



**Michigan
Technological
University**

College of Computing

Computer Science Department

CS3141 Team Software Project

Semester

2 Player Pac-man

Section: R01 Fall 2023

Team #: 10

Name	Role
Kyle Jarvis	Developer
Tyler Gingras	Developer
Jacob Brendemuehl	Developer
Ryan Osentoski	Scrum Master
Cade Haas	Developer

Instructor:

Serein AL-Ratrout

Abstract

This project aims to cure boredom when students have free time. The game follows a similar format of the widely known game pac-man with two players on one keyboard with a potential ghost-player option in the there as well (possibly A.I.). We thought the older pac-man was too boring and we decided to add another player in the mix with more interactive parts. Two players will face off against each other and four CPU controlled ghosts to evoke a more exciting and connective playing experience. We plan on using Unity to achieve this, some of our members have used it while others have not, so it's a learning experience.

Table of Contents

Table of Figures

Chapter1

Specification

1.1 Introduction

Two player pac-man game that is developed using Unity with CPU controlled ghosts. We feel like this is important since many students on campus have a lot of free time between classes and this will give them something to do in the meantime other than homework. We expect to have a fully working and fun game for said students which can kill time for them.

1.2 Problem Statement

We're trying to solve boredom, for when there's nothing else to do. We think that by developing this software, we will be able to lower the boredom levels of the user(s) through entertainment and give them something to do. By needing to learn strategies for the games, we are increasing the iq levels of our users. By learning patterns and constraints of the game, we will make our users smarter.

1.3 Aim and Objectives

The aim of this project is to develop a desktop game where two players can play as pac-man characters and potentially a ghost character. The game will also have CPU controlled ghosts that will try and kill the players as they play. Instead of one person playing an original pac-man game by themselves, they have a friend or two to play with instead, which improves social interactions.

Some of the functional requirements of our game would include:

- User(s) can control the pac-man characters with the WASD &/or arrow keys
- Have different abilities, attacks, powerups, etc for each player.
- Have CPU controlled ghost characters.
- Have alternate game modes depending on the situation.
- Will have power ups and a point system involved.

Some of the non-functional requirements of our game would include:

1. The game should look aesthetically pleasing
2. The game should not be jittery or otherwise unpleasant to control
3. The game should have a leaderboard to report the highest scores of either player
4. Users can change the color/appearance of the pac-man characters.

1.4 Stakeholders

Stakeholders will be us as the developers, people who play this game, upper management/managers of the project once it is sent out, and our teacher.

1.5 Methodology

SCRUM methodology will be used for this project, we will follow our sprint requirements week by week until the project is finished, changing things as we go along. We will probably use Jira to manage the process as well. We plan to first decide the layout/format of our game which includes the map design(s), sprites, and other gameplay mechanics that would dictate how we develop some of the front & back end. Then we would begin to develop some of the backend components including basic user interface, artificial intelligence for the non-player characters, and any processes that do not directly interact with the user. Finally we would focus on the front end stuff, particularly in the non-functional requirements category.

1.6 Tools

We will be using the Unity Engine and the C# programming language in order to complete this project. We will also use GitHub to maintain a repository of code, and keep track of any changes that are made. This will help make sure everyone working on the project has the same code, and can access previous versions if needed. We will use Jira to keep track of our Sprint requirements, timespans, and documentation. We will likely also use softwares such as photoshop or audacity for the creation of external resources to be integrated into the programming.

1.7 High-Level Business Requirements

Some of the functional requirements of our game would include:

- User(s) can control the pac-man characters with the WASD &/or arrow keys
- Have different abilities, attacks, powerups, etc for each player.
- Have CPU controlled ghost characters.
- Have alternate game modes depending on the situation.
- Will have power ups and a point system involved.

Some of the non-functional requirements of our game would include:

- The game should look aesthetically pleasing
- The game should not be jittery or otherwise unpleasant to control
- The game should have a leaderboard to report the highest scores of either player
- Users can change the color/appearance of the pac-man characters.

1.8 Product backlog

More * = Higher Priority.

You can use the following table:

Priority	User Story	Tasks	Estimated effort	Sprint
***	As a player, I want to be able to control my pac-man character, use abilities obtained in the map, and maximise my score to beat others.	Player controls	1H	1
		Obtaining abilities	1.5H	
		Maximising score	1H	
*	As a player, I want to see my high score, so that I can try and beat it.	Leaderboard system	1.5H	2
**	As a player, I want a pleasant gameplay experience.	Prototyping	2W	1
*	As a player, I want to change the appearance of my character, so that I can feel special.	Optional appearances menu	0.5H	2
		Different sprites	2H	
**	As a player, I want to control the ghosts, so that I can play against another player.	Controllable ghosts	1H	2
		Unique player-ghost interactions	2H	

1.9 Security Requirements engineering practice(s)

[you can apply one or more of the following practices]

1. Establish Security and Privacy Requirements
 - o Identify key milestones and deliverables
 - o Assign security experts
 - o Define minimum security and privacy criteria for an application
 - o Deploying a security vulnerability tracking system
2. Create Quality Gates/Bug Bars
 - o Define minimum acceptable levels of security and privacy quality
3. Perform Security and Privacy Risk Assessments
 - o Identify the need for threat modeling and security design reviews

Chapter 2

Analysis and Design

For chapter 2 analysis and design,
Analysis specifies what the system should do, and design is the process of planning a new system or replacing an existing system by defining its components or modules to satisfy specific requirements.

In this chapter, you need to draw the following UML diagram + textual format to explain them

UML use case diagram + textual format to explain it

UML class diagram + textual format to explain it

UML sequence diagram + textual format to explain it

You can keep revisiting and updating this chapter (updating the design for the proposed system) during project development since the scrum software process allows this.

- **Security Design practice(s)**

[you can apply one or more of the following practices]

1. Establish Design Requirements

- o Address security and privacy concerns early to reduce a project's expense.
- o Validate all design specifications against a functional specification

2. Perform Attack Surface Analysis/Reduction

- o The points attacker can use to break your system
- o Analyze the overall attack surface (points)
- o disable or restrict access to system services
- o apply the principle of least privilege
- o Employing layered defenses

3. Use Threat Modeling

- o Simulate attack scenarios.

Chapter 3

Implementation

For the implementation section, you add screenshots for your implementation, and interfaces of your product/ project with simple statements to explain how users will use them.

Keep revisiting and updating this chapter during project development since the scrum software process allows this. (updating with screenshots for the new increments/components)

Security implementation practice(s)

[you can apply the following practice]

1. Deprecate Unsafe Functions
 - o Analyze all project functions to ban unsafe ones
 - o Replace the unsafe functions with safer alternatives

Chapter 4

Validation

For Chapter 4 (Validation): here you need to write about the process of checking that your software system meets specifications and requirements so that it fulfils its intended purpose, and to confirm or prove the accuracy of your project.

Write about your testing and validation; **level of testing** you had, unit testing, integration testing, validation testing, and acceptance testing. Did you have **manual or automated** testing or both? specify the part(s) that have automated testing and part(s) that have manual testing, and **What is your oracle?**

Write the test cases for valid and invalid **input** (please see Week3 Automated Testing/ slide 11),

then confirm that no errors in the code that the application is able to operate in the required condition (OS, web browsers) and that you have created the code correctly.

For validation and acceptance testing write who tested your system. MTU students? computer science student? other department students? your group only? other college students? public users? How many students/users? How many times? could they use it easily or did they make mistakes?

You keep revisiting and updating this chapter during project development since the scrum software process allows this. (updating with test cases/ test suites you have done for the new increments/components)

Create a **testing document** with a **comprehensive list of actions to be performed and the expected results of those actions**

#	Description	Test data/input	Expected data	Actual result	Pass/Fail	Comment			
1	Login	admin/abcd	logged in	as expected	Pass				
2	Login	user/user	error msg	as expected	Pass				
3	navigate to "https://testpro	https://testproject.io/	application is displayed	as expected	Pass				
4	click "learn more"	click "learn more"	product page is displayed	as expected	Pass				
5	click "plan"	click "plan"	plan page is displayed	error page	Fail				
6									

Security validation practice(s)

[you can apply one or more of the following practices]

1. Perform Dynamic Analysis
 - o Perform run-time verification of the software
 - o Penetration test.
2. Perform Fuzz Testing: an automated software testing technique that attempts to find hackable software bugs by randomly feeding invalid and unexpected inputs and data into a computer program in order to find coding errors and security loopholes.
3. Conduct Attack Surface Review

- o* Review attack surface upon code completion
- o* Ensure that any new attack vectors created as a result of the changes have been reviewed and mitigated including threat models

Chapter 5

Limitations and Future Work

For Chapter 5

Limitations: address everything that the project left, if some project backlog items/features/ requirements have not been implemented then mention them in this part with an explanation/justification why you couldn't implement them (Time constraints the time was not enough, some developers were unavailable, because of COVID19, or tool limitationetc.). Many students tend to feel that presenting the limits of their work makes work weaker. on the contrary, approaching this section shows maturity for the academic universe, and writing about them actually strengthens your work by identifying any problems before reviewers/readers find them.

Future work : if the limitations can be addressed in the future then add this in here in future work, moreover, if you believe this project can be extendable (add more features/more parts) that the project is worth extending to a Final Year Project (FYP) by you or other students or can be adopted and extended by industry as a product so you can give directions for that in future work.

Chapter 6

Conclusion

For Chapter 6 (Conclusion),: write what you have concluded.

Examples:

I solved many problems in the project...

This application/project/system was applied to improve the learning process.

The results of this project showed that the system significantly facilitated the students' learning process.

The system is useless, acceptable, usable, beneficial, or maybe enjoyable and why do you believe that?

References

(Include any references to external documents or materials (for example, tutorials the team will be using, literature , web references or links to documentation of third-party tools you will use) here.

The references should be properly numbered and correctly used in the text.

The Reference section should be in the following fashion:

References

Journal, Magazine/ Newspaper Article

[1] Author, "Title," *Journal name*, p. pages, year.

Book

[2] Author, Book Title, publisher, year.

Internet Web page:

[3] Author, "Name of the Web Page," [Online]. Available: URL. [Accessed Date].

Appendix:

Security Release practice(s)

1. Create an Incident Response Plan
 - o Prepare an Incident Response Plan to address new threats that can emerge over time
 - o identify appropriate security emergency contacts
 - o establish security servicing from other groups
2. Conduct Final Security Review
 - o Review all security activities that were performed to ensure software release readiness. Examine threat models, tools outputs, and performance against the quality gates and bug bars defined during the Requirements Phase
3. Certify Release and Archive
 - o Certify software prior to a release to ensure security and privacy requirements were met
 - o Archive all specifications, source code, binaries, private symbols, threat models, documentation, emergency response plans, and license and servicing terms for any third-party software.