CS 426 - Shadow Ninjas

Team Role	Full Name	UIN
Game Designer/Manager	Kevin Kowalski	kkowal28
Engineer	Marcin Perkowski	mperko6
Engineer	Jonathan Vega	jvega30

Pitch Presentation (containing formal elements):
https://drive.google.com/file/d/1eNNX7loy0RJ2vTp9Lr_E_d9wo2D8Gf1G/view?usp=sharing

Final Presentation (quick demo video + what we learned as a team):
https://docs.google.com/presentation/d/1PL024VRayPfNc3d_7uizbwFREp8NN0qRwvUfXitf2dI/edit?usp=sharing

Collaborative Tools

Online Code Repository	Bug Tracker	Formal Design Document	Calendar
GitHub	GitHub	Here	Here and here

Important Dates

- ~~Assignment 6 + prototype due: Monday, March 30 @ 11:59 PM & Wednesday, April 1~~
- ~~Assignment 7 + demo due: Tuesday, April 7 @ 11:59PM~~
- ~~Assignment 8 + demo due: Tuesday, April 14 @ 11:59PM~~
- ~~Assignment 9 + demo due: Tuesday, April 21@ 11:59PM~~
- ~~Assignment 10 + demo due: Wednesday, April 29 @ 12:05 PM~~

Important Note

- Enable table of contents in *View → Show document outline* for easier navigation

Weekly Progress Report / Changelog

Week	Kevin	Jonathan	Marcin
11	• Created initial prototype level with pushable crates, extinguishable torches, and basic movement. • Wrote up the design document outline and organized this Google Doc	• Created Level Design Sketches and Writeup. • Pair-programmed with Marcin on incorporating features	• Cleaned the movement script • Fixed jump so the player can only jump once • Added a win message and exits game after 5s • Pushed version 2 of the package to github • Added a couple more issues to github • Added level components section to the level design section.
12	• Add patrol AI, lights, and sound • Animated torches, enemies, and a bit of ninja • Textured castle and skybox • General improvements	• Added grass texture and optimized it for low-end computers • Added jump animation to Ninja model • Implemented SFX for extinguished touches	• Added crate texture • Added lose screen and level reset script when you lose • Added hitboxes on all lights on the levels so when player walks into it he gets discovered and loses • Added walk animation to the player character • Added firing range puzzle with bow and arrow textures, sound when arrow gets shot, light and light animation since the arrows are flame arrows and script for enemies to shoot them in random intervals
13	• Add several menus to game • Added new lock-on ability for easier and better aiming • Added sound to shuriken, win screen, and main menu • Fixed AI animations and attached tracked objects to their hands	• Added background 3D sound of crickets depending on player location • On player detect, added NPC alert sound • Adjusted player to fall at start of level, and display a quick intro message	• Added pushing sound for crate alongside the script for it • Added background sound of wind • Added burn sound effect for torches and flame arrows • Critiqued the sounds of existing game

	• Extended the game • General improvements	• Put initial shuriken image for later displaying ammo count • Made adjustments to the 'Start' and 'Goal' font to match the default fonts used throughout the game • Repositioned Goal text from its Status area to the center to keep design	• Fixed the grass collision with player bug that caused played to get stuck in it • Changed the lose/win message, font and placement • Added light for a player so he can see in dark places better • Added tips to be displayed in 3 areas on the first level to inform user of the controls • Made a script for trigger areas for control tips
14	• Adjusted movement mechanics per the feedback from alpha • Improved the controls screen in main menu to be more clear • Added opening cutscene to game • Improved first level to have more clear path • Added new level	• Added Credits scene (+ current references used) with scrolling text animation • Began construction and initial design of level_2 (which takes place in the entrance of the castle) • Created a wireframe shader [doesn't work with <Metal> graphics API]	• Added shader for diffuse lighting for player • Added lives and shurikens for player UI along with icons and scripts limiting use of them • Added end level screen with message, 2 buttons to go to main menu or continue to next level, auto saving the game and animated display for score • Added save/load for the game, added test-debug canvas with 2 save/load buttons to test it, it saves shurikens, lives and position of player
15	• Fixed and improved parts of level 0 and level 1 • Additional tweaks to menus for better look, and placement of items • Added aura to ammo refill item and sound effect • Added camera pan at beginning of level	• Used level design ideas from American McGee and John Romero to start and finish Level_2 (which is the third stage) • Created model for spikes to harm the player	• Started and finished implementing level_3 alongside all scripts required for it • Cleaned up bugs and documents • We are done!

Design Document (*following textbook structure, pg. 448*)

---

1. Design History
    1. Version 0.6 *(Assignment 6 - Prototype)*
        - Placeholder models for characters and props
        - Pushable crates
        - Basic movement and shooting
        - Simple torch light source model and script for extinguishing

    2. Version 0.7 *(Assignment 7)*

        New features:
        - Waypoint and attack AI
        - Environment textures
        - Enemy, torch, and stepping sound effects
        - Torch and flaming arrow particle system
        - Light sources from handheld enemy torches and flaming arrows
        - Player and enemy character models
        - Idle, sneak, jump, and throw animations for player
        - Idle, walk, and shoot animations for enemies
        - Win/lose scripts
        - Scene changes on level complete

        Changes:
        - Improved player movement
        - Improved torch hitbox for easier collision
        - Improved light rendering
        - Updated first level's design
        - Updated/replaced some placeholder models

    3. Version 0.8 *(Assignment 8)*

        New features:
        - Main menu with controls menu and music
        - Pause menu
        - Lock-on aiming
        - In-game tip text
        - Additional sounds
        - Extended game a bit

        Changes:
        - Improved consistencies with text
        - Minor fixes

4. Version 0.9 *(Assignment 9)*

New features:

- Opening cutscene
- End-of-level screen
- End credits screen
- New in-game UI
- Saving and loading game saves
- New shaders
- More levels

Changes:

- Improved main menu controls screen
- Improved movement mechanics
- Improved level design for clearer navigation
- Minor fixes

5. Version 1.0 *(Assignment 10 - Final)*

New features:

- Camera pan at start of levels
- Shuriken replenish item
- Aura effect for refill and goal
- Two new levels

Changes:

- Improved main menu controls screen
- Improved legibility of opening cutscene text
- Improved first two levels for movement and challenge
- Minor fixes

2. Vision Statement

The vision for this game is to put a new and unique spin on the average puzzler, platformer, or some combination of the two genres. To do so, the player will have to put themselves in the mindset of a ninja to act in the dark and progress through levels without alerting enemies. Several kinds of obstacles and puzzles will be laid out within a map, and the puzzle aspect of this game is to navigate around these obstacles, which are the openly lit areas and the enemies themselves. Some of this navigation will come in the form of a platforming, or skilled jumps and maneuvering to get from one point to another.

The obstacles previously mentioned will be primarily in the form of sensical light sources, such as torches, campfires or even sunlight. The player will have limited resources to interact with the environment in several ways so that they can either extinguish these light sources to sneak past them or avoid them entirely. One such example is to throw a ninja star at a torch to extinguish it or drop a crate onto a campfire, thus suffocating the flame. Removing these light

sources will present the player with new paths to traverse closer towards the goal. As a result, players must think about their choices before attempting to physically make their way towards the goal.

1. Game logline

   "Shadow Ninjas" is a single-player stealth-based puzzle platformer where the player must avoid getting caught in the light while attempting to traverse towards the goal.

2. Gameplay synopsis

   - **Uniqueness**: Most games have the player avoid the dark and attack enemies whereas this game has the player operate from the dark and avoid enemies.

   - **Mechanics**: The player must use their limited resources (such as throwing shuriken) and the environment to extinguish light sources in their path so that they may traverse the level undetected. Traversing the level includes jumping and climbing ropes/ladder/trees.

   - **Setting**: In ancient Asia, with mountains in the close distance but the gameplay takes place in fields and forests occupied by medieval-looking enemy castles and camps.

   - **Look and feel**: Simplistic shapes, low-polygon models, and cartoony graphics, similar to modern Super Mario games by Nintendo.

3. Audience, Platform, and Marketing

   1. Target audience

      Primarily for boys and girls aged 8-18 around the world but acceptable for anyone interested in puzzle platformers.

   2. Platform

      Windows PC per project requirements.

   3. System Requirements

      - Windows 10 operating system
      - Modern hardware from the last decade capable of running Unity games

   4. Top Performers

      - Super Mario Odyssey:

        - October 27, 2017 release date
        - 16+ million copies sold
        - cartoony platformer
        - part of long-lasting franchise
    - Portal:
        - October 10, 2007 release date
        - 4+ million copies sold
        - portals as puzzle mechanic
        - contains a sequel with additional fan-made sequels

5. Feature Comparison

    Shadow Ninjas creates a new spin on traditional platformer and puzzler games by using light mechanics to provide the gameplay.

6. Sales Expectations

    This game is an academic project and not being sold.

4. Legal Analysis

This game is part of an academic project that is not intended to be released to the general public for any kind of financial profit. As a result, we do not believe there is any form of legal or financial obligations to list.

5. Gameplay

1. Overview

    The player will use their limited resources and the environment of the level itself to clear out obstacles and puzzles that block the path to the goal. In between these puzzles, the player will need to platform around the level like a ninja, jumping and climbing ledges or cliffs.

2. Gameplay description

    The player will play through a few levels, each level requiring the player to reach the goal from the starting point. To reach this goal, the player will need to navigate the terrain via a combination of platforming and puzzle solving, where the puzzles are simply an obstacle that is blocking the path forward. These obstacles or puzzles involve avoiding light to prevent the player from being caught by the enemies sight. In the cases where the light cannot be avoided, the player will have some way to destroy the light source so that they can continue through without alerting any enemies.

3. Controls

- W/A/S/D - 4 directional movement
- Space - jump
- ESC - pause menu
- E - plant/retrieve only on last level
- Left Click - navigate menus and throw shuriken
- Right Click - Lock-on to torch targets

1.1 Interfaces

- Tips displayed in left upper corner on the first level
- Getting caught message displayed in the center of screen
- Pause menu located on the center of the screen when ESC is pressed
- Level complete screen displayed after reaching goal located at the center of the screen
- Lives and shuriken icons located at the bottom left corner of the screen to inform player how many are left

1.2 Rules

- Control only one ninja at a time
- Player does not have the same resources every level
  - Some levels provide more ninja stars
  - Some levels have more shadows whereas other levels might be during the day with more light to avoid
- Player immediately loses a life if caught outside the shadows
  - Next ninja takes over from there
- Player can manipulate environment (where intended to be manipulated)
  - An example is throwing a ninja star at a torch to extinguish it

1.3 Scoring/winning conditions

Winning a level is simply making it to the end of a level. Doing so can only be accomplished if the player navigated the obstacles, puzzles, and enemies without getting caught. Winning the game simply requires that the player complete all of the levels.

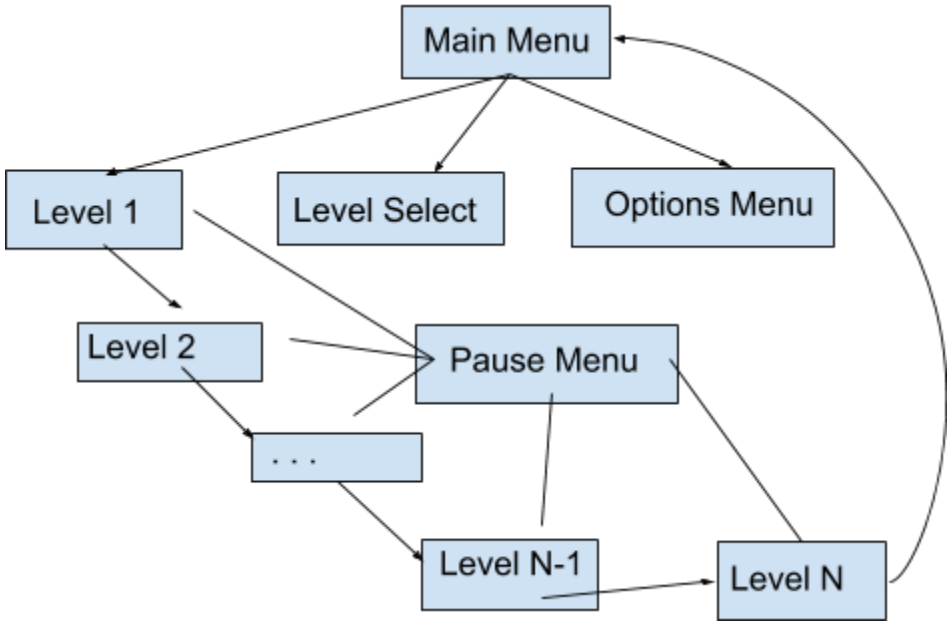
4. Modes and other features

The game has a main story mode that contains the entire game and an additional mode to go back and replay any previously completed level.

5. Levels

Click here to see the Level Design section.

6. Flowchart



7. Editor

No need to create a proprietary level editor; Unity will work well.

6. Game Characters

1. Character Design
    - Shadow Ninjas:
        - In stereotypical black clothing with some sort of visual accessory, such as a bandana.
        - Abilities:
            - Single-jump
            - Move silently
            - Throw ninja-star
            - Climb ropes or ladders
    - Villagers:
        - In traditional Asian outfits
        - Mixed ages and heights
    - Foreign soldiers:
        - In stereotypical medieval leather armor
        - Might carry some visual accessory:
            - Bow or sword on back
            - Torch in hand

- Scans or patrols an area that it is guarding

2. Types

1.1 Playable characters

- Shadow Ninjas

1.2 Nonplayer characters

- Villagers (exist in story, not in gameplay)
- Foreign soldiers

1. Behavior
    - Soldiers guard or patrol their designated area
    - If soldiers spot anything in the open light, they will alert other soldiers

2. AI
    - Soldiers will patrol on their specific defined paths
    - When a Shadow Ninja is in the open light and in close enough proximity, a guard will spot the ninja and instantly alert the other soldiers

7. Story

1. Synopsis

A foreign army is preparing to invade the lands of ancient Asia and the Shadow Ninjas are going to protect their home. The army has already set up multiple basecamps, castles and other infrastructure to prepare for the invasion. As a result, the Shadow Ninjas must sneak through undetected in the shadows to get to the heart of the army and destroy them from the inside.

2. Complete Story

In ancient Asia there is a small, desolate mountain village on the outskirts of the borders. A villager notices something unusual in the distance and reports it to the Shadow Ninjas, an elite squad of just a few ninjas whose greatest strengths are stealth and their dedication to protect the country. Upon further investigation, the Shadow Ninjas have discovered that a foreign military was the thing in the distance and this army is setting up camps and castles just outside the borders, presumably to invade sometime soon. The Shadow Ninjas see that they are hopelessly outnumbered and so they've determined that the only way to drive the foreigners away is to destroy them from the inside.

After finally making their way into the heart of the enemy basecamp, the Shadow Ninjas find a place to set up their secret weapon, the Dragon Firecracker. After setting it off, a fiery dragon made of gunpowder animates from the firework, scaring the enemies and exploding the basecamp. Even the elite Shadow Ninjas cannot escape the power of the Dragon Firecracker and after setting it off, they too are destroyed by it. The villagers see the firework explosion go off in the distance, understanding that the Shadow Ninjas have signaled a job well done. To honor their sacrifice, all lights in the village (such as lanterns and candles) are extinguished.

3. Backstory

There is no backstory.

4. Narrative devices

Still-image cutscenes (like pages in a storybook) at the start of a new game and after beating the game to explain the story.

5. Subplots

There are no subplots in Shadow Ninjas.

8. The Game World

1. Overview

Takes place in a world similar to ancient Asia but not necessarily in our real-world timeline or universe.

2. Key locations

- Mountain village: location that identified the foreign army from the distance
- Outskirts: the forests and fields at ground level beneath the mountains currently occupied by the foreign army infrastructure
- Foreign army infrastructure: the basecamps, tents, castles, towers, and other infrastructure that this army has built on the outskirts of the mountains

3. Travel

The player can only travel on foot as intended to complete levels, and these levels only take place in the outskirts and foreign army infrastructure.

4. Mapping

Maps are similar to the real world in terms of design. Lots of plants and greenery in the forests and less of it when man made structures such as a castle are present. The maps should feel like they make sense in terms of what is placed and where.

5. Scale

   The scale is similar to our real-world scale of things but slightly modified to fit according to level design. For example, trees are taller than characters but might not always be several times taller than a single character. Additionally, towers or castles might not be to the scale of a traditional medieval castle that contains dozens of chambers and rooms.

6. Physical objects

   The majority of the physical objects will be greenery in nature, infrastructure built by the foreign army, and obstacles or objects that belong to the puzzles, such as torches, wagons, or crates.

7. Weather conditions

   Weather will be calm.

8. Day and night

   Some levels might take place during the day and others during the night.

9. Time

   The story takes place in an ancient world, but the game itself has a story that spans across a few days.

10. Physics

    The physics are similar to our real-world physics, where crates can be pushed, things sink in water, items drop when falling. However, these physics may be simplified to fit the cartoonish theme.

11. Society/culture

    Not applicable.

9. Media List

1. Interface assets

   Menu items such as color scheme, cursor, and layout need to be created.

2. Environments
    - Trees/plants for forest sections of levels
    - Stone walls, crates/barrels, torches for castle portions of level

3. Characters

    Models for player ninja and different kinds of enemies

4. Animation
    - Movement for characters
        - Walking
        - Jumping
        - Look-out
    - Character interaction
        - Throwing ninja stars
        - Shooting a bow
        - Enemies being alerted
        - Pushing movable objects
    - Object interaction
        - Flames extinguished
        - Swinging ropes

5. Music and sound effects
    - Stealthy background music
        - Slow but tense music
        - Not distracting
    - Interaction sound effects
        - Throwing ninja stars
        - Shooting a bow
        - Enemies being alerted
        - Pushing movable objects
        - Flames extinguished
    - Movement sound effects
        - Enemies patrolling on foot
        - Wagon driving by

10. Technical Spec

1. Technical analysis

    1.1 New technology

    No new technology is being developed.

    1.2 Major software development tasks

Unity will provide all of the major software development tasks.

1.3 Risks

No known risks; no new technology and Unity is well documented.

1.4 Alternatives

Not applicable.

1.5 Estimated resources required

Not applicable.

2. Development platform and tools

1.1 Software

Unity (2019.3.6.f1)
Paintbrush (macOS)
Paint.NET (Windows10)
Audacity
Visual Studio
Visual Studio Code

1.2 Hardware

- Custom Windows 10 Machine
  - i7-6700K @ 4.5GHz
  - 16GB RAM
  - NVIDIA GTX 1080 8GB
- Custom Windows 10 Machine
  - i5-9600K @ 3.7GHz
  - 32GB RAM
  - NVIDIA GTX 980TI 6GB
- MacBook Pro w/ MacOS
  - 4-Core @ 2.5GHz
  - Radeon R9 M370X 2GB

3. Delivery

Download the compiled game found in the zip file from our public GitHub releases list.

1.1 Required hardware and software

Our game will require that the player has an Internet connection for the sole purpose of downloading the game, and has a modern Windows machine capable of playing 3D games.

1.2 Required materials

Not applicable.

4. Game engine

1.1 Technical specs

We did not use a custom engine, all the work was done using Unity (2019.3.6.f1).

1.2 Design

Unity is a fully featured modern game engine with plenty of documentation.

1. Features

   Built-in level editors, animators, UI builders, asset store, external editors, and more.

2. Details

   Unity contains many features that we will find useful in addition to its active community.

1.3 Collision detection

This was primarily used for checking if the player was caught under a light source.

1. Features

   Collisions were necessary to make sure players were getting caught if they touched a light source, such as that from torches or flaming arrows. This was also used for checking if a player reached a goal at the end of the level or some item pickup.

2. Details

   These collisions were set as triggers so that the game would know if a player interacted with any of the specified collisions mentioned above.

5. Interface technical specs

Our interface was designed entirely within Unity.

1.1 Features

Unity has plenty of documentation and support for UI development such as canvases, buttons, and more.

1.2 Details

Buttons were used to navigate a menu, and canvases used to draw images or text on screen.

6. Controls' technical specs

Standard controls were used for this game.

1.1 Features

Only a keyboard and mouse are required to play this game.

1.2 Details

These devices allow for standard computer game inputs as required by our game in terms of controlling the ninja and navigating menus.

7. Lighting models

Our game's primary mechanic is to avoid the light and so lighting plays a large role in our game. As a result, we needed to enable Unity to render more than a few lights at a given time.

1.1 Modes

1. Features

   Any model that emits light will have a collision on it to detect if the player ever came into contact with the light.

2. Details

   As a result, the player isn't actually colliding with the light but rather the illusion of it. Also, the light will no longer have a trigger if it has been extinguished, of which it has an animation to die out.

1.2 Models

- Torches
- Campfires

1.3 Light sources

- Sunlight

- Torches
- Campfires
- Flaming arrows shot from a bow

8. Rendering system

Unity handles all of the rendering systems in our game.

1.1 Technical specs

See official Unity documentation for detailed technical specs of the engine's rendering system.

1.2 2D/3D rendering

Menus, cutscenes, and UI elements such as the lock-on indicator are the only sources of 2D rendering, whereas everything else such as models and stages are 3D.

1.3 Camera

1. Operation

   Automatically follows the player, so no need to operate it.

2. Features

   Pans across level at the beginning of each stage before returning to the player model.

3. Details

   In fixed position in front of the player to not reveal too much about what is ahead in the level.

9. Internet/network spec

The game does not require an Internet connection to play.

10. System parameters

1.1 Max players

One player only

1.2 Servers

No servers.

1.3 Customization

No modification supported.

1.4 Connectivity

No multiplayer or other connectivity.

1.5 Websites

No website.

1.6 Persistence

Not applicable.

1.7 Saving games

Local storage saving of game state.

1.8 Loading games

Loading of local save data.

11. Other

1.1 Help

In-game instructions to explain the rules.

1.2 Manual

No manual.

1.3 Setup

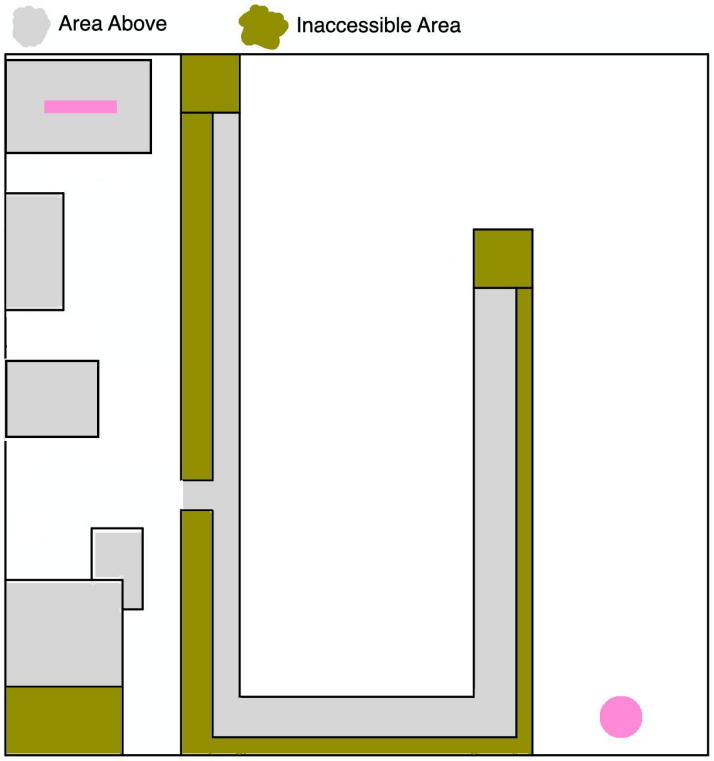
No setup other than running the `.exe`.

Level Design

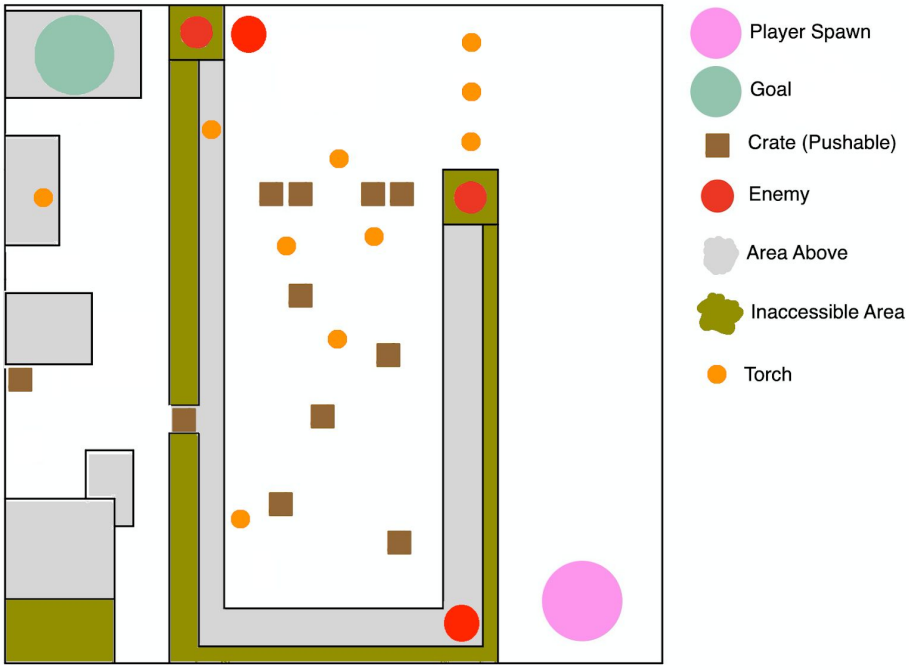
- For the floor plan, an abstract style is used to help convey the path to the player, playing as a ninja. This helps focus on the gameplay and see what challenges can be implemented for the player while introducing gameplay elements.
- We opted for elevated areas early to let the player know that they can maneuver to higher grounds. Jumping alone isn't enough, so pushable crates are placed around the map to allow access to a new area to explore.
- In regards to helping make sure that a player doesn't get lost, a sort of pseudo-linear design aids in making it easy to know where to go, but since this is also a puzzle game, the player would have to figure out how. The paths to get to the goal are up to you to create, each with their own varying difficulty.
- This also provides both flow and opportunities for strategy. Having limited shurikens, you have to decide which path you can take and how difficult the trek could be. Aiming poorly and missing a torch or other obstacle can make current situations very difficult. Saving shurikens and facing the tough paths could help later with finding shortcuts around difficult scenarios.
- For helping support the story, the pseudo-linear design helps scale the journey you embark on. For the software prototype, the level was designed to mimic and feel as if you were at the outskirts of a castle, with the goal of trying to enter it while avoiding detection.
- Traps play a role in making the journey difficult. The torches scattered in the prototype level have to be avoided. Enemy guards above are on the lookout for any ninjas, and coming into proximity with a torch can give away your location. Choosing to evade them or extinguish them with a shuriken is up to you. Darkness is your friend!
- As we progress in development, future traps will vary, but some ideas we have are puddles of water making player movement hard and lit arrows making traversal difficult due to the light they cast. The goal is to have the player blame themself and not the gameplay mechanics or controls.
- Currently items and power ups are not in the game, but under review is a powerup to replenish shurikens for the player. If we do intend to implement a shuriken power up, then limits on how much is replenished will be imposed to continue strategic mechanics . However, the concept of a power up helping lead the player is still followed with the trap placements.

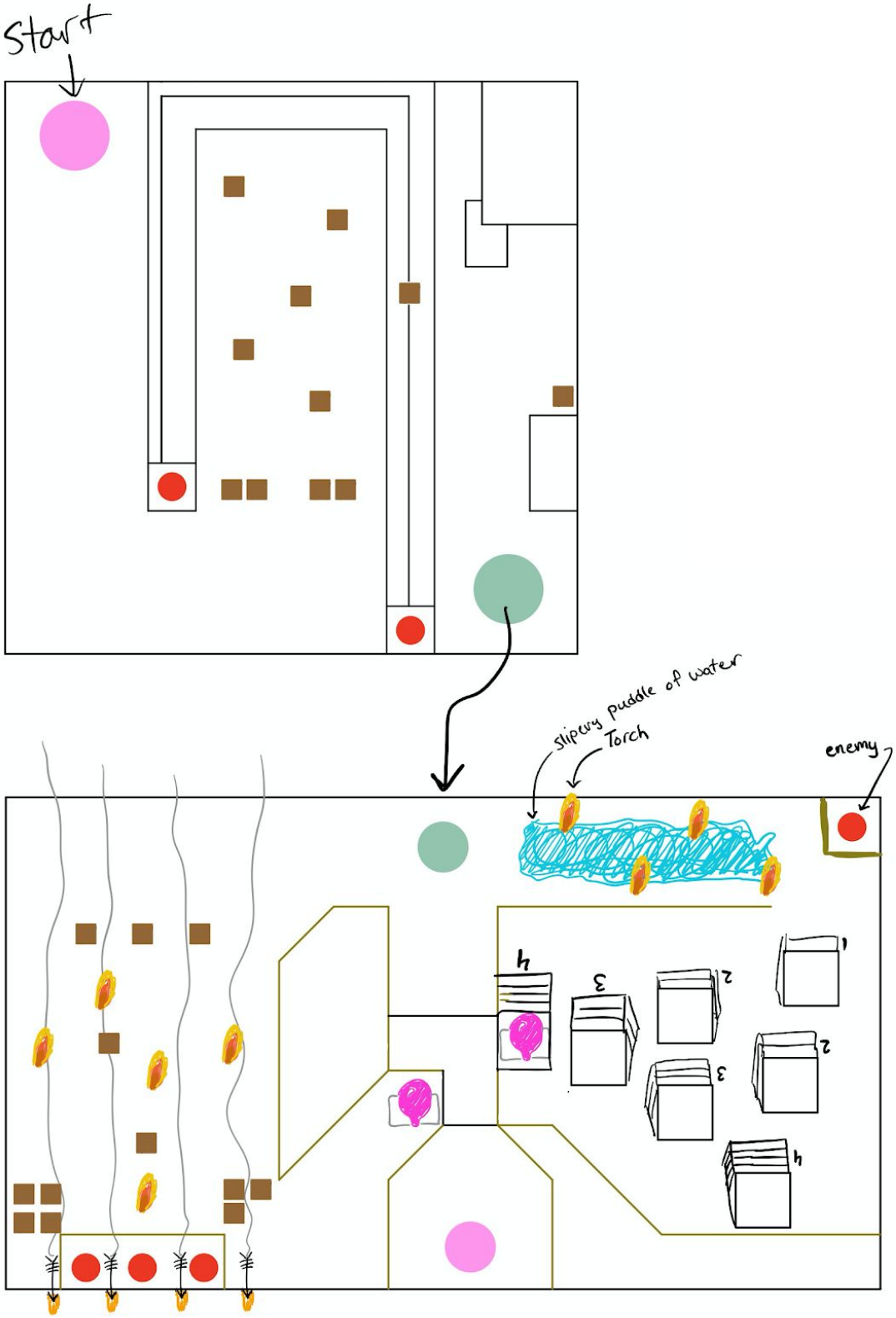
Sketches

Floor Plan



Trap Plan



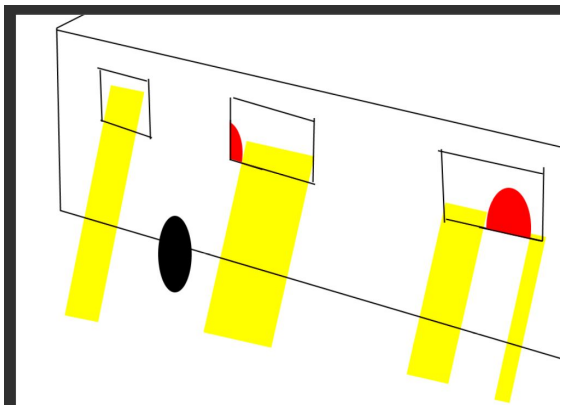


**The integer values at each platform represent the height from the ground.*

Level Design Ideas

- Idea: A firing range target practice of enemies shooting flaming arrows (light source) at target dummies.
    - Solution: make it through the firing range without getting hit by an arrow, which can be easily spotted because it's on fire

- A castle wall has windows with light beaming through it onto the ground outside. See sketch below for concept, where black oval is ninja, red ovals are soldiers, and yellow is the beaming light
    - Solution: As soldiers walk past the windows, light is momentarily obstructed



- Idea: Enemy patrols in and out of a stable/tent.
    - Solution: find a pushable crate to block the enemy when they are in side the stable/tent

- Idea: too many torches/campfires to extinguish in one concentrated spot
    - Solution: a conveniently placed wagon transporting dirt/gravel beside the torches can have its rear wheels knocked out, thus dropping the dirt/gravel on the torches. This not only extinguished the flames, but creates a ramp or mound that can be climbed to reach a new area

Level Components

- Level Geometry
    - Outskirts in forest with patrols and wagons
    - Soldier camps with watchtowers
    - Castle with soldiers, main base
    - Crates
    - Ledges
    - Walls
    - Watchtowers
    - Wagons

    - Torches
    - Camps
    - Trees
- Characters and Topology for Animation
    - Ninja
    - Soldiers
    - Fire
- Character Animations(e.g, idle, walk, run, pull-lever and death.)
    - Ninja Idle
    - Ninja Walk
    - Ninja Death/Lose
    - Ninja Jump
    - Ninja Shuriken Throw
    - Ninja Push Object
- Other Animations
    - Soldiers Walk
    - Soldiers Idle
    - Soldiers Discover Ninja
    - Soldiers Shooting Arrow
    - Torches Burning
    - Burning Arrow

Sound Design (Assignment 8):

- Well rated commercial game: Styx: Master of Shadows
- Gameplay video: https://youtu.be/QuGOnrHAiRY?t=347
- Sounds identified in game:
    - Total: 6
    - Kevin hears:
        - Ambience of wind moving/humming through chamber/room:
            - Where/When:
                - Whenever the player is in a large room, the wind or whatever is making background noise that can be slightly heard
            - Impact on Visuals:
                - I think the sound does a good job of portraying the environment; it helps draw the scale of the large rooms and that they are inside a building made of stone that can let air through. Despite the sound almost always being there, I don't think it's too strong because although it is noticeable it is subtle and easy to ignore when focusing on the gameplay.

- Repetition:
    - Because it's quite subtle, it's difficult to pick out when it loops but in general, it does not seem repetitive despite almost always being played. This makes sense because wind blows in different directions and speeds so it should make different sounds; the developers probably achieved this by having several minutes worth of wind sounds recorded or digitally manufactured.
- Balance:
    - This particular sound is well balanced because as mentioned above, it doesn't consume the player's ears and is more of a background noise that's there to prevent silence but not to annoy.

- Pleasant jingle on item pickup:
    - Where/When:
        - When the player picks up a quest/objective item such as their dagger where I heard it, a quick little jingle is played to inform that progress was made.
    - Impact on Visuals:
        - Because the game is stealth based, the jingle isn't too loud or distracting to fit with the quiet theme. It's also a jingle that isn't futuristic or something that doesn't fit with the theme of the game's world itself.
    - Repetition:
        - I don't think this sound is repetitive because there doesn't appear to be a need for picking up quest items so often that the jingle would be playing every few minutes. As a result, the player shouldn't get annoyed or tired of the sound when it plays but possibly look forward to it. As for creating the sound, it was probably made in a music-making software.
    - Balance:
        - As mentioned above, the jingle is quite calm to match the slow and stealthy feel of the game. In addition, it isn't played often at all so it feels quite balanced in the gameplay.

- Jonathan hears:
    - Liquid churning in a tank:
        - Where/When:
            - When the player is near the tanks

- Impact on Visuals:
    - It feels that they are brewing something important and tells me that I could be near something else that's important too. They probably sampled it from a bucket of real milk! This helps the environment stand out and makes you feel vulnerable being near sound since this is a stealth game.
- Repetition:
    - Given that the sound is a little pleasant to hear and is only heard when near the tank, it is not repetitive or annoying. When near two tanks, it amplifies the churning sound making the pleasant repetition into an unstructured forceful process.
- Balance:
    - This only plays when near the tank, so it's up to you whether the sound is tolerable. Being near multiple tanks can drown out other nearby sounds, but being near just one tank can make you still pick up faint sounds.
    -

- Seeing Amber Colored Waypoints:
    - Where/When:
        - The player performs an action that allows them to see hidden markers guiding them to where they have to go.
    - Impact on Visuals:
        - The sound is basically an exhale sound, but lingers at the end. Visually, with your eyes closed, it can feel as if something was entering.
    - Repetition:
        - It's up to you, but if you are constantly switching between both views then it can be annoying since the transitions aren't very fast.
    - Balance:
        - Since the game is mostly quiet, I feel that the direction of this SFX was to be quiet too. So it does fit more for the hiding, but not so much for exploring. It doesn't affect other sounds at first, but the ability pinpoints very close sounds and amplifies them.

- Marcin hears:
    - Lava sound in the distance:
        - Where/When:

        - Pretty much through the whole level, the sound seems a bit more distant the farther player is from lava.
    - Impact on Visuals:
        - It goes well with the game and makes the user feel that he is deep underground close to the lava. Considering the game is a stealth game and underground is usually dark the theme works well with the game design.
    - Repetition:
        - Even though the background noise loops all the time with all other sounds and voice acting it just adds to the whole atmosphere and it is very difficult for the user to spot this specific sound or the fact it loops all the time. It is very fitting with the theme and non intrusive.
    - Balance:
        - The sound is pretty low and slow so it fits a stealth game perfectly. I would say the sound is well balanced.

- Picking the lock:
    - Where/When:
        - Whenever a player picks a lock of any door the sound plays along with the circle to inform a user he is currently picklocking. The sound and circle are gone once the lock has been picked.
    - Impact on Visuals:
        - It adds realism that the player character is actually picking the lock. It makes the sound along with animation of character tinkering near the lock.
    - Repetition:
        - There is no repetition since there are not that many doors to unlock and picking lasts like 2 seconds. User will not get tired or annoyed by the sound.
    - Balance:
        - The sound is quiet and metallic enough to make it believable. The sound is well balanced.

Sounds Created:

- Kevin:
    - Background music or sound:
        - Main menu music:
            - The music helps bring a bit more life into the main menu and the game itself when the user first launches it. It also

helps build the theme of the game: a slow drum beat with an Asian aesthetic to it.

- Sound effects:
  - Throw Shuriken:
    - Previously the player made no sounds because in this theme, he is an expert in stealth and makes no sound however, the shuriken he throws needs to make a sound and so two sounds randomly chosen are played when thrown to indicate a throw
  - Gong hit on level complete
    - Previously nothing happened when a player beat a level except for a small win message and 5 seconds before level switched. Now the player is played a positive gong effect to fit with the Asian/ninja theme and to congratulate the player.

- Jonathan:
  - Background music or sound:
    - Sound 1:
      - Added crickets chirping to give the forest and castle entrance a little life. Persits through the first level.
  - Sound effects:
    - Sound 1:
      - When being spotted by either a guard or entering inside the zone of a torch, an alert sound plays to announce that you were detected
    - Sound 2:
      - When the player falls at the beginning of the level, a SFX plays to signal the ninja's entrance
  -

- Marcin:
  - Background music or sound:
    - Wind background sound:
      - Since the action on the first level takes place in the forest a background sound of a wind in the forest was necessary. It should go well with the aesthetic of the stealth game.
  - Sound effects:
    - Torches burning:
      - It adds more realism to the game, you can see torches everywhere so they need to make a burning sound to make it feel like the player is there.
    - Crate being pushed:

- When you push a wooden crate you expect to hear some kind of sound to confirm that the character is indeed pushing the crate.

UI Design (Assignment 8):

- Kevin
    - Instance 1:
        - Issue here: No main menu
            - The user is immediately dropped into the game with no indication of how to play, what to do, or where they are in the game.
            - 
        - Solution here:
            - When launching the game, the user will see this menu which lets them know what game they are playing, an option to exit the game, and also to view controls/info on how to play the game. As a result, there is now clarity and what to do as well as feedback on the buttons when hovering over them.

- 

- Instance 2: No indication of where player is aiming
  - Issue here:
    - Previously a player could not tell exactly where they were aiming nor could they control it well with orienting themselves perfectly. This allowed players to shoot and miss often times and nothing would happen as a result.
    - 

  - Solution here:
    - There is now a lock-on feature so that when a user right-clicks an object that can be shot, it has a floating indicator above that object that it can be shot. It also enables the ability to shoot so that only useful things can be shot at. This minimizes error possibilities.

- 

- Instance 3: Inconsistent theme text
  - Issue here: Inconsistent tip text colors with theme of game and other text
    - The tip text that appears early in the game is small and in light blue color that is inconsistent with the theme of other menus and overall game.
    - 

  - Solution here:
    - The text was made slightly larger for easier legibility as well as adjusting the colors to be consistent with the theme of the game and other menus

- 

- Jonathan:
    - Instance 1:
        - Start and Exit Level Font Didn't Match Other Fonts:
            - When the player starts the game, the font that displays 'Start' didn't match the style of the font of the tutorial messages. A consistent theme is necessary to follow. The colors however shouldn't match from the dialog messages.
            - 

        - Solution here:
            - The photo above captures the new change of font which is good, but the color is off. Green text signals that you reached the end, but in the game world, the green color can blend in with the environment and prove cumbersome. Making the font stylized and in white proved to be easier on the eyes, and also stood out signalling to the player they are near the goal

- 

- Instance 2:
  - Odd Win/Lose Message Location:
    - When you reach the goal, the congratulatory message shows at the top left of the screen almost like it was a status message. Reserving sections of the screen for varying information would prove to be better (i.e. top-left = status, bottom-left=shuriken count, center=important info, etc.)
    - 

  - Solution here:
    - Since reaching the goal and losing is important, the text is centered towards the screen, and an image of a shuriken is shown alongside it. The player now knows that the next area is about to begin, quick and seamless, and this avoids constantly looking at the small text status messages at the top-left of the screen.

- 

- Instance 3:
  - Shuriken ammo display:
    - Currently, the ninja can throw an infinite amount of shurikens, and once we begin adding the code to display shuriken ammo the ninja has, implementation of this should follow the design of the entire game. It is important to place this in the bottom-left section and nowhere else throughout the game.
    - 

      The grass is hard to see in the image but placement would be here
  - Solution here:
    - Instead of displaying a number of shurikens left, the amount of shuriken images left on the UI shows the amount the player has (in this case, one left).

- 

- Marcin:
    - Instance 1:
        - No movement control tips:
            - The player does not receive instructions about the controls. The user should be informed on the very first level about the controls that are available to him.

        - Movement control tips:
            - I made a trigger box that displays control tips along the level. Once the player enters the collision the tips will be displayed in the

top left corner of the screen. Once a player leaves the area the tips will disappear.

- Instance 2:
  - Not fitting lose/win message:
    - Both win and lose message did not fit the theme in terms of message and font. Also, I felt it was too intrusive in the middle of the screen with a big font.

  - Lose/win message font and location:
    - I changed the font used for the message so it fits better the theme of ninja. Additionally, I moved it to the top left part of the screen as I felt it was too intrusive. Lastly, I made the message be more fitting like the ninja was on a mission.

- Instance 3:
  - Too dark to see in torchless places:
    - Player can see character well in places where there are torches, however places that are dark without torches makes it very hard for player to see.

  - Added directional light:
    - I added directional light with very small intensity to make the players screen a bit brighter so he can see what his character is doing in dark places without torches.

Feedback from Alpha Release and Response (Assignment 9):

- **Playtester #1**: Allen Breyer
    - Time to complete first level:
        - 14 minutes
    - Feedback provided:
        - Trouble moving/controlling character in desired direction
        - Trouble aiming (in terms of knowing what to aim at)
        - Trouble navigating level by understanding where to go
    - Observations made:
        - Player tried to explore everything, despite some things being not part of level or intended to be part of the “background”
        - Player struggled to jump onto things because of how finicky movement is

- **Playtester #2**: Osama Ahmad
    - Time to complete first level:
        - 17 minutes played, but did not complete level
    - Feedback provided:
        - Movement (specifically rotating) was too sensitive and difficult to be precise with
        - Also confused on how to find the correct path of where to go/what to do
    - Observations made:
        - Also struggled with jumping onto things due to movement mechanics

Beta Release (Assignment 9):

- Improvements from Feedback:
    - Adjusted movement mechanic
        - The biggest issue that our playtesters had was the difficulty in controlling the character. Originally, W/S moved the character while A/D rotated him. Also, the camera was fixed in one orientation. This resulted in unfamiliar 3D movement mechanics to the player.
          To remedy this, we removed the rotation controls from the player and implemented typical WASD movement to be more familiar and easier to use for the player. This WASD movement is linear in 4 directions to prevent issues with getting stuck in diagonal movement and to also make it easier to attempt the puzzles we have, such as crate pushing and platforming.

    - Improved controls menu in main menu
        - Our testers both struggled figuring out how to deal with our shooting mechanic. One tester wasn't sure what to aim at despite knowing to aim at something. The other tester had a much more difficult time figuring out what to aim at and why.
          As a result, we improved the controls menu in the main menu for players that first start the game. The controls menu is a series of pages of what to do with pictures to show the player exactly what to look for. This should visually clarify how to play our game.

    - Clearer navigation
        - Playtesters got lost in the beginning of the first level near the forest. They weren't sure which direction to go and couldn't progress until they eventually figured it out by trial and error of exploration.
          We decided to add more props to the start of the game to build an artificial wall so tell the player where not to go. For example, there are some fallen trees acting as a barrier. In addition, there are firefly particle systems that build a path to the first checkpoint, designed to hopefully have the player want to follow the fireflies.

- Improvements from Shaders:
    - Shader for tiling the stone textures
        - The stone texture is a common texture in our game for any castle-like walls. These textures are applied to a prefab of a small wall and we use

this prefab often. We often scale this wall but this caused the textures to either be too small or too large and so we needed a consistent way to apply the texture. The AutoTiling shader we applied makes sure that the texture is always applied at a constant size no matter what scale the wall prefab has.

    - Shader for making models into wireframe
        - Initially thought it could be used as a comedic effect, but proved useful in development rather than gameplay since I wouldn't have to render a lot of polygons, making the frame rate easier on my computer for development.

    - Lighting diffuse for player
        - I wanted to make the player respond to light around the map a bit more realistic. With a diffuse shader the player's lighting and dark areas are responding better to the angle from which lighting is coming from.

- Other Improvements:
    - End-of-level screen
        - We needed a way to let the player know that they beat the current level and to give them a chance to see their current score before progressing.

    - Lives and ninja star counter
        - We needed a way for the player to know how many resources they have remaining, in terms of lives and available shuriken. We put this on the UI of the gameplay so that it is always visible.

    - Saving/loading game data
        - Our game has multiple levels, and so to allow the player to complete it over multiple sittings, we needed a way for a player to save their progress and continue from it during their next session.

Beta Feedback and Public Demo (Assignment 10):

What we learned from our beta release and from our public release:

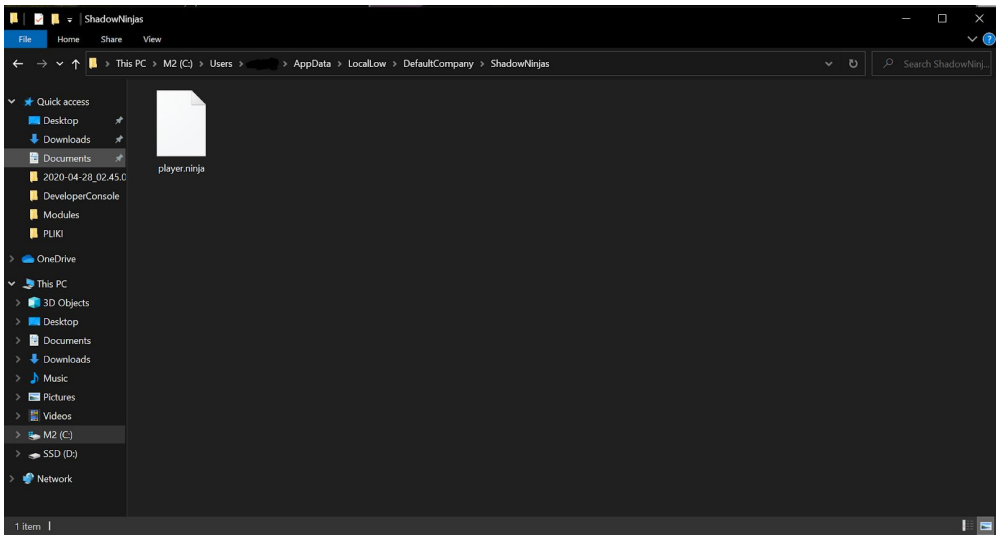
- We learned that different people see and do things differently in the game than what we believed to be natural for a player to want to do. Our mindset as developers moved us away from thinking like players who have never seen the game before and playtesters reminded us we cannot continue to develop the game in such a way. In short, we as

developers need to constantly think like playtesters when it comes to balancing our game.

What we are most proud of about our game:

- We are proud of several features in our game. For one, our aiming mechanic was very lackluster, as it required the player to orient themselves in order to hit a target accurately. Our lock-on mechanic makes this much less stressful for the player.

Another feature we are proud of is the player stats that are logged in an encrypted binary file that cannot be modified outside of the game. Aside from saving the player's data, it prevents any user from cheating.



Finally, we are the most proud of the fact that our project actually resembles something of a working, finished game. We have multiple levels, a theme, menus, animations, and much more that can be shown off and played for fun!

What changes we made to our original design for technical reasons and why:

- Our game did not feature any form of save data, and so when needed to incorporate loading levels we had to spend some time rewriting the player scripts to allow for saving the data that is happening in game, such as shuriken count, player lives, and current level.

In addition, our game is heavily based on light sources. However, by default Unity only renders up to 4 light sources and so we needed to force Unity to allow for more so that our game does not look like some torches are lit and others are not.

What changes we made to our original design for playability reason and why:

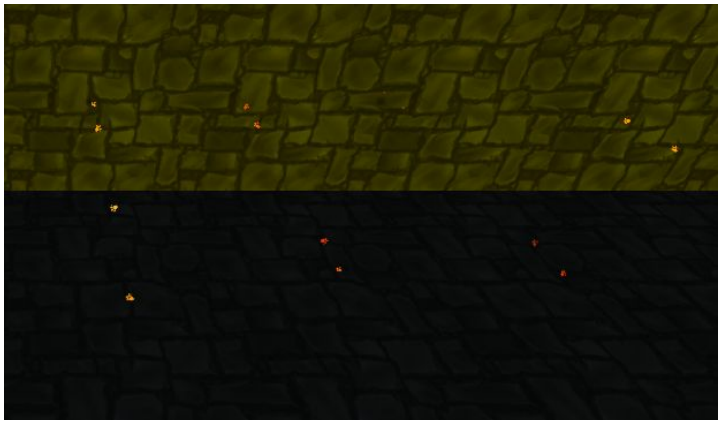
- Our biggest complaint during early stages of playtesting was the movement being too clunky. Our player only moved in two directions, while also rotating in two directions. This was unnatural for players and so we redesigned the movement to allow for traditional WASD movement in 4 directions. Preventing the player from moving in diagonals also allowed for an easier time platforming or navigating a level.
  Our second biggest complaint was aiming, and so we created the lock-on feature as mentioned above in an earlier question.
  Lastly, all players had difficulty navigating a level from spawn because it isn't necessarily clear on which direction to go given the camera angle and so we have each level start with a camera pan from the goal to spawn to indicate the correct direction.

What we learned from our playtesters and what changes we made as a result:

- **Playtester #1**: Michael Wiszowaty
    - Time to complete first level:
        - Only played for 5 minutes; almost beat level
    - Feedback provided:
        - START text at spawn visible through walls
    - Observations made:
        - Explored spawn entirely, despite it being intentionally empty to persuade player to move elsewhere
    - Changes made given feedback:
        - Added a camera pan from goal to player spawn at the start of each level to let the player know where to go. This should prevent the player from needlessly exploring the spawn for the way forward.

- **Playtester #2**: Marcos Gonzalez
    - Time to complete first level:
        - 14 minutes played, made it up to first part of second level
    - Feedback provided:
        - Difficulty making accurate jumps that required diagonal movement
        - Some parts of game too dark; recommended brightness adjustments
    - Observations made:
        - Even after reading controls menu, still got confused on how to hit torches
    - Changes made given feedback:
        - Improved platformer level design to reduce need for diagonal movement. Also adjusted brightness in some areas where it was too dark so that the player can see more easily.

What we would do next if we had more time:

- We feel that our movement could use more work, such as possibly making it 8-directional to allow for more dynamic movement. In addition, the player model can get stuck in models momentarily when jumping and colliding which reduces the feeling of fluidity.
  The archers that shoot arrows are currently working, however there are some issues such as arrows being to pass through walls (despite not being coded to do so) as well as archers only being able to shoot in one direction despite their rotation. Improving these things would allow for more interesting levels and enemies.
  
  Lighting is another thing that could always be improved. Our game heavily relies on lighting and so creating more light source models, as well as adjusting light to be not too dark or not too bright would also be a plus.
  Interactivity is something we also wish we could improve. Currently, the player can extinguish torches and push crates but we wanted to create many types of interactivity for all kinds of puzzle ideas and designs.
  Lastly, animations, art, and sound are all things that are difficult to make or even find. As a result, our game has many reused textures and props. To prevent staleness and to

promote more level themes, we would have liked to increase the amount of variety in the animations, art, and sound in addition to improving the current selections of each.
We'd also like to make something special happen when the player is caught, such as all the enemies chasing the player rather than instantly resetting the level.

What we would do differently next time:

- Our game would have definitely benefited better from a bit more planning in terms of level design. Originally, we all worked on the same level but this caused many merge conflicts. To remedy this, we did begin working on individual levels but this prevented anyone from having a clear plan on how to build their level, and so the levels are mostly free-styled. We should have clearly written and diagramed what we want each level to be in terms of theme, puzzles, length, and more. This would allow us to know what assets to look out for, and what scripts to write.

Sources

[post links to any code, assets or useful tutorials used that were not made by you]

- Stone Texture
- Fire + Smoke Particle Textures
- Crate Texture
- End Screen Tutorial
- Movement Tutorial
- Barrel Texture
- Bow and Arrow Textures

- Bow and Arrow Tutorial
- Bow Fire Sound Effect
- Torch Burning Sound Effect
- Grass step sound effects:
  - 1
  - 2
  - 3
  - 4
- Wood step sound effects:
  - 1
  - 2
  - 3
- Stone step sound effects:
  - 1
  - 2
  - 3
  - 4
- Music:
  - in-search-of-asia
  - Windy Trees Blowing Stereo A1 Sound Effect - wind background
- Shuriken Model
- Skybox
- Font
- Shuriken Crosshair
- Shuriken throw sounds:
  - 1
  - 2
- Gong hit
- Castle fortress
- Nature props
- Ninja Life Icon
- Village background
- Forest background
- Save/Load Tutorial
- Ammo pickup sound

CONTENT BELOW THIS LINE IS NOT PART OF OFFICIAL DESIGN DOCUMENTATION

Game Details (subject to change)

Menus:

- **Main Menu**:
    - New Game: erases any saved data and starts a new game
    - Continue: resumes game from last save point
    - Level Select: takes you to level select menu
    - Options: takes you to menu for viewing controls, audio/graphics settings, etc…
    - Exit: exit the game and return to desktop

- **Pause Menu**:
    - Resume: exits the pause menu and resumes gameplay
    - Restart: restarts the current level
    - Options: View Controls
    - Save and Quit: saves progress and returns to main menu
        - Saving only records the level reached, not the progress done in that level. Basically after saving and later playing again, the game takes you to start of that level

- **Options Menu**:
    - Show controls of game with a small image of keyboard (or controller)
    - Adjustable bar for audio level(s) (assuming one for sound, one for music)
    - Adjustable dropdowns for graphic levels such as screen resolution

- **Level Select Menu**:
    - Shows all “worlds” of the game (think New Super Mario Bros)
    - Clicking a world reveals a list of levels from that world
    - Clicking a level starts that level

Player Resources:

- **Lives**: based on remaining Shadow Ninjas
    - Running out means game over, must restart game from level one
    - Cannot gain more lives during gameplay

- **Shuriken**: throwable ninja stars that can be used to interact with the level, such as cut ropes, knock out enemies, extinguish torches, etc…. Limited amount each level.

- **Dragon Firecracker**: end game weapon, not controlled by player except for activating it

Level Design:

- ~~Three main "worlds", ~3-5 levels per world for ~12 total:~~
- **!!! It appears that the professor wants lots of quality in levels vs. quantity of them, and so we will not be able to make ~12 short quality levels but rather ~3 very good levels with lots to do. However, the concepts below can still be used. !!!**
    - <u>World 1: Enemy Outskirts</u>
        - Levels themed to be in forests, presence of enemy patrols/wagons, etc..
        - Not many threats for player yet; trying to teach the game and ease player into it
    - <u>World 2: Enemy Camps</u>
        - Levels themed to be closer to enemies, so watch towers, tents, campfires, supply crates stacked everywhere, etc…
        - More enemies present and more light sources as a result
    - <u>World 3: Enemy Castle</u>
        - Level inside/around main enemy castle
        - Much more enemies and much less shade to hide behind

- Individual levels must contain some aspects of platforming, puzzles, and shadows/light interaction

- Puzzles don't need to be complicated or too involved, but rather keep things simple and logical. Examples:
    - Pushable crates to block enemies/unblock path
    - Enemies light up unlit torches, so player must quickly extinguish torches and move past them before they are relit
    - Distracting enemies in some way to move past them
    - Waiting for an enemy to walk into a tent, throw shuriken at tent to collapse it, and walk by as enemy is trapped in tent

- Levels aren't expected to be large and long, but rather small and plenty of them… think size of a soccer field

Gameplay:

- A new game starts with still images telling the story as described above in the Story section. Each image is the background of the screen with the explanation in text. Each image transitions to a new one with a fade; after < 5 images, the first level begins.

- "Birds-eye" view camera, but not straight down. Imagine if the level fit inside a cube, then the camera would be near an upper corner of the cube

- An individual level might have "sections", where the camera pans over to a new part of the level after the player reaches a certain point

- The end of each level should be relatively obvious, marked in someway that makes sense, but finding a way to get to it should require some thought and coordination

Visuals / Sound:

- Cartoonish graphics for easier modeling and texturing; I'm thinking something like this game: https://www.youtube.com/watch?v=tey4Bm2Ymm4

- Cartoon violence, as this game is aimed for younger audiences) so no blood or anything scary

- Audio should be relatively slow and stealthy sounding, similar to 20:50-21:20 of this: https://youtu.be/HygYI5QIM1Q?t=1250 (in general this soundtrack has some good stealth tracks to gain inspiration from)