

CHEEZ PLZ

<http://wac.450f.edgecastcdn.net/80450F/kbat.com/files/2014/01/Swiss-Cheese-credit-iStock-176841575-630x429.jpg>

Game Name: *Cheez Plz*

Team Name: Team Fog

Team Members: Vincent Doan, Carlos Aguirre, Jimmy Wu, Leon Matsuki and Erwin Tung

Overview

Cheez Plz is a cute 2D, single player puzzle/maze game. The game is in third person with a top-down point of view. *Cheez Plz* also has 2D stealth game elements where the player must avoid being detected and caught. The objective is all about getting the cheese and returning the cheese to your mousehole as quick as possible. Be warned! There are patrolling kittens on the prowl ready to send you packing back to your mousehole!

There are quite a few genres incorporated into our game. Most notably it is a 2D, top-down puzzle game. The player will be venturing through a series of maze levels with hidden cheese. The objective is to move the mouse character onto the cheese and return back to their mousehole starting point. Other than the challenges of the maze, the character must avoid the patrolling kittens. The kittens have a specific patrolling patterns controlled by AI and will chase the player's mouse once detected within a certain field of vision. If the cat touches the mouse, the player is respawned back to the beginning of the map with the clock still running. The player has infinite opportunities to win the level, however your completion time will be rated upon the three-cheese system. So go grab that cheese as quick as you can!

There are many video games that influence this game. Many of the influences are drawn from 2D puzzles. A couple important influences of the game are *Transformice* and *Metal Gear Solid*. *Transformice* is a 2D multiplayer networking

puzzle platformer game. Other than the obvious influence of the mice, the puzzle elements are a basis of the game. The player will have to solve multiple puzzles within the maze to gain access to certain areas. *Cheez Plz* also draws heavy influence upon *Metal Gear Solid*'s third person stealth gameplay mechanics. The patrolling kittens will be the main opponents looking to catch the player, creating a more challenging experience.

The art will be 2D sprite work for all objects of the game. The sprites will keep to the cute and cuddly themes of young cats and mice playing a game against one another. Other than innocuous and fun-loving tones, quirky science fiction themes will also exist throughout the game. The setting of the cheese factory will have some impressively advanced machinery that the cats and mice have constructed.

The game will be programmed in C# using the 2D UNITY engine. Much of the physics will be adapted from UNITY with customized approaches in specific object behavior. The targeted platform will be Windows PC computers. Finally, the game will be played with simply a keyboard.

Background Story

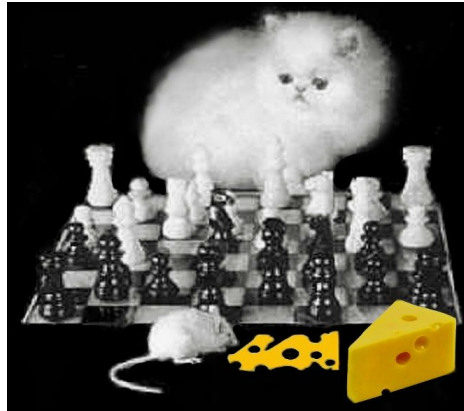


http://heroarts.com/wp-content/uploads/2015/03/card_5552.jpg

The story takes place in a fluffy world where cats and mice are friends. Cats and mice work in a factory together to produce delicious cheese. One special day, the cats and mice gather to celebrate. The youngest mice and cats play a game of

cops and robbers where the cats protect cheese while the mouse try to bring the cheese back to their base. This in turn helps teach the younger mice and cats their basics instincts and motor skills. It teaches the younglings how to forage for cheese ingredients and how to look out for another. At the end of the day, everyone gathers together for a great cheese feast.

Gameplay & Balance



http://2.bp.blogspot.com/-rgoSpDqrOAc/UszWV-Lxbxl/AAAAAAAAATZc/af-lLv2jpQw/s1600/cat-mouse-chess_00.jpg

In *Cheez Plz* the player can do several things such as traversing through the maze, interacting with objects and switching between two different modes: “sneaking mode” and “normal mode”.

During the game there will be several boxes that the player can pick up and move. Some of these boxes will consist of a specific power-up and will appear in specific places along the maze. In addition, there will be traps that will also be located in specific places throughout the maze, forcing the user to be very careful when maneuvering about the maze. Normally, these traps will be located in places the player should seek to avoid. For example, areas with gum traps will be close to pitfalls. Throughout the maze, there will also be environmental boxes that the mouse can push and “grip” onto with their teeth using the interact key. This interaction will in turn give the player the opportunity for more flexible puzzle solving such as filling pitfalls to walk over.

One of the interesting characteristics of *Cheez Plz* is the sound specifically composed and recorded for the game. The music found online will carry the cute and whacky tones of the game. Other than the background music playing during the traversal of the maze, there will be music if a cat is alerted from your presence, as

well as a victory theme. In addition to the music, the player creates imaginary “audible” footstep noises. If the player is near any cats when not walking carefully, and within the range of a cat, there is a risk of alerting the cat. To avoid making sounds (though at the cost of movement speed), the player can enter “sneaking mode” by holding down the left Shift key.

Speed is also a major component of the game that needs to be balanced. It is such an important aspect that will affect the gameplay feel and challenge. The cats are placed by challenging obstacles, and will be alerted if they spot the mouse. It is highly advised to break line of sight and run as fast as you can after being spotted. The player must use their guile and knowledge to manipulate the elements of the map to succeed.

When balancing debuffs and traps, the level must be taken into account placement and the appropriate difficulty. The speed of the mouse when moving through the level while carrying the cheese is also critical to balance for user experience. While carrying the cheese, speed is reduced and getting back to your base is even more dangerous.

Another important feature of *Cheez Plz* is the puzzle system. The player will be often tasked with finding the correct keys to open doors that block off sections of the maze. The player can pick up these keys through simply moving onto the sprite of the key. After picking up the key, the player can simply collide into the correct door to open it up.

The player will also be challenged by environmental factors throughout the game. These challenges will include: pitfalls that the mouse can fall in, bubblegum traps that have a slowing effect, and finally electrical traps that briefly stun the player.

Characters



http://comps.gograph.com/cartoon-doodle-of-cat-and-mouse-for-coloring_gg61138386.jpg

There will be two characters in the entire game. The main character is the lone hero mouse controlled by the player. The mouse and player's fluffy opponents are the kittens trying to catch him.

1. Mouse (Player Controlled)

- The mouse will be named by the player. At the end of the level, the name and the best completion time will be displayed for that particular level. Voiced by Vincent Doan.

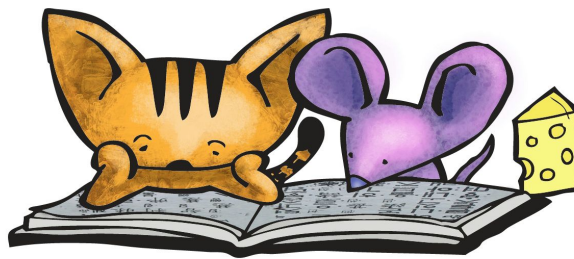
2. Cat (Opponent/AI Controlled)

- Cats come in two forms. The sleeping/idle cat and the awake/active cat.
 - i. Sleeping/Idle Cats
 - The sleeping cats are idle and won't pursue a player's mouse if still sleeping. If a player is not in sneaking mode and movement noise is within range, the sleeping cat will awaken and become an active cat. Once active, the cat will patrol and pursue the player on sight. After evading the cat for a certain amount of time, the cat will go back to its former napping location and go back to sleep. Sleeping/Idle cats *do* appear on the minimap.

ii. Active Cats

- The active cats take the role of guarding the cheese. They will patrol in a set pattern and only break from that pattern if they spot the player or detect suspicious activity. After a certain amount of time of pursuit or investigation, they will return to their original position and resume their former patrolling pattern. Active cats appear on the minimap as a smaller version of themselves.

Rules & Mechanics



<http://www.pvld.org/sites/default/files/legacyblog/uploads/2013/05/Murphy-Cat-and-Mouse-Reading.jpg>

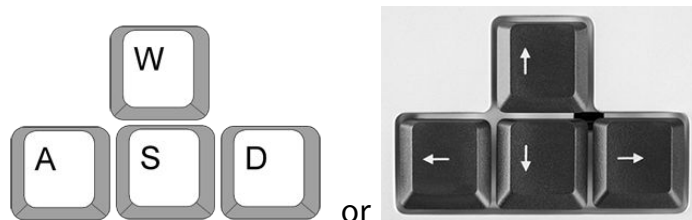
1. Game Rules

- a. Return to spawn point if a cat touches the mouse
 - There will be several kittens in the game. If anyone of these kittens catch the mouse, the mouse will returning to the spawn point with the game timer still running upward.
- b. Get the cheese to complete the level
 - The rules of the game are simple. The player plays as the mouse and must locate the hidden cheese somewhere in the maze. The player must then bring the cheese back to their mousehole, the starting area. Along the way there are traps that hinder the player and puzzles to solve. However the most dangerous opponents are the patrolling cats. If the cats touch the player, the player will be sent back to their mousehole empty handed. The game clock keeps running upward despite the player

respawning. The player will have infinite chances to finish the level.

2. Mechanics

- a. A sneaking mechanic to avoid alerting an idle cat.
 - There will be a special movement assigned to the Shift key to change how the mouse moves. The mouse crouches, moves slower, and walks silently. When the player enters “sneaking mode” the mouse will walk slower and move silently. The sneaking mode mechanic is signified by the player’s mouse being transparent.
- b. The Interact button
 - Mapped onto E as a default, this button will be for various puzzles that the player will encounter as they progress through the maze. The mechanic here will allow the mouse to grip and move boxes as well as utilize other objects in the game.
- c. Mice cannot walk through walls
 - The mouse will not be able to walk through walls, the player is instead forced to traverse and solve the puzzles within the maze.
- d. WASD/Arrow Key Movement.
 - The WASD controls means W will move the mouse upward, A will move it left, D will move it right, and S will move it downward. Arrow Key controls will obviously follow the direction that the keys have.



In-Game Controls	
Move Left	A / Left Arrow
Move Right	D / Right Arrow
Move Up	W / Up Arrow
Move Down	S / Down Arrow
Sneak / Crouch	Shift
Interact	E
Restart Level	R
Menu Controls	
Open/Close Menu	Escape
Select Menu Item	Left Click

3. Score Systems

- After the completion of the level, the score system will be based on how fast the player can beat each of the game's levels. Each level will have set finishing times for each of the 3 cheese ratings. The rating the player will receive will depend on how quick their time is compared to the predetermined cut off points (See picture below for example). A timestamp, along with the username, and three-cheese rating will display the player's performance.
- a. Record High-Scores per level

- If we have time, one of the menu options will be the high scores. The high scores will access the local machine's best level scores. The player will be able to see the 10 top scores of each level including completion time, three-cheese rating, and the player's name.



- In the picture above, the cutoff time for the 3 cheese rating is 2:00, 2 cheese rating is 2:30, 1 cheese rating is 3:30.
- For now it just shows the cheese rating for the level the player has completed.

b. Faster time = higher score

- Rather than using a game over system, a level of *Cheez Plz* does not end until the player decides to quit the game or finish the level. To this extent, in order to separate the good players from the great, we will be implementing a three-cheese rating system that will rate how fast the player finishes the level. The local high score three-cheese rating system will let various players know how well they did in each level.

c. Timer keeps ticking up even if you respawn

- The timer will continue ticking upward even if the player is caught by a kitten or fallen into specific traps that respawn the player back to the beginning of the maze.

4. Mini-Map

- The game will use a basic minimap that will displayed in the top-right corner of the screen. On the minimap, the player will always appear as a shrunken version of them mouse signifying the direction the player is facing in respect with the maze. This is intended to help the player navigate through the maze.
 - a. The mini-map is always present in top-right
 - mini-map will always appear in the top-right of the game's window. It will be static and the player will not be able to remove it.
 - b. The mini-map does not have an adjustable size or any zoom functions
 - The mini-map only displays the immediate surrounding area to the mouse up to a specific range. This means the player cannot look too far forward or backward using the mini-map.
 - c. Mouse will always appear on mini-map
 - The mouse will always appear and be in the center of the mini-map. The mouse will be represented by a smaller version of itself.
 - d. Cats appear in mini-map
 - Patrolling cats and sleeping/idle cats will be shown on the map just like the player. Cats will be represented as a mini version of themselves.

5. Items

- a. Boxes located throughout the maze
 - Throughout the map, there will be boxes that the player can move to their advantage.
 - There will also be power up and debuff boxes that the player can run through and pick up.
 - 1. Power Ups
 - Power ups are intended to give the player a slight boost to get through the level a little bit faster.
 - 1. Increased Speed (5 sec)

- This power up will include a brief increase in the player's movement speed for 5 seconds

2. Debuffs (If time permits)

- There will also be debuffs that will be spawned by active, patrolling cats.

a. Decrease Speed (5 sec)

- This will attempt to slow down the player's movement speed.

b. Doors & Keys

- Most levels will consist of doors that can only be unlocked with a key. The doors will be located in specific locations in each level and the keys will be scattered in specific locations in the maze for the player to find.

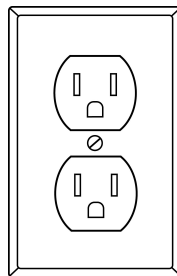
6. Traps & Environmental Effects

- The maze itself will not only consist of cats and puzzles. There will be a number of traps and environmental challenges that will slow the player down. The traps are planned to be simple, obvious, but potentially devastating to the player.

a. Avoid them!

b. Electric Mouse Trap

- If the player runs into an electric mouse trap, they are stunned for 1 second and are vulnerable to cats.



http://sweetclipart.com/multisite/sweetclipart/files/electrical_outlet_lineart.png

c. Adhesive Bubblegum Trap

- If the player runs into an adhesive trap, the mouse will be slowed down within a certain radius of the trap. If the mouse

leaves this radius, it is returned to full speed and is flung forward for a temporary speed boost.



<http://gloriagarvey.com/wp-content/uploads/2012/05/abc-gum.jpeg>

d. Pits

- Pits are mouse sized traps that will be placed in various positions throughout the level. Walking over a pit will cause the mouse to fall in and be forced to restart from the spawn point. Due to their small size, cats will not be able to fall in. Pits can be filled in with boxes! Once a pit is filled with a box, the mouse can walk over the newly filled pit without any penalties.



https://image-gr.s3.envato.com/files/33628151/hole_preview.jpg

Artwork & User Interface

The look and feel of the game should be cute and cuddly(keeping in mind that the artwork will be done by novices). Everyone in the game are pals playing a game against one another and the level design of the cheese factory reflects a

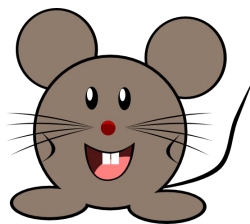
scientific yet adorable theme. The art assets will be custom 2D sprites for the level, background, mice, and cats. The artwork will be made through computer software such as MS Paint and Adobe Photoshop as well as some lifted through the internet. The setting will take place within a massive cheese factory, while the design centers around cute, fluffy, and harmonious themes. There are obvious light-hearted and comedic tones in the design focusing on small, round characters and objects. The characters and background exhibit a soft and fluffy nature, wrapped in a pastel of colors. Beyond the cuddly themes, there will also be tongue in cheek science fiction elements. The cats and mice of this game have become intelligent enough to engineer and operate the cheese factory, while creative enough to preserve the cute architecture they are accustomed to. There is limited UI other than a minimap and notifications overhead objects in the game. The game will be intuitive and simple enough to understand. The player will learn about how to play the game through intuitive level design as well as a good tutorial level.

Boots of Speed Sprite - League of Legends

<http://i.ytimg.com/vi/WMXcN7S8gV0/hqdefault.jpg>

Bubblegum Trap Sprite - Trend Monitor

<http://trndmonitor.com/wp-content/uploads/2013/07/gum.jpg>



<http://www.clipartbest.com/cliparts/niE/Egy/niEEgybKT.png>



http://img13.deviantart.net/bf7e/i/2012/276/f/1/penburokuu_the_scottish_fold_by_nyatto-d5gcood.jpg

Music & Sound



https://storybird.s3.amazonaws.com/artwork/Cat_alogue/full/who-let-the-mice-out.jpeg

The music is picked from quirky and cute themes. The music ratchets up when a cat has spotted the player. The sound effects are tongue in cheek and completely voiced by the development team, acapella.

Game background Music:

[Menu Music] Bubbly Clouds by Jun Ishikawa:

<https://www.youtube.com/watch?v=8zyUkAVFqw4>

[Tutorial Stage] Puyo Pop Fever Rules by Hideki Abe:

https://www.youtube.com/watch?v=v9NFZAr3hcs&index=4&list=PL_7SZhTmUFWl8W-LwV9Kf-0Q00ltBqlyl

Nyan Cat Nyanyanyanyanyanyanya!" was uploaded by "daniwell":

<https://www.youtube.com/watch?v=QH2-TGUlwu4>

[Level 1] Thriller by Michael Jackson:

<https://www.youtube.com/watch?v=sOnqjkJTMaA>

[Level 1] Indiana Jones Theme by John Williams:

<https://www.youtube.com/watch?v=GzLZJaAm2hw>

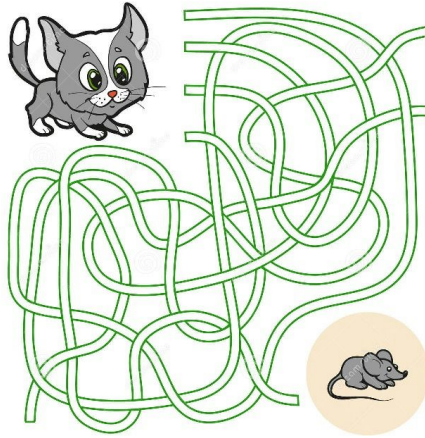
[Level 2] Playing with Danger by Valve Orchestra:

<https://www.youtube.com/watch?v=llOvd2yGEio>

The game will have the following sounds effects:

1. Background Music
2. Cat Alerted Theme
3. Mouse Moving (very quiet)
4. Door Opening
5. Door Locked
 - If the user try to open the door and doesn't has the key
6. Picking Up Keys
7. Picking Up the Cheese
8. Picking up Power ups/debuffs
 - There will be a sound effect when the user mows through a power up box and gains a power up or a debuff box
9. Traps & Environmental Effect Sound
 - Each trap & environmental effect will have its own sound. For example, the vacuum will have the sound of a working vacuum.
10. Cat catching the mouse
11. Level Complete

Levels



<http://thumbs.dreamstime.com/z/maze-game-children-cat-mouse-48089309.jpg>

Cheez Plz is a story mode game that consists of 3 different levels as well as a tutorial to introduce the game to the player. Each of these levels will have different degrees of difficulty, amount of enemies, boxes, and sizes of the maze. In addition, each of the game levels will have a mini-map of the immediate area around the player. The mini-map will be very useful for the player to traverse the maze.

This mini-map will be located in the top-right part of the screen and will be always there. The player will always appear on the map as well as all the cats, the cheese, doors, keys, traps, and puzzles. (For more information about the mini-map and its options go to Rules and Mechanics section number 4: mini-map)

For each level there will be cheese somewhere in the maze, thus the mouse will have to find it and bring it back to his mousehole to finish the level. When the level is finished, the game will record the player's time to complete the level and show the rating the user received out of 3 cheeses (0 being the lowest). A story cutscene will appear showcasing the player's progression before finally moving onto the next level.

If the player is caught by a specific trap or kitten, the player will automatically return to the spawn point. There are several ways a player can be returned to the start point such as being sucked into a vacuum, touched by a kitten, or caught in a specific trap. (For more information of how to restart the game go to Rules and Mechanics section number 6: Traps & Environmental). Upon respawning, the clock timer will continue to increment upward and the player will have infinite chances to complete the level. The player can optionally quit the level and restart by pressing ESC, going back to the menu screen, and selecting the level once again. There is also

a shortcut where the player can press R and restart the clock and redo the level. Finally, some of the three levels will have a diminished field of view for heightened challenge.

1. Tutorial Level:

- In this level we will introduce the most important characteristics of the game, so the user will learn all the essentials of gameplay. We want the user to learn the following things:
 1. How to move along the maze using WASD controls or arrow keys.
 2. Using combinations of movement keys to effectively traverse the maze.
 3. Interact with Boxes
 - Pushing boxes after picking them up
 - Be able to pick-up some box with some features. (Power-Ups)
 4. Use of “sneaking mode”
 - Crouch and walk silently to not alert any cats.
 - Sneaking behind boxes will make cats unable to see you
 5. Picking up keys by moving the mouse onto them.
 - The key will be displayed at the top of the screen.
 - Open a door by first picking up a key and moving into the door.
 6. Interacting with enemies and avoiding enemies
 - Running away from enemies
 - Sneaking past enemies
 7. Snatch the cheese and come back to your mousehole
 - Win the game

2. Level 1:

- This is the first level of the game *Cheez Plz*. With the story cutscene being shown after beating the level. This level is considered the easy level. Some of the characteristics this level should have are:
 1. 1 Idle Cat and 1 Active Cat
 - The idle cat and the active cat are in easy positions to avoid.

2. 1 Door with his respective Key
 - The Door and the key will be relatively close
3. A small maze
4. Boxes
5. Pitfall and gum traps

3. Level 2:

- The user will be shown a cutscene at the end of the level with medium difficulty.

It should have the following things:

1. 3 different puzzles and a finale room
 - One room will consist of solving a sound puzzle, where the player must match the pattern of sounds played by stepping onto the buttons in the correct order.
 - Another room will consist of a number puzzle, where the player must watch the pattern of the numbers and step onto the number buttons in the proper order to fill in the blanks.
 - Last puzzle is a pitfall puzzle where the player must choose a path from 2~3 choices and follow it precisely without falling into surrounding pitfalls in order to reach the destination that deactivates the pitfall traps.
 - The finale room will expect the player to combine the knowledge from previously shown traps and dodge the cheese bullets while avoiding pitfalls and timing each traversal carefully before making a run for it.

Scripts

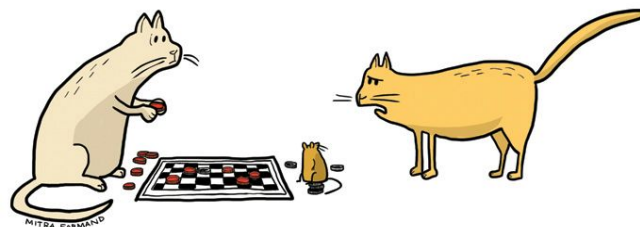


<http://previews.123rf.com/images/worldofvector/worldofvector1507/worldofvector150700055/42895706-Happy-Friendship-Day-Cute-cartoon-red-cat-with-mouse-Talk-think-bubble-Card-Kids-background-Flat-des-Stoc-k-Vector.jpg>

There will be random lines for the voice actors to speak. The lines are voice acted sound effects which are triggered when falling down holes, being spotted by cats, opening doors, getting keys, getting cheese, and winning the game.

There will also be text-based interfaces notifying the player of being spotted by a cat, finding a key, opening a door, falling down holes, acquiring the cheese, and winning the game.

Cutscenes



"This is not how normal cats play with mice"

<http://mitrafarmand.com/wp-content/uploads/2015/02/cat-and-mouse-and-checkers1.jpg>

We're planning on having single shot cutscenes as a treat to showcase what has transpired after a level has been completed. We're hoping to have an ending with a credits roll showcasing all the team members roles for working on the game.

Artificial Intelligence

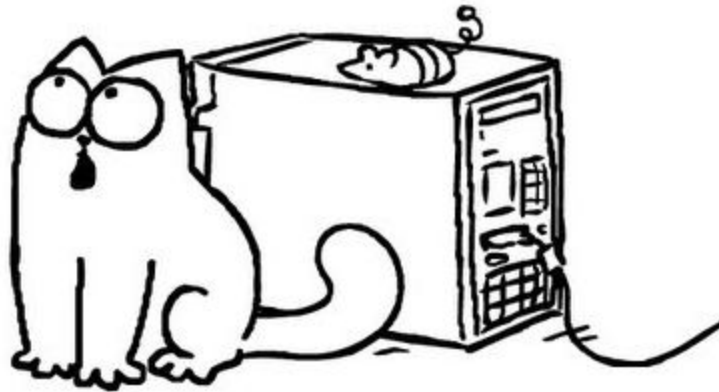


<http://thumbs.dreamstime.com/t/cat-computer-robocat-8524712.jpg>

The artificial intelligence utilized in Cheez Plz is not particularly sophisticated. A standard AI pathfinding algorithm is utilized to allow the cat to perform 2 essential tasks:

1. Chase the player when they get within a predefined distance from any Cat.
2. Patrol certain regions of the map by defining 2 waypoints for it to navigate back-and-forth.

Technical Specifications



<https://i.ytimg.com/vi/BWIPZvwcnX8/hqdefault.jpg>

Cheez Plz is a game designed for standard desktop PCs running Microsoft Windows. We may include Mac as a supported platform since Unity allows easy deployment for multiple platforms. Our game is also quite simple and is made in 2D, providing very low hardware requirements to play the game. Consequently, there is a heavy focus on using the keyboard to control and navigate the gamespace to traverse all the maze environments. Players are able to freely switch between using the standard WASD or arrow keys to move their player.

As mentioned earlier in the design document, users play mice in a maze. The opponents are cat NPCs which patrol the maze and attempt to foil the mice from getting their delicious cheese. The cats' movements are driven by an ensemble of algorithms designed to encourage more interactive and engaging gameplay. These algorithms are discussed more in-depth in the Artificial Intelligence section above.

To take advantage of an already large, supportive community, *Cheez Plz* is developed around the well-known and battle-tested Unity game engine. The breadth of features provided by Unity is helpful in having physics, sprite support and other necessary features in one package. Others have already developed and provided to the community, implementations of various features and helpers for Unity that *Cheez Plz* utilizes such as the A* algorithm. Unity's reliance on delegation is also appealing because it allows us to simply point to customized methods for events and actions. Its use of C# is also a significant help, as our group is well-versed in Java-like languages and the use of the Mono framework greatly facilitates

cross-platform development – though our target platform remains Microsoft Windows – since our development team is split between Windows and Apple Computers.

Game Objects & Classes

Due to the modular nature of Unity and time constraints. Objects in Cheez Plz tend to be standalone classes that simply have a script assigned to them. All players in the game—whether user-controllable or not (*i.e.* NPC)—are defined as a subclass of `MonoBehaviour` and implement the various Unity delegation methods, plus any object specific methods and variables.

Each level of the game features unique Objects—known as `GameObjects` in Unity—that either build upon basic traps we designed and introduced earlier in the game or are one-off objects that exist in a single level. For instance, the Indiana Jones-esque “giant ball of cheese” is found only in Level 1. Several Object persist for the lifetime of the game and are reused without modification. These are the player’s mouse, cats and the various standard traps—boxes, stuns and so forth. Their respective Unity scripts provide all the necessary interaction for any level. Case in point, the mouse needs only to be driven by the user’s keyboard controls. The traps and other objects then rely on collisions with the mouse to handle an event. Similarly, the cats’ movements are defined in a Unity script and recycled for each level. Any modifications need to a cat’s behavior are determined by publically exposed variables found in the cat’s Unity script.

Handling Interactions

Cheez Plz relies heavily on 2D colliders determine when events occur and how to act on them. A case in point are the pit traps that are found in every level. There are two kinds of pit traps, ones where a player can push a box into them (*i.e.* fillable) and those that the player just falls into and dies. While our implementation separates them out, they both remain essentially a simple event handler for detect a collision depending on whether or not the `GameObject` is a mouse or a box. If it determines that the collider is a mouse, it will call the method to kill the mouse

otherwise, if it's a fillable pit, it will allow the box to be dropped in. This behavior is fairly standard for virtually all of the interactions with objects in the gameworld. Objects are colliders that hit objects, check what they have collided with and respond accordingly.

Animations

The player's mouse and the cats utilize the Unity animator to help showcase their movements during the game with custom sprites created by Carlos Aguirre. To do this, our team chose to use the Unity state machine animator to ensure that both the player's mouse and the cats would have smooth transitions as they changed positions in the maze.

Algorithms

As mentioned earlier in the Artificial Intelligence section, the cats are NPCs and find their way to the player or predefined waypoints by means of the classic A* pathfinding algorithm. The A* algorithm is used to control all of the cats' movement whether it be chasing the player or simply patrolling a certain area; in order to patrol a region of the map, a destination on the map is assigned for it to traverse back-and-forth from its initial location at the start of the game until it observes the player, in which case a cat will start chasing the player. Despite the ease of implementing A* and a heap in C#, the game utilizes a free Unity Asset known as PolyNav2D to properly implement the AI. This is due to the unfortunate fact that Unity2D does not possess a built-in graph to use. This feature is found in Unity3D as the Nav Mesh, but because it interacts on (Y, Z) coordinates and Unity2D is based on (X, Y) coordinates it remains unported. Due to time-constraints and discussions it was felt unwise to try and develop a homebrew graph to apply A*. The Unity forums are littered with attempts to address this to varying degrees of success.

While we have other predefined behaviors. They are limited to checking for conditions or timeouts; the giant ball of cheese will reset to its start state anytime a player's mouse is killed and the bubblegum trap is a simple timeout.

Version Control (Backup)



http://www.aha.io/assets/integration_logos/github-bb449e0ffbcbcb7f9c703db85b1cf0b.png

The game source is kept in a git repository for version control and backup purposes using GitHub's cloud backend. The use of git is largely driven by its current popularity and wide-scale availability on sites such as GitHub or Bitbucket. Additionally, the ease of creating experimental branches to entertain potential features was a significant consideration in choosing the appropriate version control system.

Game Engine & Assets

- Unity 5
- Language: C#
- Built-in assets
 - UnityEngine.UI
 - System.Collections.Generic
- Manage assets in Github using modified .gitignore file
 - Ignores the Library/Temp folders, Visual Studio/MonoDevelop files
- Visual Studio 2013 will be used to edit C# scripts
- Sound effects made using Audacity and Bfxr

Game Rules (details)

Player Speed

- The player will start in "normal mode" by default, which means the speed is statically set to 3.5.

- When the player enters “sneaking mode” by holding down the left Shift key, this represents the mouse crouching and can move past sleeping cats at 75% of the original speed without waking them up. As soon as the left shift key is released, the player is returned to “normal mode”.
- The player speed will also be reduced to 75% when carrying a cheese, and will stack with the speed reduction from crouching.
- To show a visual cue that the player is in sneaking mode, the alpha level of the mouse sprite will be reduced by 50%.

Interacting with Items

- The player will be able to pick up boxes using the interact key, which is set to E key. The player can pick up the boxes as long as they are not holding anything else and is standing inside the BoxCollider2D assigned as the trigger for the box. When the box is picked up, the box stays with the player until the interact key is pressed again.
- These boxes will contain specific power-ups or debuffs that are designed to help the player complete the level more quickly or add challenge to the current level, and will not be random.

Power-ups

- Power-ups that the player may obtain from the boxes include speed increase by 25% for 10 seconds, sneakiness mode without the loss of speed for 10 seconds, mouse cardboard boxes that the player can use as a decoy for the cats, and others.
- Debuffs that the player may receive include speed reduction of 25% for 10 seconds, unable to enter sneakiness mode for 10 seconds, bubblegum trap, electric trap, or others.

Respawning

- There will be one set spawn point per level for the player, and the mouse will return to that position if the player falls into a pitfall or comes in direct contact with a cat.

- Currently the respawning is set to one second after the mouse animation disappears.
- If the player is holding the cheese and comes into contact with the cat, the cheese will be dropped at the location and the player will be able to retrieve it after respawning.

Timer & Scoring

- The timer will start as soon as the player moves on the level and stop when the player enters the trigger box while holding the cheese. These times will be directly used to calculate the cheese rating for that particular level and be saved locally onto the player's computer.
- The timestamp will also be displayed on the level selection screen on the menu along with the cheese rating for the level if the player has completed it, in the format MM:SS.
- The timestamp will stop after hitting 99 minutes and 99 seconds, where the player will be rated zero cheese if the player finishes the level at that time.

Game Scripts

1. Entity Scripts

- a. Player
 - Moving speed
 - Alive/Dead
 - Crouching
 - Can move
 - Carrying cheese
 - Carrying key
- b. Cats
 - Moving speed
 - Patrolling
 - Can move
 - Is chasing
 - Chase range

- Catch range
- Is sleeping

2. Camera Scripts

a. Follow Player

- Follows the player throughout the map, but doesn't rotate along with the player sprite when the player gets caught by the cat or falls into a pitfall.

b. Mini-map

- On the top right of the screen, we may either draw our own minimap based on the level or use a second camera with a further zoom highlighting the player and the cat to simulate a minimap.

3. UI Manager

a. Load Levels

- Contains a list of the levels and loads the correct level depending on which level the player cleared.

b. Dim Renderer

- Alters the renderer's alpha value to dim or brighten the screen on special events such as victory/defeat.

c. Dim Delay

- Contains wait method to delay the dim/brighten by certain amount of seconds.

4. Trigger Colliders

a. Cheese Collider

- When the player is in cheese collider, the player can carry the cheese.

b. Key Collider

- When the player is in key collider range, the player will pick up the key automatically.
- c. Door Collider
 - When the player is inside the door collider, the door will open if the player has acquired the key.
- d. Pipe Collider
 - The pipe collider will prevent the player from entering the pipe while carrying cheese.
- e. Conveyor Belt Collider
 - The belt collider will automatically move the character to the proper direction every Fixed Update.
- f. Pitfall Collider
 - The pitfall collider will play a death animation where the player spins and reduces the scale until it is not visible. Then the player will be returned to the spawn point.
- g. Win Collider
 - The win collider will check if the player is holding the cheese and play the victory screen, then move to the cut scene/next level.

Schedule and Personnel

Week 6:

Vincent Doan	Design 3 levels past the Tutorial level. Each of the levels should introduce a new concept (not a main mechanic) and should incorporate mechanics from previous levels to provide challenge to the player, while not making it too difficult.
Leon Matsuki	Implement Menu system for the title and & in-game menu. The menu should include play, high score, credits. Play will lead the player through series of three or more levels, tutorial will start the tutorial level, and credits will display the team/roles as well as

	credits to external assets if any are used.
Jimmy Wu	Implement timer system to be used to calculate total score. The timer will start the moment the player moves the character, and stop as soon as the objectives of the map are met(usually bringing cheese back to the starting point). The score from each level should be saved until the player returns to main menu or clears the last level, where it will be added up.
Erwin Tung	Complete AI (pathfinding) for the enemies (cats), the cats should chase after the player for a certain distance while using the A* pathfinding algorithm. When the player is out of reach, the cat will return to the previous position. The cats also need to respond to non-crouching player while in range vs crouching player.
Carlos Aguirre	Implement limited field of view when lights are off / cheese is retrieved. When lights are off, only a small circular area around the player should be visible. There should be a fog of war for areas that have been discovered on levels with light toggling.

Week 7:

Vincent Doan	Implement the three levels with collision detection. This involves drawing all boundaries using the edge collider and testing that the player/cat cannot traverse past the boundaries.
Leon Matsuki	Implement the mini-map at the top corner for each map. The player/enemies should be dots and the map could be represented in blocks. Another option is to just add a second camera with a further zoom, with obscured view for out of range sections of the map.
Jimmy Wu	Add scoring system utilizing the timer system implemented last week. Add up the total score from each map and save into a file, and is called when the user enters the high score screen.

Erwin Tung	Gather input from other members/playtest the AI and tweak according to difficulty. Adjust the range of mouse, patrolling patterns, speed, chasing distance.
Carlos Aguirre	Implement the speed up/down, lower timer power-ups. The speed up/down should only affect the player for a certain time or while the player is in a certain area. The lower timer power-up should utilize Jimmy's timer system.

Week 8:

Vincent Doan	Implement the map objects such as pipes, doors, keys, cheese, conveyor belts of the new levels. When the player interacts with those objects (or enters the trigger box), the objects should react differently.
Leon Matsuki	Implement the player/cat positions for the mini-map. The player and the cats will be dots or squares and move depending on actual position.
Jimmy Wu	Implement the main character dropping cardboard distraction. The object should interact with the cat and also be there for a certain duration and disappear afterwards.
Erwin Tung	Work on art assets for map objects (door/keys/cheese/etc). The objects should be same or smaller to the size of the player, and be a 2D sprite.
Carlos Aguirre	Implement stun/sleep/slow traps power-ups. Each trap should disappear when in contact with the player only, and modify the player's attributes accordingly.

Week 9:

Vincent Doan	Record/Add in SFX/Voices (Mice SFX). The pitch/length of the
---------------------	--

	recorded voices are to modified and added into the Audio Source of the player/cats.
Leon Matsuki	Compose/Add in Music (Level/Victory/Defeat Songs). Depending on how long this takes, some free music asset might be used instead.
Jimmy Wu	Research whether networking can be applied to our game before the deadline and collaborate with the team. If possible, revise the member schedule
Erwin Tung	Playtest the maps and tweak the AI. Depending on difficulty, the cat should either have varying speed or distance to chase the player.
Carlos Aguirre	Implement sneak/stealth system to work with cats. Playtest and check that crouching enables the player to get by sleeping cats and the AI is behaving properly based on difficulty.

Week 10:

Vincent Doan	Playtest the maps and tweak gameplay for user experience. This is for the final polishing and tweaking phase, such as parameters, volume, visible range, etc, fun.
Leon Matsuki	Implement sound manager system to handle sounds/songs for all levels. Research and implement a system that allows easier handling of sounds/music playing/stopping/pausing.
Jimmy Wu	Update the design document to reflect the current game. Actual goals met, chase in game design due to limitations, additional mechanics that were implemented to get the game working, etc.
Erwin Tung	Configure the default controls to be comfortable to play with. Get feedback from team and update the game document accordingly.
Carlos Aguirre	Implement Level progression system and keep track of scores. Utilize Jimmy's timer system to accumulate score and make sure

	the correct level is loaded after each level completion.
--	--