Software Project Management Plan

Project Domino

Version 1.0
October 2nd, 2022

Domino Squad: Avik Shenoy, Thanh Tran,
Amir Roochi, Daniel Sarkisian,
Brendan Beagin

Computer Science Dept/ CSUN

COMP490, Edmund Dantes

Revisions

Version	Primary Author	Description of Version	Date completed
1.0	All Team Members	Original Document	10/2/2022
1.1			
1.2			
1.3			
1.4			
1.5			

Table of Contents

1. Introduction

1.1 Project Overview

Purpose

Project Domino is a video game being created in Unreal Engine 5. The game will be a third-person horror-centric action adventure game with a focus on combat and exploration. The game's art style will be one that is inspired by the PlayStation 1 era from the late 1990s. The game's theme will be inspired by dark medieval/dark fantasy, with horror elements. The gameplay will feature multiple levels, enemies, bosses, non-player characters, and role-playing elements (upgrades, inventory). The game will also feature a narrative that guides the player along the journey.

Scope

Project Domino will be available on Steam. Anyone, including students at CSUN, can play the game so long as they have a computer. The intended audience are fans of the dark fantasy/ medieval genre, as well as fans of action games (such as Souls-like games) and horror games (such as Resident Evil). The game is also aimed at retro game fans.

Art Style

The graphics of Project Domino will harken back to the era of Playstation 1. The choice to go retro was made because Playstation 1 is a treasured console that birthed fond memories for millions of people. We want to appeal to those fans and create a game that reminds them of their childhood. The PS1 art style is also popular with modern audiences as well, and has made quite a resurgence lately. In addition, the PS1 art style's low-poly nature allows us to create assets, such as models and animations, ourselves. The PS1 art style is a charming and classic aesthetic that will forever be timeless, and therefore we want to make a game that honors that generation.

Story Synopsis

The story follows a knight wanderer. He's a lone wolf, no allegiance to any kingdom or clan. He does odd jobs and favors for a living. He's stoic and phlegmatic, and exudes an aura of mystery. No one can get a read on what he's thinking or feeling. But just by looking at him, one may get the feeling that he had a difficult past.

One day, the knight was riding along a deserted road when a man pleaded for his help. The man told the knight that his carriage had been raided by a group of men, and that they had taken his daughter. The father managed to escape from the hooded men, and sought help from the first stranger he could find. The knight berated the father for his cowardice in the face of losing his daughter, and dismissed him, for he had no time for his concerns, especially since the father had no money to offer. But the father begged the knight, and made a point of the brutality of the hooded men and what they might do to his daughter. This caught the attention of the knight, and pressed for details on these "hooded men". Upon hearing the details, the knight took a moment to contemplate in silence. He took a long look at the father, and agreed to look for his daughter. The father was elated with joy, and tells the knight to go to the nearby village, for that is where he thinks his daughter was taken. The knight struts off in the direction of the village.

The knight arrived at the village, and took note of how dreary and decrepit it was, as if it was abandoned a long time ago. That did not seem right to the knight, noting how the village was a vast plot of land, and a large castle loomed tall behind the village. Surely someone must live here, the knight thought to himself. The knight carefully made his way into the village, observing how old and unkempt the houses and other structures looked. It was as if nature had reclaimed much of the village. However, the knight soon took note that there were hints of human life. There were some tools strewn about, but they looked like they were placed there recently. There were some dead animals, such as cows and chickens, yet they had not died naturally, they were slaughtered, and the blood was still fresh. Most unnerving of all, the knight could hear faint sounds of bells. There would be a ring here and there, sometimes spaced out, sometimes close together. It did not sound like the wind. The village emitted an ominous and eerie atmosphere, there was simply something off about it. Even the knight, who was usually dauntless and resolute in the face of fear, was shaken by the ambience of the village. The knight slowed his pace as he walked through the dilapidated village, a ripple of unease surging through him.

The knight walked guardedly through the village, watchful for any slight change in the atmosphere. But then he came to an abrupt halt, and looked ahead leaning forward, straining his eyesight to confirm what he thought he was looking at. There in the distance was a man sitting on a bench next to a fire. The man was motionless, so much so that the knight was not sure if it really was a person at first glance. The knight tensed up, but then slowly walked towards the man, with a level of apprehension. As the knight got closer, he could tell the man was staring intently into the fire, his gaze unceasing, despite being approached by the knight, whom he should have noticed by now. The knight stopped a few meters away from the man. The knight took a moment to study the man. He was disheveled, his hair beard unkempt and his clothes dirty. However, he almost looked normal. Despite his unease, the knight began to wonder if perhaps this was a misunderstanding. Perhaps this man was just an ordinary villager, and that there was nothing to fear. Just as the knight thought before, there had to be someone living here in this village. Now reassured, the knight finally broke the silence and called out to the man. But the man did not react. He continued to stare into the fire, unwavered. The knight slowly walked closer, calling out again, until he was only about a meter away from the man. Finally, the man slowly stood up, and turned his gaze to the knight, his eyes open wide with intensity, his expression now contorted with tight-lipped anger and hatred. The knight took a reflexive step back, his eyes intently staring back at the man's, now feeling a sudden rush of alarm and panic.

The man then revealed a small ax hidden behind his clothes, and held it tightly next to his hip. As to not make any sudden movement, the knight slowly adjusted his sheath with his left hand and carefully grasped the hilt of his sword with his right hand. The man and the knight stood there, weapons in hand, their gaze upon each other unbroken, locked in a standstill.

Who were the group of hooded men? Why was the girl taken? What is the mystery of the village, and why is it so foreboding? What will happen to the man with the ax and the knight? Play the game to find out!

Gameplay

Combat

Project Domino will be a single-player third-person action horror game. The player will be able to perform basic functions such as look around, walk, run, and so on. The game will feature a combat system. The combat system will allow the player to perform strong and light attacks, as well as block or dodge incoming attacks. The player can perform a quick sidestep to evade light attacks and barrel roll to avoid heavier attacks. Enemies will be able to attack the player and can block and dodge the player's attacks. The player has a health and stamina bar. Performing actions like attacking and dodging costs stamina. Blocking an attack also costs stamina, unless the player blocks just as an attack is about to land, which is a perfect block. Taking damage lowers health, and when health reaches zero, the player will die. Enemies are powered by AI and will make choices and decisions based on the player's actions. Enemies will be designed to feel lifelike and intelligent. Enemies have a health bar as well and they will die when it reaches zero. Enemies also have a stance meter that the player can fill up by dealing damage and perfectly blocking enemy attacks. Once filled, the player can execute a finishing blow. The player can also push enemies back when surrounded. There will be a wide variety of enemies in Project Domino, each sporting unique attacks and abilities. Some enemies include humanoid enemies like villagers, lycans, and cultists. Other enemies include animalistic enemies like dogs. And scariest of all, there will be enemies that can be considered both animal and human. These enemies will draw inspiration from Lovecraftian media. There will also be challenging bosses that the player must overcome. The combat is designed to be fun and thrilling, as well as tense and suspenseful. It will be challenging as well as terrifying, as the enemies will be designed to frighten the player.

Role-playing

Project Domino will be designed to make the player feel like a participant in an adventure. The game will incorporate many features to make the player feel like a part of the story. Project Domino will feature friendly NPCs. The player can speak to these NPCs and they may help the player on their journey. Some NPCs may have a side questline the player can follow. This involves the player needing to do something, such as find an item, for an NPC in order to progress their questline. In addition, the game will feature an inventory system, where the player can store items they find during the game. Items include consumables, like health potions, key items, such as keys or other important items, and gear such as talismans that increase player stats. There will also be a crafting system where the player can use materials to create useful items. The player can also equip various weapons to better suit their playstyle. The combination of these stat increasing items and weapons will allow the player to play the game they want to play, and enjoy their own experience.

Exploration

Project Domino will allow the player to explore many locations. These locations include the village (Act 1), the castle (Act 2) and the forest (Act 3). The game will feature level design similar to that of Resident Evil and Metroidvania games. This means that the player will need to explore areas to find keys or solve puzzles to unlock new areas. Each location in Project Domino will be hand-crafted and designed to be fun and thrilling to explore. The player will experience a sense of wonder the first time they set foot in the village or castle. The player is encouraged to explore off the beaten path, and find hidden secrets that can aid them on their journey. The level design of Project Domino will encourage player thought, as it will be a challenge to figure out how to unlock new areas in order to progress.

Multiple Protagonists

Project Domino will feature two protagonists, the knight and the daughter. Both characters will be important to the narrative. The knight has more combat focused sections, while the daughter has more stealth and puzzle focused sections, with light combat mixed in.

Assumptions and Constraints

- The project is being designed to run on Windows PC
- The project playtime is moderate length
- The project supports playing over multiple sessions
- The project shall have a PS1/ retro graphical style and feel
- The project shall be developed using a mix of UE blueprint visual scripting language and C++
- Assumes the user has the appropriate .NET frameworks installed
- Assumes the user is running on Windows 10/11 64-bit
- Assumes the user meets the minimum hardware requirements
- Assumes the user has access to a keyboard and mouse or gamepad

Hardware Requirements

Minimum:

Operating system: Windows 10/11 64-bit
Processor: Quad-core Intel or AMD, 2.5 GHz or faster
RAM: 8 GB
Video Card/DirectX Version: DirectX 11 or DirectX 12 compatible graphics card
Mouse and Keyboard devices or gamepad

1.2 Literature Review

Ideas / History / Audience

According to our research, although PS1 style games may seem outdated these days, there are still a lot of good reasons to develop and design a game which is inspired by the PS1 era from the late 1990's. PS1 originally launched in 1994 and it was a significant revolution in gaming industry, and made a lot of fans in a short period of time, games like Tomb Raider that released in 1996 or Resident Evil, and Metal Gear Solid which released in 1998 were quite unique and there wasn't any game like them at the time, and their characters became the most iconic game characters of all time.

Looking back to PS1 era and developing a game which is inspired by that era can fulfill the sense of nostalgia for many gamers and generations and using new technologies like Unreal Engine 5 to develop this game gives us a great opportunity to develop a high-quality game that still has touches from the past and the time that many gamers still talk about and remember.

This game is going to run on windows 10/11 systems with minimum requirements of 8 GB of RAM and Direct X 11 or 12 compatible graphics card, so the simplicity of the graphical style (PS1 era), helps more gamers to play the game, because our research shows that there are still a lot of people who like to experience games from PS1 era, but are not able to find a platform or even the games themselves so based on this research we believe that our game will have audience among older gamers, and also new gamers who are looking for a game which is inspired by PS1 era, and they don't want to spend money and time to find an old gaming platform.

Version Control

We are debating either to use GitHub, Perforce or a local personal server for our version control, because the game has lots of assets and our research shows that although we can upload the game code on GitHub but for the asset management it looks like GitHub is not a good option for game development. Another available option is Perforce (Helix Core) which is industry standard for game development version control and is free for small teams (5 users), it also has good features that support different versions of Unreal Engine. It provides useful tools for both designers and developers. "Helix Core scales without limits and delivers high performance at scale. You can have developers working on code in multiple locations. And you can easily manage large files and complicated data."(2). Advantages of Helix Core include the client's intuitive integration into Unreal Engine as well as the limitation of storage space being a hard drive on a system we have physical access to, rather than a service that must meet the demands of many users, such as GitHub. If we decide on using Perforce, the Helix Core server will be running as a service on a linux server in a team member's closet, the connection to which will be secured using the tls protocol supported by Helix Core. After doing more research our final decision for version control is using a local personal server since it is easier to manage for our project and may deploy some version control servers like Helix core later, if we needed to.

Resources:

1. https://www.perforce.com/resources/vcs/helix-core-pricing
2. https://www.perforce.com/products/helix-core
3. https://www.ign.com/articles/best-ps1-games-playstation
4. https://www.thegamer.com/playstation-1-video-games-look-amazing-fans-forgot
5. https://github.com/topics/game-development

2. Project Requirement

2.1 Roles and Responsibilities

Team Member	Roles	Email
Avik Shenoy	Team leader, C++ and blueprints developer, blender modeler and animator, gameplay designer, narrative designer, gameplay tester	avik.shenoy.858@my.csun.edu
Thanh Tran	blender modeler and animator, C++ and blueprints developer, narrative designer	thanh.tran.713@my.csun.edu
Amir Roochi	C++ and blueprint developer, gameplay tester	amir-roochi@outlook.com
Brendan Beagin	C++ and blueprint developer, gameplay designer, gameplay tester	brendan.beagin.339@my.csun.edu
Daniel Sarkisian	C++ and blueprint developer, gameplay designer, gameplay tester	danedsar@gmail.com

Role	Responsibility
Team Leader	Responsible for managing and organizing both the team and the project. In charge of gathering any questions and concerns and relaying them to the professor, as well as submitting materials and assignments. Responsible for leading team discussion and gathering information pertaining to the project. Also responsible for assigning tasks and work to team members.
C++ Developer	Responsible for writing C++ code. Main purpose is to program, design, analyze, and debug C++ code for the game.
Blueprints developer	Responsible for writing Blueprints code. Main purpose is to program, design, analyze, and debug Blueprints code for the game.
Blender Modeler	Responsible for 3D asset creation. Main purpose is to model game assets, such as objects and characters.
Blender Animator	Responsible for 3D animation creation. Main purpose is to create animations for objects and characters.

Gameplay designer	Responsible for designing the game. Main purpose is to design gameplay mechanics, scenarios, encounters, etc.
Narrative designer	Responsible for the game's story and lore. Main purpose is to write the story and characters, create story elements and ideas, and expand the game's lore.
Gameplay tester	Responsible for testing the game. Main purpose is to play the game and provide feedback on the gameplay, suggest improvements, as well as check for any potential bugs or glitches.

2.1 Tools and Techniques

Unreal Engine 5 (Game engine. Editor environment that our video game will be developed in. Provides creational tools, architectural visualization, simulation, real-time 3D content creation)
C++ (General-purpose object oriented programming language. Used for the development of our game)
Blueprints (Visual node scripting language. Allows for the development of our game)
Blender (3D creation software. Provides tools for modeling, rigging, animation, simulation, rendering, and video game creation. Allows for creation of model and animation assets for our game)
Perforce (Service that allows us to host project on a server and allows all team members to work on project, as well as rollback to previous project versions if current project is corrupted)
GitHub (Online repository service. Allows us to store and manage resources and code, in addition to tracking and controlling changes to code)
Discord (online social messaging platform. Allows us to communicate with each other through text and voice chat)
Waterfall method (Project development method approach that involves planning out the project first, and then executing the plan, without iterating)
Gimp (Image editing software. Used for textures)

Project Management Plan

3.1 Tasks

Milestone	Deliverable	Due
Progress Report #1	Report detailing our progress on our project.	9/25/2022
Project Presentation	Presentation that showcases our project idea.	9/25/2022
Progress Report #2	Report detailing our progress on our project.	9/25/2022
Learn UE5/ Blender	Knowledge.	Before 10/2/2022
Ideate and plan project (Gameplay mechanics, story, theme, audience, version control)	Knowledge.	Before 10/2/2022
Software Project Management Plan (SPMP)	Assignment where we plan out our project.	10/2/2022
Progress Report #3	Report detailing our progress on our project.	10/9/2022
Project progress	Game includes player movement, version control	Before 10/16/2022
Software Requirement Specification (SRS)	Assignment where we write project requirements.	10/16/2022
Progress Report #4	Report detailing our progress on our project.	10/23/2022
Progress Report #5	Report detailing our progress on our project.	11/3/2022
Code Review #1 Artifacts	Peer review assignment.	11/3/2022
Project Progress	Game includes basic player functions, enemies, combat, level design, environments.	Before 11/10/2022
Software Design Document (SDD)	Assignment where we design our project.	11/10/2022
Progress Report #6	Report detailing our progress on our project.	11/20/2022
Code Review #2 Artifact	Peer review assignment.	11/20/2022

Progress Report #7	Report detailing our progress on our project.	12/4/2022
Progress Report #8	Report detailing our progress on our project.	12/11/2022
Project Progress	Game includes player functions and additional features, multiple enemy types, level design, environments, pickups, doors. Act 1 Village prototype nearly complete.	Before 12/11/2022
Software Test Plan (STP)	Assignment where we test the project.	12/11/2022
Code Review #3 Artifacts	Peer review assignment.	12/11/2022
Project Presentation Demo	Full Presentation/Demo of code with documentation.	12/15/2022

3.2 Assignments

Task	Deliverable/Progress	Description	Due	Assigned
Learn basic fundamentals of UE5	Knowledge	Most of the team is unfamiliar with UE5, C++, and Blueprints, so they're taking the time to learn the tools.	Fall Semester (September)	Thanh Tran, Amir Roochi, Daniel Sarkisian, Brendan Beagin
Strengthen existing knowledge of UE5	Knowledge	Avik has experience with UE5, but wants to strengthen his knowledge of the tools.	Fall Semester (September)	Avik Shenoy
Strengthen knowledge of Blender	Knowledge	Thanh and Avik have experience with Blender, but want to strengthen their knowledge of the tools.	Fall Semester (September)	Avik Shenoy, Thanh Tran
Ideate story	Planning	Figure out the	Fall Semester	All

ideas, story premise, characters		premise of the game, the main characters and their motivations, the villain.	(September)	
Ideate gameplay ideas, combat, enemies, RPG-elements	Planning	Figure out the gameplay mechanics, the combat mechanics, types of attacks and abilities, enemy types.	Fall Semester (September)	All
Research version control	Planning	Figure out if we’re going to use Git, Perforce, or Plastic SCM. After making a decision, learn how to use it.	Fall Semester (September)	All
Write Software Project Management Plan	SPMP	Class assignment where we write a plan for our project.	10/2/2022	All
Create UE5 project	UE5 project	Begin project development by creating the project	Fall Semester (October)	Avik Shenoy
Create character C++ class	Character C++ class	The character class is for the player, it’s the most important class.	Fall Semester (October)	Avik Shenoy
Program basic character functions (basic player and camera movement)	Programming	Allow the player to move around and move the camera around	Fall Semester (October)	Avik Shenoy
Create character models (main character, villagers)	Character model assets	Main character models and enemy models	Fall Semester (October)	Avik Shenoy, Thanh Tran
Create animations for	Character animations	Character animations	Fall Semester (October)	Avik Shenoy, Thanh Tran

characters (for main character, villagers)				
Program walking animation for characters	Programming	When player moves, walking animations will play	Fall Semester (October)	Avik Shenoy
Create basic enemy C++ class	Programming	The enemy will be an entity the player can fight	Fall Semester (October)	Avik Shenoy
Setup server and workspace for version control; add project to workspace	Programming	Version control allows the whole team to work on the project together.	Fall Semester (October)	All
Continue programming character features	Programming	Finetune walking, implement attacking and dodging.	Fall Semester (October)	All
Write Software Requirements Specification	SRS	Class assignment where we write our requirements for our project.	10/16/2022	All
Create assets for Act 1 (the village)	Environment assets	Assets include village houses and structures, trees, ground, etc.	Fall Semester (October)	Avik Shenoy, Thanh Tran
Continue programming character features	Programming	Further finetunement of walking, running, attacking, dodging.	Fall Semester (October)	All
Continue programming enemy features	Programming	Implement AI features, walking, attacking, etc.	Fall Semester (October)	All
Program basic combat gameplay	Programming	Player can attack enemies, health, stamina, etc.	Fall Semester (October)	All
Start designing the village area	Environment design, level design	Put assets together and early level	Fall Semester (October)	Avik Shenoy, Thanh Tran

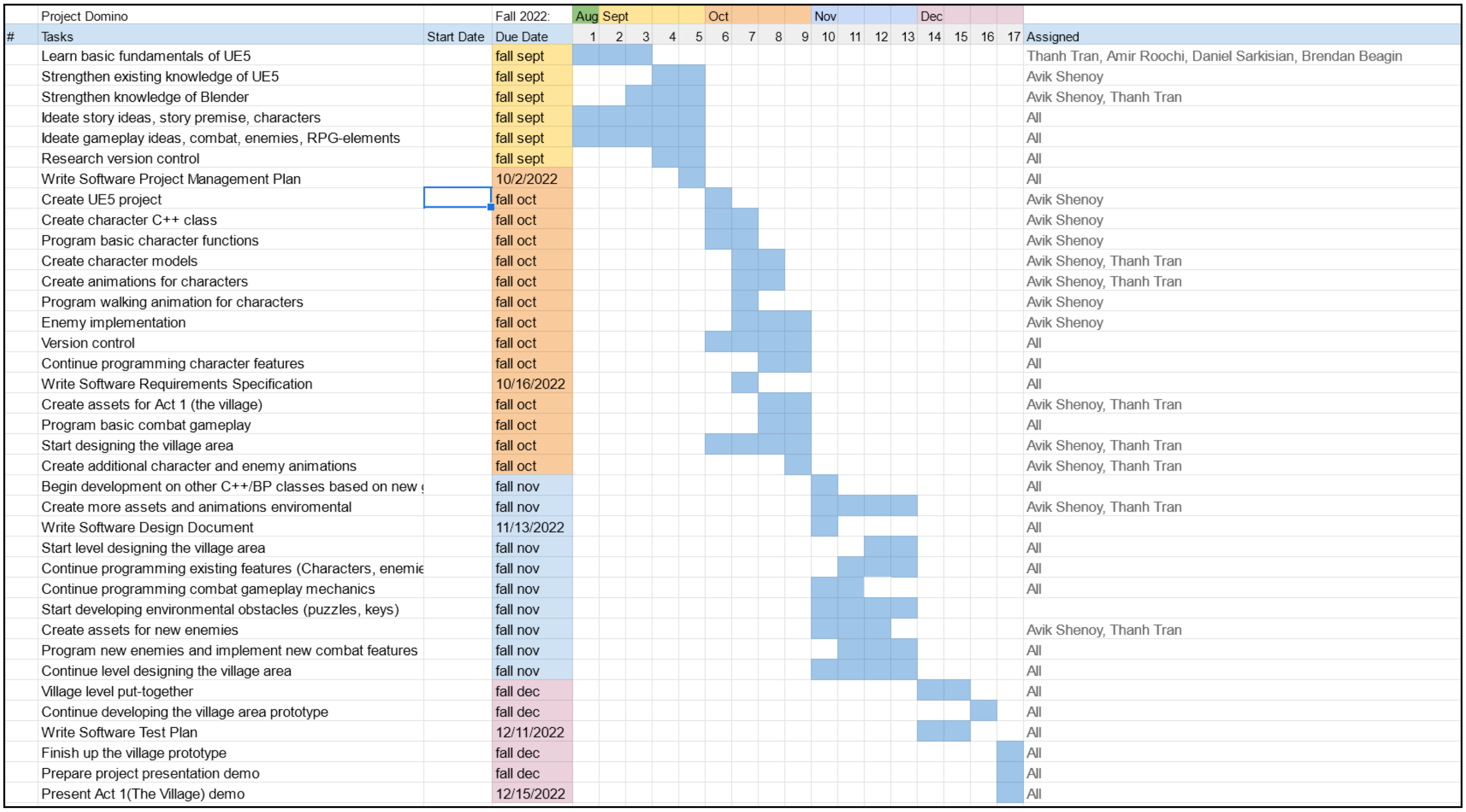
		design.		
Create additional character and enemy animations	Character animations	Make new abilities, attacks, and features for the enemy.	Fall Semester (October)	Avik Shenoy, Thanh Tran
Continue programming character features (with new animations)	Programming	Finetune existing character features.	Fall Semester (October)	All
Continue programming enemy features (with new animations)	Programming	Improved AI	Fall Semester (October)	All
Continue designing the village area	Environment design	Continue putting assets together and early level design.	Fall Semester (October)	Avik Shenoy, Thanh Tran
Begin development on other C++/BP classes based on new game mechanics and features (pickups, items)	Programming	New features such as items that the player can pick up, like keys to doors.	Fall Semester (November)	All
Create more assets and animations (for environments, characters, enemies, etc.)	Assets	New features, finetunement of features.	Fall Semester (November)	Avik Shenoy, Thanh Tran
Write Software Design Document	SDD	Class assignment where we design the project	11/13/2022	All
Start level designing the village area	Level design	Really begin designing how the level will play out gameplay-wise	Fall Semester (November)	All
Continue programming existing features (Characters,	Programming	New features, finetunement of features.	Fall Semester (November)	All

enemies, pickups, items)				
Continue programming combat gameplay mechanics	Programming	New features, finetunement of features.	Fall Semester (November)	All
Start developing environmental obstacles (puzzles, keys)	Programming, game design	Obstacles include puzzles that the player needs to solve, keys the player needs to find.		
Create assets for new enemies	Enemy assets	Create models for new enemies.	Fall Semester (November)	Avik Shenoy, Thanh Tran
Program new enemies and implement new combat features	Programming	Make new enemies that have their own attacks, abilities, AI.	Fall Semester (November)	All
Continue level designing the village area	Level design	Continue level designing the village	Fall Semester (November)	All
Start putting everything together and properly develop the village area prototype (level design, placing enemies, items, keys, puzzles, etc.)	Programming, level design, environment design	Start developing the first prototype of the game. The first area, The Village, Act 1.	Fall Semester (December)	All
Continue developing the village area prototype	Programming, level design, environment design	Continue the prototype.	Fall Semester (December)	All
Write Software Test Plan	STP	Class assignment where we test the project.	12/11/2022	All
Finish up the village prototype	Programming, level design, environment design	Finish up the village area prototype.	Fall Semester (December)	All

Prepare project presentation demo	Presentation	Prepare the prototype for a presentation showcase.	Fall Semester (December)	All
Present Act 1(The Village) demo	Presentation	Present a demo of our game.	12/15/2022	All

3.3 Time Table

Project Domino Gantt chart and misc



	Project Domino		Fall 2022:	Aug	Sept				Oct				Nov				Dec				
#	Tasks	Start Date	Due Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Assigned
	Learn basic fundamentals of UE5		fall sept	■	■	■															Thanh Tran, Amir Roochi, Daniel Sarkisian, Brendan Beagin
	Strengthen existing knowledge of UE5		fall sept				■	■													Avik Shenoy
	Strengthen knowledge of Blender		fall sept			■	■	■													Avik Shenoy, Thanh Tran
	Ideate story ideas, story premise, characters		fall sept	■	■	■	■	■													All
	Ideate gameplay ideas, combat, enemies, RPG-elements		fall sept	■	■	■	■	■													All
	Research version control		fall sept				■	■													All
	Write Software Project Management Plan		10/2/2022					■													All
	Create UE5 project		fall oct						■												Avik Shenoy
	Create character C++ class		fall oct						■	■											Avik Shenoy
	Program basic character functions		fall oct						■	■											Avik Shenoy
	Create character models		fall oct							■	■										Avik Shenoy, Thanh Tran
	Create animations for characters		fall oct							■	■										Avik Shenoy, Thanh Tran
	Program walking animation for characters		fall oct							■											Avik Shenoy
	Enemy implementation		fall oct							■	■	■									Avik Shenoy
	Version control		fall oct						■	■	■	■									All
	Continue programming character features		fall oct								■	■									All
	Write Software Requirements Specification		10/16/2022							■											All
	Create assets for Act 1 (the village)		fall oct								■	■									Avik Shenoy, Thanh Tran
	Program basic combat gameplay		fall oct								■	■									All
	Start designing the village area		fall oct						■	■	■	■									Avik Shenoy, Thanh Tran
	Create additional character and enemy animations		fall oct									■									Avik Shenoy, Thanh Tran
	Begin development on other C++/BP classes based on new		fall nov										■								All
	Create more assets and animations enviromental		fall nov										■	■	■	■					Avik Shenoy, Thanh Tran
	Write Software Design Document		11/13/2022										■								All
	Start level designing the village area		fall nov												■	■					All
	Continue programming existing features (Characters, enemie		fall nov											■	■	■					All
	Continue programming combat gameplay mechanics		fall nov										■	■							All
	Start developing environmental obstacles (puzzles, keys)		fall nov										■	■	■	■					
	Create assets for new enemies		fall nov										■	■	■						Avik Shenoy, Thanh Tran
	Program new enemies and implement new combat features		fall nov											■	■	■					All
	Continue level designing the village area		fall nov										■	■	■	■					All
	Village level put-together		fall dec														■	■			All
	Continue developing the village area prototype		fall dec																■		All
	Write Software Test Plan		12/11/2022														■	■			All
	Finish up the village prototype		fall dec																	■	All
	Prepare project presentation demo		fall dec																	■	All
	Present Act 1(The Village) demo		12/15/2022																	■	All

Additional Materials

Definitions, Acronyms and Abbreviations

Acronyms and Abbreviations	Definition
Unreal Engine 5 (UE5)	Unreal Engine 5(Game engine editor)
Blueprint (BP)	Blueprint(Visual node scripting language)
Asset	Something that is used in the video game, such as an object, character, sound effect, particle effect, etc.
Souls-like	Popular action adventure game genre. Focused on difficult but rewarding combat, extensive exploration, and epic world design.
Resident Evil	Popular horror game series. Focused on exploration and survival horror.
Dark fantasy/ dark medieval	Sub-genre of the popular fantasy/medieval genre. Focused on a more bleak and horrific rendition of the fantasy/medieval genre.
Playstation 1 era	Playstation 1 is a popular game console developed by Sony. The playstation 1 era refers to the lifespan of the console, from its release in 1994 to 2000, ending with the launch of Playstation 2.
Non-player character (NPC)	A character in a game that is controlled by artificial intelligence (AI).
Artificial intelligence (AI)	A machine that is programmed to act like a human.
Steam	Popular game client service. Facilitates purchasing, storing, and playing of games.
Single-player	Describes a game in which only one player can play.
Third-person	Describes a game in which the player camera is detached from the player's eyes, allowing the player to see themselves in the game. The player camera usually follows closely behind the player.
Metroidvania	A popular game genre based on the Castlevania and Metroid games series. Focused on exploration and level design in which new areas are unlocked upon finding keys, obtaining new abilities, or solving puzzles.

Role-Playing game (RPG)	A popular game genre. Focused on making the player feel like they are a part of the world and story, as well as allowing them to create their own character and role-play as them.

References

1. Unreal Engine FAQ [https://www.unrealengine.com/en-US/faq]
2. Blender home page [https://www.blender.org/about/]
3. Github FAQ [https://kinsta.com/knowledgebase/what-is-github/]
4. Perforce manuals [https://www.perforce.com/manuals/p4sag/Content/P4SAG/Home-p4sag.html]