages and still needs to advance in the fighting functions along with keeping score/health of the fighters.

4) Challenges, changes in the plan and scope of the project and things that went wrong during this increment

Please describe here in detail:

- anything that was challenging during this increment and how you dealt with the challenges

- any changes that occurred in the initial plan you had for the project or its scope. Describe the reasons for the changes.

- anything that went wrong during this increment

-A major challenge during this increment is the overall inexperience with pygame and attempting to learn how to incorporate the distinctive features needed to make the game fully functional. Luckily, there is a plethora of outside resources like YouTube, ChatGPT, google search, etc. for assistance in learning how to utilize the different assets within the foreign language.

-Another challenge seems to be trying to implement unique components while not wasting too much time. It is easy to conjure up a feature that will improve the game, but very easy to waste time, go down the rabbit-hole, and forget the main game functionality.

5) Team Member Contribution for this increment

Please list each individual member and their contributions to each of the deliverables in this increment (be as detailed as possible). In other words, describe the contribution of each team member to:

a) the progress report, including the sections they wrote or contributed to

b) the requirements and design document, including the sections they wrote or contributed to

c) the implementation and testing document, including the sections they wrote or contributed to

d) the source code (be detailed about which parts of the system each team member contributed to and how)

e) the video or presentation

a) Parker: 6
Mason: 1, 4, 7
Jacob: 1, 4

Tori: 1, 2, 4
Will: 1

b) Parker: 1, 3
Mason: 2, 3, 6
Jacob: 1 , 2
Tori: 3, 6
Will: -

c) Parker: 2, 3, 4 ,5
Mason: 1, 2
Jacob: 3, 4
Tori: 4, 6
Will: -

d) The creation of the repository was performed by Mason and Parker. Code contributions were performed by Mason and Jacob.

e) The video recording was performed by Mason, Parker, Tori, and Jacob.

6) Plans for the next increment

If this report if for the first or second increment, describe what are you planning to achieve in the next increment.

- We currently spent most of the first increment planning and discussing the style and functionality of the game. We currently have just the game window, but we plan for the next increment to add a capped framerate of 60 fps, keymapping for movements, displaying the fighters and hopefully start on health and damage. Once we get the main game loop finished, we can start to work on a main menu, character selection (if we choose to make many characters), and map selection.

7) Link to video

https://youtu.be/DdpIjapDaZQ

Software Implementation and Testing Document

For

Group <7>

Version 1.0

Authors:

Mason Russo

Parker Thompson

Tori Lee

Jacob Sonn

William Andrews

1. Programming Languages (5 points)

List the programming languages used in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).

- As of now, Python is the main programming language used in the project. Other programming languages used in the future are yet to be decided.

2. Platforms, APIs, Databases, and other technologies used (5 points)

List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).

- Pygame is the main library we are using to implement game functionality. Other libraries are yet to be decided and will be determined later as more functionality is implemented.

3. Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the **functional requirements** listed in your RD).*

- We have not done any testing yet as we have not started coding yet. We just finished discussing the functions of the games and features we want to implement.

4. Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the **non-functional requirements** listed in your RD).*

- We have not done any testing yet as we have not started coding yet. We just finished discussing the functions of the games and features we want to implement.

5. Non-Execution-based Testing (10 points)

Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).

- We have not done any testing yet as we have not started coding yet. We just finished discussing the functions of the games and features we want to implement.

Software Requirements and Design Document

For

Group <7>

Version 1.0

Authors:

Mason Russo

Parker Thompson

Tori Lee

Jacob Sonn

William Andrews

Overview (5 points)

Give a general overview of the system in 1-2 paragraphs (similar to the one in the project proposal).

- The project we are developing is a 2-D retro-style fighter game called Pixel Pummel. We plan to implement a few different map locations, and characters. The game will have a capped framerate, multiple characters, health bars, and menu options. More advanced features have not been completely agreed upon yet and won' be until we have the foundation of the game created.

Functional Requirements (10 points)

*List the **functional requirements** in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

1. Each character will have their own unique attack (high)
2. A simple menu will be implemented that allows for easy navigation (high)
3. Backgrounds will be able to select from a list (high)
4. A settings option will allow for basic customization (high)
5. 2 players will be able to play on the same keyboard (high)
6. A player will be able to enter a training stage by themselves (low)
7. An animated startup screen (low)
8. Health bars that accurately display players health (high)
9. Allow for a rematch (medium)

Non-functional Requirements (10 points)

*List the **non-functional requirements** of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

1. The game shall load within 5 seconds after being launched.
2. Game size will be minimal
3. Source code will be well documented
4. New characters can be added without any major change to the core infrastructure
5. Game will be able to run on Mac, Windows and Linux
6. The game will be able to run-on low-level hardware

Use Case Diagram (10 points)

*This section presents the **use case diagram** and the **textual descriptions** of the use cases for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*

Textual descriptions of use cases*: For the first increment, the textual descriptions for the use cases are not required. However, the textual descriptions for all use cases discovered for your system are required for the second and third iterations.*

Class Diagram and/or Sequence Diagrams (15 points)

This section presents a high-level overview of the anticipated system architecture using a ***class diagram*** *and/or* ***sequence diagrams****.*

If the main ***paradigm*** *used in your project is* ***Object Oriented*** *(i.e., you have classes or something that acts similar to classes in your system), then draw the* ***Class Diagram of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***

If the main ***paradigm*** *in your system is* ***not*** ***Object Oriented*** *(i.e., you* ***do not*** *have classes or anything similar to classes in your system) then only draw* ***Sequence Diagrams****,* ***but for all the use cases of your system.*** *In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

Class Diagrams *show the* ***fundamental objects/classes*** *that must be modeled with the system to satisfy its requirements and* ***the relationships*** *between them. Each class rectangle on the diagram* ***must also include the attributes and the methods of the class*** *(they can be refined between increments). All the* ***relationships between classes and their multiplicity*** *must be shown on the class diagram.*

A ***Sequence Diagram*** *simply depicts* ***interaction between objects*** *(or* ***functions -*** *in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*

Fighter class:

Move – allows the character to move

Draw – sets the character image

Operating Environment (5 points)

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

- The game will be able to run on Windows and Mac operating systems only. We have no plans to include additional functionality for other operating systems.

Assumptions and Dependencies (5 points)

List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around

the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

Assumed factors that could affect the requirements include:

- Hardware capabilities depending on the device the user is playing on
- Dependencies on third party game engines