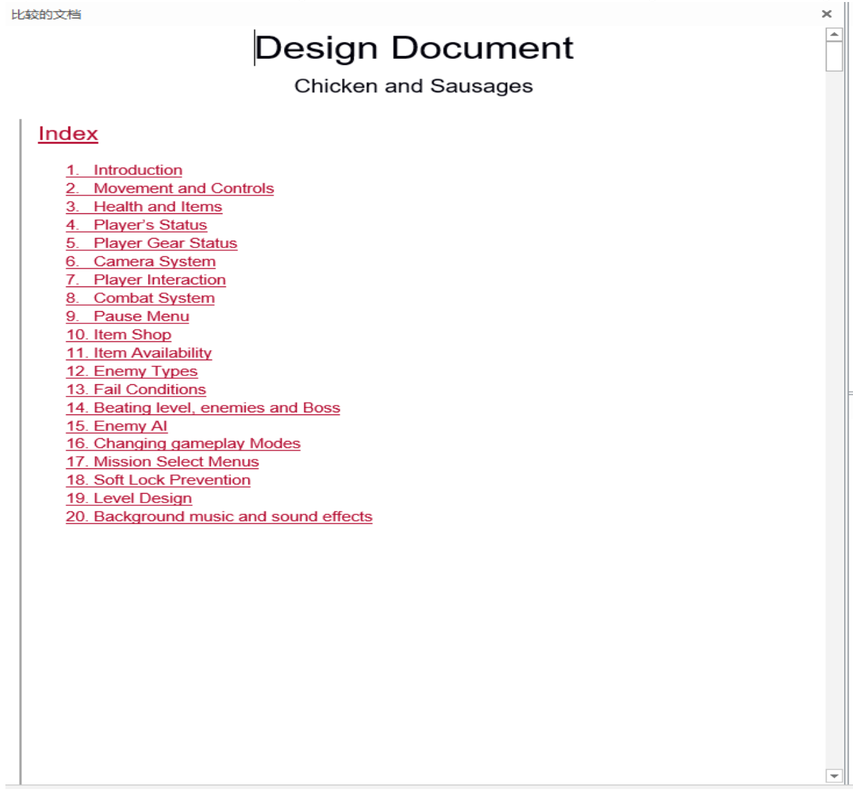
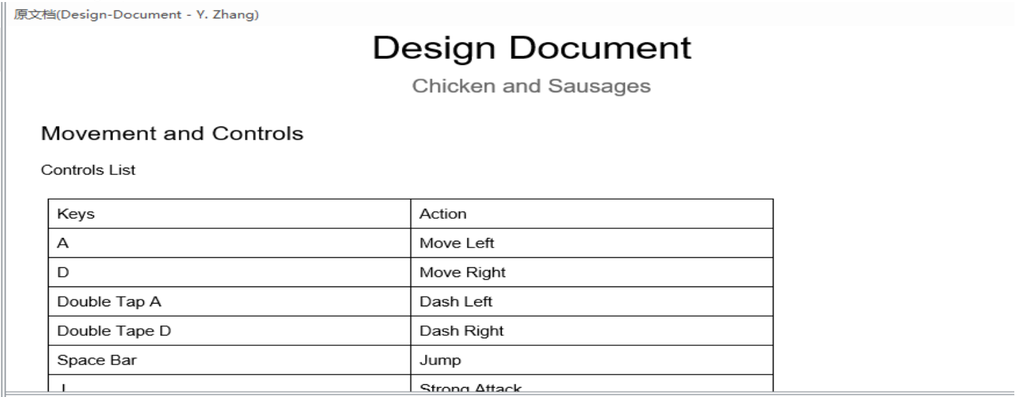
Road Map – Design Document

This is the road map for the design document. Git Diff would not be able to give the changes for words, therefore I am showing a compare version of the word in the following document. There is 233 changes made. 134 of them is insertion and the other 99 are deletion. Below are screenshots with summaries of what's being changed and their explanation. On the left is the compared version, where changes made are all shown in red. On the right top is the old design document, and right bottom is the revised version. Please see Road Map Design Document.doc for word compare vision. Sorry that the text is really small, please zoom in to read.

1. Added an index, format and rephrase the entire document to make it readable as stand-alone. Fixed spelling and grammar errors.
   Explanation: Easier to read as a design document for audiences. Discussed design decisions according to each category.





1. Introduction

This document will outline and explain the design choices and mechanics of the game. Mechanics include player movement, player status, combat system, menus, enemy types and AI, gameplay modes, and items, and camera system.

~~1.~~2. Movement and Controls

Controls List

Keys	Action
A	Move Left
D	Move Right
Double Tap A	Dash Left
Double Tape D(hold)	Dash Right
Space Bar	Jump
J	Strong Attack
I	Attack 1
I -> I	Attack 2
I -> I -> I	Attack 3 Launcher
O	Shoot Bullet
P	Use repair kit

The player will move right by adding a force of 300 in the positive direction, every time he moves his velocity will be set to zero to prevent the character from sliding like he is on ice and to disallow the movement ramping up in speed, this will allow the player to have complete control over their character. A force of 300f because of it a speed that allows the user to have the best control over his character, it is not too fast or too slow. The player moves at maximum speed at the start because it allows our movements to be precise and sharp. The game will have many dangerous objects and hazards which will require precise movements, having movement ramp up in speed will only make the game more difficult and frustrating for the player. There is no lead time for the same reasons above

Movement and Controls

Controls List

Keys	Action
A	Move Left
D	Move Right
Double Tap A	Dash Left
Double Tape D	Dash Right
Space Bar	Jump
J	Strong Attack
I	Attack 1
I -> I	Attack 2

修订的文档(Design-DocumentRev1 - 作者)

1. Introduction

This document will outline and explain the design choices and mechanics of the game. Mechanics include player movement, player status, combat system, menus, enemy types and AI, gameplay modes, and items, and camera system.

2. Movement and Controls

Controls List

Keys	Action
A	Move Left
D	Move Right
Double Tap A	Dash Left
Double Tape D(hold)	Dash Right

Implementation of moving right (refer to playerController)

```
64    public void playerMoveRight(){
65        if (!attacking) {
66            anim.SetTrigger ("Walk");
67
68        }
69        if (dashing) {
70            rb2d.velocity = (new Vector2 (0, rb2d.velocity.y));
71            rb2d.AddForce (new Vector2 (moveForce*2, 0));
72            return;
73        }
74        rb2d.velocity = (new Vector2 (0, rb2d.velocity.y));
75        rb2d.AddForce (new Vector2 (moveForce, 0));
76
77    }
```

```
64    public void playerMoveRight(){
65        if (!attacking) {
66            anim.SetTrigger ("Walk");
67
68        }
69        if (dashing) {
70            rb2d.velocity = (new Vector2 (0, rb2d.velocity.y));
71            rb2d.AddForce (new Vector2 (moveForce*2, 0));
72            return;
73        }
74        rb2d.velocity = (new Vector2 (0, rb2d.velocity.y));
75        rb2d.AddForce (new Vector2 (moveForce, 0));
76
77    }
```

~~Yes~~

The player can move in the air because it gives the player more control over their character, allowing him to adjust himself to land on other platforms, to attack an enemy in the air and dodge potential hazards coming at him. By allowing him to move in mid-air, opens our design space allowing tougher enemies and more complicated platform schemes.

Dashing is an interesting mechanic that was added within the game. It has a lot of positive and negative benefits. Firstly, it does limit our design space forcing us to have larger maps and farther platforms because the increase in the distance and speed a player can potentially traverse the map. It forces us to design levels where the player may be dashing around or walking. The increase in speed for the character would also affect the size of our camera, if the player is moving faster he will have less time to react to enemies being unfairly hit and platforms that require a dash jump may not be seen within. This will be

Implementation of moving right (refer to playerController)

```
64    public void playerMoveRight(){
65        if (!attacking) {
66            anim.SetTrigger ("Walk");
67
68        }
69        if (dashing) {
70            rb2d.velocity = (new Vector2 (0, rb2d.velocity.y));
71            rb2d.AddForce (new Vector2 (moveForce*2, 0));
72            return;
73        }
74        rb2d.velocity = (new Vector2 (0, rb2d.velocity.y));
75        rb2d.AddForce (new Vector2 (moveForce, 0));
76
77    }
```

Yes the player can move in the air because it gives the player more control over their character, allowing him to adjust himself to land on other platforms, to attack an enemy in the air and dodge potential hazards coming at him. By allowing him to move in mid air, opens our design space allowing tougher enemies and more complicated platform schemes.

修订的文档(Design-DocumentRev1 - 作者)

Implementation of moving right (refer to playerController)

```
64    public void playerMoveRight(){
65        if (!attacking) {
66            anim.SetTrigger ("Walk");
67
68        }
69        if (dashing) {
70            rb2d.velocity = (new Vector2 (0, rb2d.velocity.y));
71            rb2d.AddForce (new Vector2 (moveForce*2, 0));
72            return;
73        }
74        rb2d.velocity = (new Vector2 (0, rb2d.velocity.y));
75        rb2d.AddForce (new Vector2 (moveForce, 0));
76
77    }
```

The player can move in the air because it gives the player more control over their character, allowing him to adjust himself to land on other platforms, to attack an enemy in the air and dodge potential hazards coming at him. By allowing him to move in mid-air, opens our design space allowing tougher enemies and more complicated platform schemes.

2. Add boost meter to balance the dashing ability. Explanation: We decide to have this to avoid a defect which raised by another team in the git issue section.
3. Rewrite descriptions for items. Explanation: Rewrite the description to avoid user confusion. The original design document lack of description for what passive and useable items.

solved through a ~~minimap~~mini-map within the UI. We also are aware that with dashing player may only dodge enemies and get to the end of the level, our level design will have to discourage this behaviour. However, the addition of the game mechanic dashing is worth it because it allows the player to be fast~~,~~ as if they were piloting an incredible war machine. It allows the player more options and to be proactive in dodging attacks, it increases the skill level of the game.

A boost meter is introduced to balance the dashing ability. The player has the ability to dash across entire levels in very short time if the dash ability does not have a limiting mechanic. Dashing will decrease the boost meter over time, once the meter runs out the player must wait until the meter is full again to perform another dash. The boost meter will recover by itself similar to the stamina meter in Dark Souls games.

Dashing will be implemented through double tapping and then hold the d or a and will move 2 times as fast as walking. Refer to the "playerController" script for implementation.

~~No~~ We are decide not ~~using~~to use Unity's standard assets for movement controls because it is tedious to create a new ~~tags~~tag for each key, although the benefits would allow us to have controls configurable keys from the unity build menu, but it would also have less control over the implementation of our keys for our movement. ~~No the control's~~The controls are not going to be configurable, the keys for the game have been designed to be played in that specific way, the game experience will be the most fun using that exact key layout. Key swaps will provide us with more work with little benefit and make the game menus more complicated. For instance, if the user randomly clicks through the menus and swaps keys, ~~it's~~it is possible that they weren't aware of what keys they inputted or forget~~,~~ which ~~can~~ potentially can soft-lock the game. This would also force us to design how key swaps work, it would have to be simple, quick and intuitive and how we would save their new configurable set-up. We also want every player to be using the same keys, to prevent strategic placements of keys to provide optimal movements that give players advantages for doing so.

~~2.~~3. Health and Items

~~The~~There are two types of item: useable and passive. HP repair kit as the usable item, damage upgrade and health upgrade as the passive item. We decide to make attributes for items ~~can stack. It is to reward~~to be stackable, which would encourage the player who ~~take~~takes the time explore levels~~, their items attributes can stack. There are one useable items: repair kit. Only One repair kit can be used at a time. No more can be used when hp is full. There will not be simultaneous use case with this type of item.~~

~~Currently a player can use hp recovery kit as useable item, we have healing item and collectable items, and hp recovery kit belong to healing item, iron/bronze/silver/gold/platinum belong to passive item.~~

~~HP repair kit: recover HP to full.~~

dashing player may only dodge enemies and get to the end of the level, our level design will have to discourage this behaviour. However the addition of the game mechanic dashing is worth it because it allows the player to be fast, as if they were piloting an incredible war machine. It allows the player more options and to be proactive in dodging attacks, it increases the skill level of the game.

Dashing will be implemented through double tapping the d or a and will move 2 times as fast as walking. Refer to the playerController script for implementation

No we are not using Unity's standard assets for movement controls because it is tedious to create a new tags for each key, although the benefits would allow us to have controls configurable keys from the unity build menu but it would also have less control over the implementation of our keys for our movement. No the control's are not going to be configurable, the keys for the game have been designed to be played in that specific way, the game experience will be the most fun using that exact key layout. Key swaps will provide us with more work with little benefit and make the game menus more complicated. For instance if the user randomly clicks through the menus and swaps keys, it's possible they weren't aware of what keys they inputted or forget, which can potentially can soft-lock the game. This would also force us to design how key swaps work, It would have to be simple, quick and intuitive and how we would save their new configurable set-up. We also want every player to be using the same keys, to prevent strategic placements of keys to provide optimal movements that give players advantages for doing so.

修订的文档(Design-DocumentRev1 - 作者)

would also affect the size of our camera, if the player is moving faster he will have less time to react to enemies being unfairly hit and platforms that require a dash jump may not be seen within. This will be solved through a mini-map within the UI. We also are aware that with dashing player may only dodge enemies and get to the end of the level, our level design will have to discourage this behaviour. However, the addition of the game mechanic dashing is worth it because it allows the player to be fast as if they were piloting an incredible war machine. It allows the player more options and to be proactive in dodging attacks, it increases the skill level of the game.

A boost meter is introduced to balance the dashing ability. The player has the ability to dash across entire levels in very short time if the dash ability does not have a limiting mechanic. Dashing will decrease the boost meter over time, once the meter runs out the player must wait until the meter is full again to perform another dash. The boost meter will recover by itself similar to the stamina meter in Dark Souls games.

Dashing will be implemented through double tapping and then hold the d or a and will move 2 times as fast as walking. Refer to the "playerController" script for implementation.

We are decide not to use Unity's standard assets for movement controls because it is tedious to create a new tag for each key, although the benefits would allow us to have controls configurable keys from the unity build menu, but it would also have less control over the implementation of our keys for our movement. The controls are not going to be configurable, the keys for the game have been designed to be played in that specific way, the game experience will be the most fun using that exact key layout. Key swaps will provide us with more work with little benefit and make the game menus more complicated. For instance, if the user randomly clicks through the menus and

4. Changed items only to HP repair kit, damage upgrade, and health upgrade only. Explanation: This is a simplified version of the old items system, it's still the same idea. We decided to build the basic of the item system first and polish it in the future development if possible.

~~Damage function = 30~~

~~Steel armor: +30 max HP (values may change on implementation)~~

~~Titanium armor: +60 max HP~~

~~Phase shift armor: +90 max HP~~

~~Energy field armor: +120 max HP~~

~~Proto X armor: +150 max HP~~

~~Armor Piercing Rounds: +3 attack (values may change on implementation)~~

~~Hi-Kinetic Delivery System: +6 attack~~

~~Magnetic Coating: +9 attack~~

~~Newtype Psycho-communication: +12 attack~~

~~Beam Weaponry: +15 attack~~

~~Damage = attack+baseDamageOfMove~~

All item types are picked up by walking over the object with the player's avatar. The stat upgrades will take effect immediately, while the repair kit will be consumed when the player uses the item. The key to use the repair kit is "P"

HP repair kit: recover health to full

Damage upgrade: increase 5 unit to the current damage

Health upgrade: increase 5 unit to current maximum health

Only One repair kit can be used at a time and would not allow the player to use HP repair kit when the character's health is full.

The following screenshot is the usable item: HP repair kit

Damage function = 30

Steel armor: +30 max HP (values may change on implementation)

Titanium armor: +60 max HP

Phase shift armor: +90 max HP

Energy field armor: +120 max HP

Proto X armor: +150 max HP

Armor Piercing Rounds: +3 attack (values may change on implementation)

Hi-Kinetic Delivery System: +6 attack

Magnetic Coating: +9 attack

Newtype Psycho-communication: +12 attack

Beam Weaponry: +15 attack

修订的文档(Design-DocumentRev1 - 作者)

decide to make attributes for items to be stackable, which would encourage the player who takes the time explore levels. All item types are picked up by walking over the object with the player's avatar. The stat upgrades will take effect immediately, while the repair kit will be consumed when the player uses the item. The key to use the repair kit is "P"

HP repair kit: recover health to full

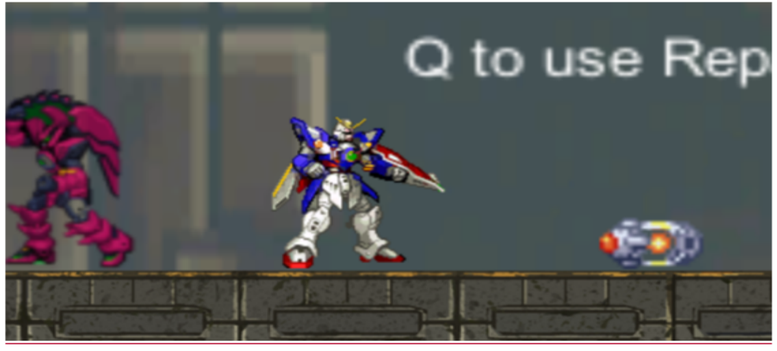
Damage upgrade: increase 5 unit to the current damage

Health upgrade: increase 5 unit to current maximum health

Only One repair kit can be used at a time and would not allow the player to use HP repair kit when the character's health is full.

The following screenshot is the usable item: HP repair kit

5. Screenshot added: scrap, player's status and shop. Explanation: Straightforward screenshots showing our assets being used in the game development.



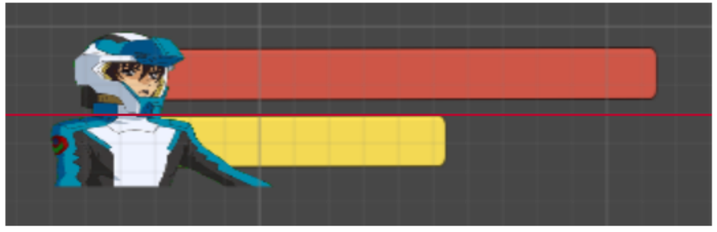
3.4. Player's Status

There are:

: HP, attack, defense, movement speed, jump force, Energy, Scraps (in game currency), #of) and repair kits

HP and Energy will be shown, there is a health bar located on top left corner, and there is a Energy bar, Energy bar is below HP meter. Scrap number below the energy bar and repair kits will be to the right of scrap number

HP and boost will always be visible by being placed at top left corner.



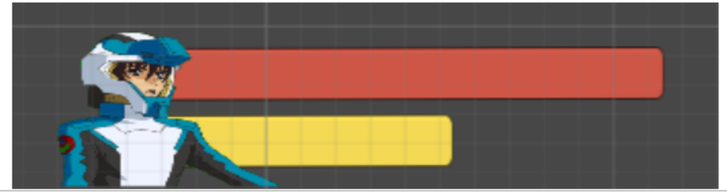
Player's Status

There are :

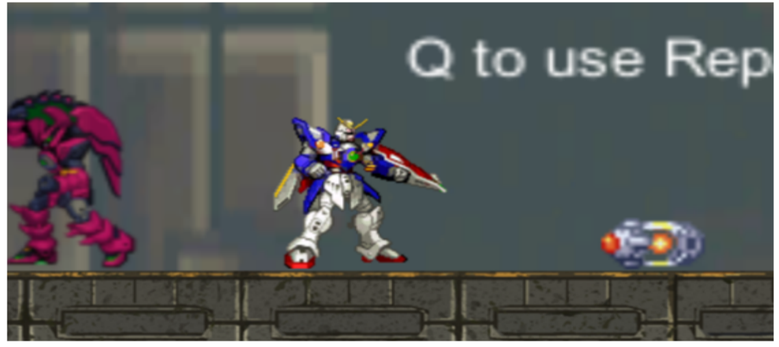
HP, attack, defense, movement speed, jump force, Energy, Scraps (in game currency), #of repair kits

HP and Energy will be shown, there is a health bar located on top left corner, and there is a Energy bar, Energy bar is below hp meter. Scrap number below the energy bar and repair kits will be to the right of scrap number

HP and boost will always be visible by being placed at top left corner.



修订的文档(Design-DocumentRev1 - 作者)

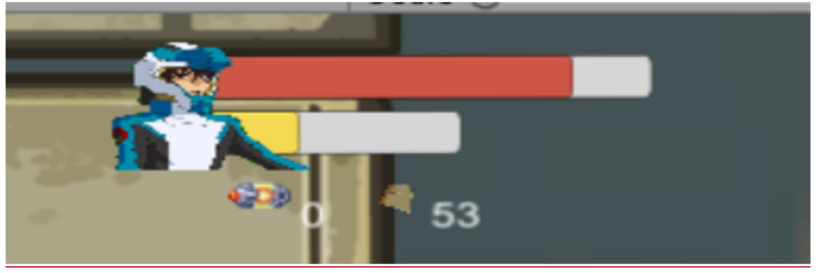


4. Player's Status

There are: HP, attack, defense, movement speed, jump force, Energy, Scraps (in game currency) and repair kits

~~Player's Status without scrap or repair kits~~

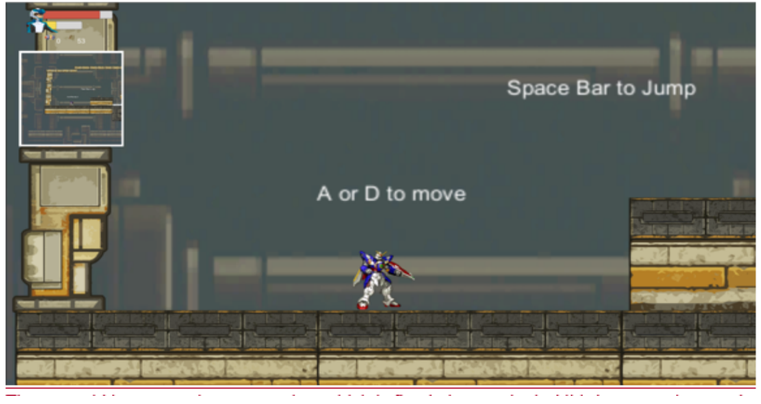
Here is a screenshot of player's status showing number of scraps and repair kits along the HP and Energy bar.



4.5. Player gear status

~~Due to the lack of different gear assets, animations, sprites, and special attack effects, there is no plan for equipable gears for the game.~~ To implement different weapons and equipment, it would require custom made sprites to accommodate the different attack animations and varied look to the player's avatar. Due to the lack of different gear assets, animations, sprites, and special attack effects. We decide not to have equipable gears for the game at the current stage of our game, but instead focusing more on to the level design and combat system.

5.6. Camera System



Player's Status without scrap or repair kits

Player gear status

Due to the lack of different gear assets, animations, sprites, and special attack effects, there is no plan for equipable gears for the game. To implement different weapons and equipment would require custom made sprites to accommodate the different attack animations and varied look to the player's avatar.

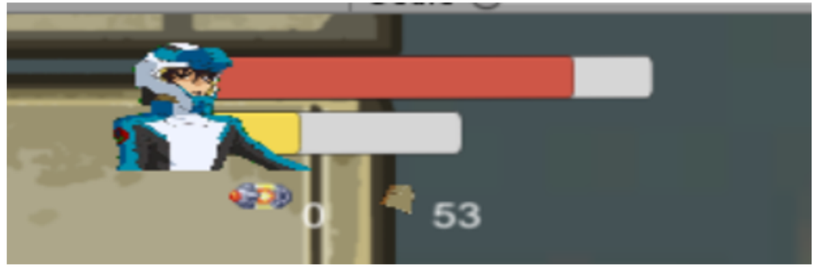
Camera System

The camera system follows the player at a given offset. This is implemented by constantly updating the camera's position by getting the transform position of the player's character and adding it to it by an offset of -10, This offset is added in so that the player and background objects are not on the same plane preventing them from being seen. The camera is orthogonal because it is a 2d game, there is no need for an additional sense of perspective. All our graphics are flat and by making it perspective it would allow users to tell that the objects are paper like in perspective. It would also make things more confusing for us to work with as objects maybe correctly placed in their respective x and y coordinates but in game appear differently due to the z axis. Yes there are plans to have multiple cameras, asides from the main camera that follows the player, there shall be one smaller camera that will be part of the HUD display that act as a minimap. It will be a small square on the screen that shows a more zoomed out view of the playing field. The player controls the main camera by controlling the player, when the player moves the camera will follow, the camera will always be centered on the player. Refer to camera Script to check for full implementation

Player Interactions

修订的文档(Design-DocumentRev1 - 作者)

Here is a screenshot of player's status showing number of scraps and repair kits along the HP and Energy bar.



5. Player gear status

To implement different weapons and equipment, it would require custom made sprites to accommodate the different attack animations and varied look to the player's avatar. Due to the lack of different gear assets, animations, sprites, and special attack effects. We decide not to have equipable gears for the game at the current stage of our game, but instead focusing more on to the level design and combat system.

6. Add offset direction to avoid confusion for the audience. Explanation: adjust unclear meaning from TA's individual comments.

There would be one main camera view which is fixed player-oriented third person view, and one mini map view on the top-left corner under the status. The camera system follows the player at a given offset. This is implemented by constantly updating the camera's position by getting the transform position of the player's character and adding it to it by an offset of -10, in the x-axis. This offset is added in so that the player and background objects are not on the same plane preventing them from being seen. We decide to have the camera ~~is~~view orthogonal ~~because~~since it is a 2D game;, there is no need for an additional sense of perspective. All our graphics are flat and by making it perspective it would allow users to tell that the objects are paper like in perspective. It would also make things more confusing for us to work with as objects maybe correctly placed in their respective x and y coordinates but in the game appear differently due to the z-axis. ~~Yes there are plans to have multiple~~There is a mini map cameras, asides from the main camera that follows the player~~, there shall be one~~. We decide the mini map camera to have a smaller camera ~~that will be part of the HUD display that act as a minimap. It will be a small square on the screen that~~ shows a more zoomed out view of the playing field. The player controls the main camera by controlling the player, when the player moves the camera will follow, the camera will always be centered on the player. Refer to camera Script to check for full implementation.

~~6.~~7. Player Interactions

This is a classic single-player game, so there are no other players in this game;. Therefore no such interactions with other player exist. The player can interact with enemies by performing and receiving attacks. This interaction will result in the loss of ~~hp~~health depending on who receives the attack. When the player performs an attack, a collision box will be created for a short amount of time. The size and distance of the collision box will be the same as how far and wide the attack sprite appears. If this box collides with the enemy collider, the enemy will receive damage~~. Vise~~ and vice versa for player receiving damage from the enemy.

~~So far~~ NPCs exist within cutscene and within Shop objects. If player's collider collides with the shop's collider and presses the basic attack button, this will introduce the shop interaction. Items available at the shop will appear in mid-air, and the player can purchase the item by jumping onto the item with sufficient currency. If the player jumps to the item without enough scrap, the item will remain in the air.

The camera system follows the player at a given offset. This is implemented by constantly updating the camera's position by getting the transform position of the player's character and adding it to it by an offset of -10, This offset is added in so that the player and background objects are not on the same plane preventing them from being seen. The camera is orthogonal because it is a 2d game, there is no need for an additional sense of perspective. All our graphics are flat and by making it perspective it would allow users to tell that the objects are paper like in perspective. It would also make things more confusing for us to work with as objects maybe correctly placed in their respective x and y coordinates but in game appear differently due to the z axis. Yes there are plans to have multiple cameras, asides from the main camera that follows the player, there shall be one smaller camera that will be part of the HUD display that act as a minimap. It will be a small square on the screen that shows a more zoomed out view of the playing field. The player controls the main camera by controlling the player, when the player moves the camera will follow, the camera will always be centered on the player. Refer to camera Script to check for full implementation

Player Interactions

There are no other players in this game, therefore no such interactions exist.

The player can interact with enemies by performing and receiving attacks. This interaction will result in the loss of hp depending on who receives the attack. When the player performs an attack, a collision box will be created for a short amount of time. The size and distance of the collision box will be the same as how far and wide the attack sprite appears. If this box collides with the enemy collider, the enemy will receive damage. Vise versa for player receiving damage from the enemy.

So far NPCs exist within cutscene and within Shop objects. If player's collider collides with the shop's collider and presses basic attack button, this will introduce the shop interaction. Items available at the shop will appear in mid air, and the player can purchase the item by jumping onto the item with sufficient currency. If player jumps to the item without enough scrap, the item will remain in air.

Player interacts with items by walking over the objects to pick up the items. Player can consume repair kits by pressing the designated button to decrease the

修订的文档(Design-DocumentRev1 - 作者)

There would be one main camera view which is fixed player-oriented third person view, and one mini map view on the top-left corner under the status. The camera system follows the player at a given offset. This is implemented by constantly updating the camera's position by getting the transform position of the player's character and adding it to it by an offset of -10 in the x-axis. This offset is added in so that the player and background objects are not on the same plane preventing them from being seen. We decide to have the camera view orthogonal since it is a 2D game; there is no need for an additional sense of perspective. All our graphics are flat and by making it perspective it would allow users to tell that the objects are paper like in perspective. It would also make things more confusing for us to work with as objects maybe correctly placed in their respective x and y coordinates but in the game appear differently due to the z-axis. There is a mini map cameras, asides from the main camera that follows the player. We decide the mini map camera to have a smaller camera shows a more zoomed out view of the playing field. The player controls the main camera by controlling the player, when the player moves the camera will follow, the camera will always be centered on the player. Refer to camera Script to check for full implementation.

7. Player Interactions

This is a classic single-player game, so there are no other players in this game. Therefore no such interactions with other player exist. The player can interact with enemies by performing and receiving attacks. This interaction will result in the loss of health depending on who receives the attack. When the player performs an attack, a collision box will be created for a short amount of time. The size and distance of the collision box will be the same as how far and wide the attack sprite appears. If this box collides with the enemy collider, the enemy will receive damage and vice versa for player receiving damage from the enemy.

NPCs exist within cutscene and within Shop objects. If player's collider collides with the shop's collider and presses the basic attack button, this will introduce the shop interaction. Items available at the shop will appear in mid-air, and the player can purchase the item by jumping onto the item with sufficient currency. If the player jumps to the item without enough scrap, the item will remain in the air.

7. Explain how damage is calculated in the code. Added a formula below. Explanation: adjust unclear meaning from TA's individual comments.

Player

The player can interacts with items by walking over the objects to pick up the items. Player can consume repair kits by pressing the designated button to decrease the counter of repair kits carried by the player.

7.8. Combat System

Players and enemies both have a pool of health and energy. Combat in this game works by players using attacks to remove the HP of the enemy. Every attack that hits an enemy puts them in hitstunhit-stun depending on the type of attack. For example, if the player is in hit stunhitstun for over 2two seconds then the player will be granted invulnerability for a few seconds. This is done by removing having a Boolean called invulnerability that triggers if hitstunhit-stun is greater than 2 seconds,two second. Once this triggers no attacks will work on the player and once invulnerability time is over it will be set to false. This is to prevent the player from being stuck in hitstunhit-stun forever, enemies will not be granted with invulnerability (refer to the enemy types for their attacks) The player performs an attack through buttons presses, refer to the table for the control list. A combo is when an attack that hits an enemy before they are out of hitstun. The hitboxes should attempt to cover the model of the attack.

They are generally box-shaped because they fit the shape of the model better than circle collider. Box collider will also allow models to stand on the ground easier. The only enemy that his afits the circle collider is the rolling dude. A player avoids damage by avoiding enemy hitboxes. Specials attacks are moves that require energy. At the moment shooting and dash attack are the only implemented special attack. Shooting in the game is meant to do low damage but stun the opponent allowing the player to move in and go for a combo attack. Shooting will cost energy limiting the amount a player can shoot and discouragingdiscourage a play style of playing safe and far and only shooting. When the player hits the enemy with a normal attack((Strong or Combo) he will recover an amount of energy back. This is to encourage the player to use both ranged and melee combat asides only ranged. A hit registers in the middle of the attack animation after the windup. Damage is calculated by a function: enemy health -= damageOfMove+playerAttack.

Damage is calculated by a function: enemy health = enemy health – damageOfMove + playerAttack. Note: enemy health is initially set to 100.

Energy Cost Table (numbers may change on play test)

Player interacts with items by walking over the objects to pick up the items. Player can consume repair kits by pressing the designated button to decrease the counter of repair kits carried by the player.

Combat System

Players and enemies both have a pool of health and energy. Combat in this game works by players using attacks to remove the hp of the enemy. Every attack that hits an enemy puts them in hitstun depending on the type of attack. If the player is in hit stun for over 2 seconds then the player will be granted invulnerability for a few seconds, this is done by removing having a boolean called invulnerability that triggers if hitstun is greater than 2 seconds, once this triggers no attacks will work on the player and once invulnerability time is over it will be set to false. This is to prevent the player from being stuck in hitstun forever, enemies will not be granted with invulnerability(refer to the enemy types for their attacks) The player performs an attack through buttons presses, refer to the table for the control list. A combo is when an attack that hits an enemy before they are out of hitstun. The hitboxes should attempt to cover the model of the attack. They are generally box shaped because they fit the shape of the model better than circle collider, box collider will also allow models to stand on the ground easier, the only enemy that his a circle collider is the rolling dude . A player avoids damage by avoiding enemy hitboxes. Specials attacks are moves that require energy. At the moment shooting and dash attack are the only implemented special attack. Shooting in the game is meant to do low damage but stun the opponent allowing the player to move in and go for a combo attack. Shooting will cost energy limiting the amount a player can shoot and discouraging a play style of playing safe and far and only shooting. When the player hits the enemy with a normal attack(Strong or Combo) he will recover an amount of energy back. This is to encourage the player to use both ranged and melee combat asides only ranged. A hit registers in the middle of the attack animation after the windup. Damage is calculated by a function: enemy health -= damageOfMove+playerAttack.

Energy Cost Table (numbers may change on play test)

Moves	Energy Cost

修订的文档(Design-DocumentRev1 - 作者)

The player can interacts with items by walking over the objects to pick up the items. Player can consume repair kits by pressing the designated button to decrease the counter of repair kits carried by the player.

8. Combat System

Players and enemies both have a pool of health and energy. Combat in this game works by players using attacks to remove the HP of the enemy. Every attack that hits an enemy puts them in hit-stun depending on the type of attack. For example, if the player is in hitstun for over two seconds then the player will be granted invulnerability for a few seconds. This is done by removing having a Boolean called invulnerability that triggers if hit-stun is greater than two second. Once this triggers no attacks will work on the player and once invulnerability time is over it will be set to false. This is to prevent the player from being stuck in hit-stun forever, enemies will not be granted with invulnerability (refer to the enemy types for their attacks) The player performs an attack through buttons presses,

Energy Cost Table (numbers may change on play test)

Moves	Energy Cost
Shooting	10
Dash	1 per frame

Damage Table (numbers may change on play test)

Move	Damage
Strong Attack	15
Attack1	10
Attack2	10
Attack 3	10
Bullet	5
Jump Attack	10

Move	Hitstun (seconds)
Strong Attack	2
Attack1	2
Attack2	2

Energy Cost Table (numbers may change on play test)

Moves	Energy Cost
Shooting	10
Dash	1 per frame

Damage Table (numbers may change on play test)

Move	Damage
Strong Attack	15
Attack1	10
Attack2	10
Attack 3	10

修订的文档(Design-DocumentRev1 - 作者)

Energy Cost Table (numbers may change on play test)

Moves	Energy Cost
Shooting	10
Dash	1 per frame

Damage Table (numbers may change on play test)

Move	Damage
Strong Attack	15

~~8.~~ 9. Pause Menu

Pausing the game will completely halt all game elements and display the pause menu, from which they can choose from a variety of options. This includes, but is not limited to, changing the settings of the game, restarting the current level, returning in the main menu and unpausing the game. If the player was mid-jump and pressed the pause button; the force applied to the player's character will be preserved but the player's character will freeze. As soon as they ~~unpause~~un-pause the game the player's character will finish the remaining jump as if the pause button was never pressed. The same applies to all animations, projectiles and movements in the game.

Main Menu (WIP)

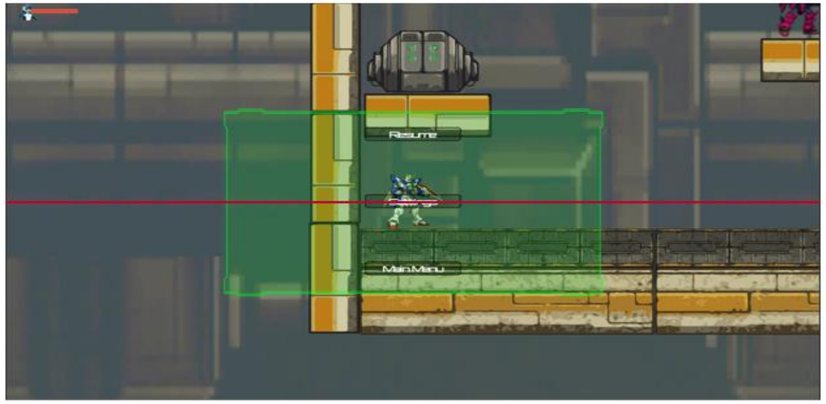
Pause Menu

Pausing the game will completely halt all game elements and display the pause menu, from which they can choose from a variety of options. This includes, but is not limited to, changing the settings of the game, restarting the current level, returning in the main menu and unpausing the game. If the player was mid jump and pressed the pause button; the force applied to the player's character will be preserved but the player's character will freeze. As soon as they unpause the game the player's character will finish the remaining jump as if the pause button was never pressed. The same applies to all animations, projectiles and movements in the game.

修订的文档(Design-DocumentRev1 - 作者)

9. Pause Menu

Pausing the game will completely halt all game elements and display the pause menu, from which they can choose from a variety of options. This includes, but is not limited to, changing the settings of the game, restarting the current level, returning in the main menu and unpausing the game. If the player was mid-jump and pressed the pause button; the force applied to the player's character will be preserved but the player's character will freeze. As soon as they un-pause the game the player's character will finish the remaining jump as if the pause button was never pressed. The same applies to all animations, projectiles and movements in the game.



Pause Menu (WIP)

The game will completely freeze, halting all movement. This includes animations and projectiles.

Pause Menu (WIP)

The game will completely freeze, halting all movement. This includes animations and projectiles.

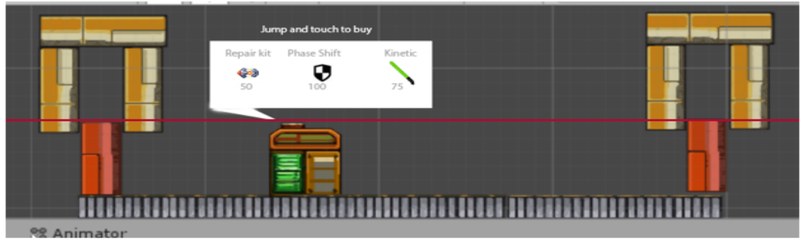
修订的文档(Design-DocumentRev1 - 作者)

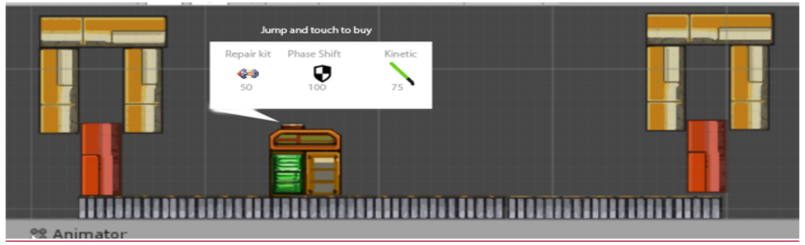
Pause Menu (WIP)

8. Adjust the number of scraps (cost) for items. Explanation: Since we decide to only implement a simple version of our item system, therefore the prices for items are redesigned accordingly.

~~9.~~10.Item Shop

Every level will have a boss ~~walk way~~walkway before the end of the level, regardless if there is a boss. The item menu will be accessible at this walkway. The player will attack the store will open a floating item menu move the shop icon. The player can purchase items by jumping into the item and having enough scrap. Once an item is purchased from the shop, it will disappear from the shop and no longer be purchasable. The item's cost will be ~~depend~~depending on how "strong" the item is. Item stats will stack.



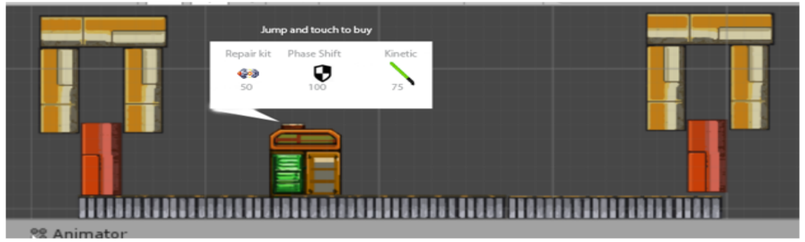


Shop in Boss Walkway

Item	Cost
Repair kit	50 scrap
~~Steel armor~~health upgrade level 1	50 scrap
~~Titanium armor~~damage upgrade level 1	~~75~~50 scrap

Item Shop

Every level will have a boss walk way before the end of the level, regardless if there is a boss. The item menu will be accessible at this walkway. The player will attack the store will open a floating item menu move the shop icon. The player can purchase items by jumping into the item and having enough scrap. Once an item is purchased from the shop, it will disappear from the shop and no longer be purchasable. The item's cost will be depend on how "strong" the item is. Item stats will stack.

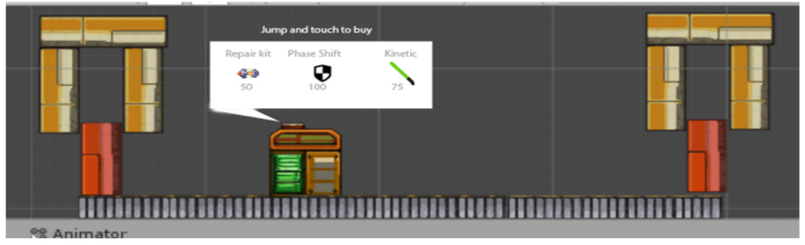


Shop in Boss Walkway

修订的文档(Design-DocumentRev1 - 作者)

10. Item Shop

Every level will have a boss walkway before the end of the level, regardless if there is a boss. The item menu will be accessible at this walkway. The player will attack the store will open a floating item menu move the shop icon. The player can purchase items by jumping into the item and having enough scrap. Once an item is purchased from the shop, it will disappear from the shop and no longer be purchasable. The item's cost will be depending on how "strong" the item is. Item stats will stack.



Shop in Boss Walkway

Item	Cost
Repair kit	50 scrap
~~Steel armor~~health upgrade level 1	50 scrap
~~Titanium armor~~damage upgrade level 1	~~75~~50 scrap
~~Phase Shift armor~~health upgrade level 2	~~100~~75 scrap
~~Energy Field armor~~damage upgrade level 2	~~125~~75 scrap
~~Proto X Armor~~health upgrade level 3	~~150~~100 scrap
~~Armor Piercing Rounds~~damage upgrade level 3	~~50~~100 scrap
~~Hi Kinetic Delivery System~~health upgrade level 4	~~75~~125 scrap
~~Magnetic Coating~~damage upgrade level 4	~~100~~125 scrap
~~Newtype Psycho communication~~health upgrade level 5	~~125~~150 scrap
~~Beam Weaponry~~damage upgrade level 5	150 scrap

~~10.~~11. Item Availability

There are repair kits, ~~weapons,~~ health upgrade, and ~~armours~~damage upgrade that can be purchased to improve player's stats. The stock will not change over time. Each level will have ~~a~~ different ~~type~~levels of ~~weapon~~health upgrade and ~~armour~~damage upgrade. For example, level 1 will have the ~~Armor Piercing Rounds~~health upgrade level 1 and ~~Steel armor~~damage upgrade level 1 and each successive level will have the next ~~armour and weapon~~level of upgrade items of the next rank. There will always be one repair kit to be purchased before the end of the level.

Item	Cost
Repair kit	50 scrap
Steel armor	50 scrap
Titanium armor	75 scrap
Phase Shift armor	100 scrap
Energy Field armor	125 scrap
Proto X Armor	150 scrap
Armor Piercing Rounds	50 scrap

修订的文档(Design-DocumentRev1 - 作者)

Shop in Boss Walkway

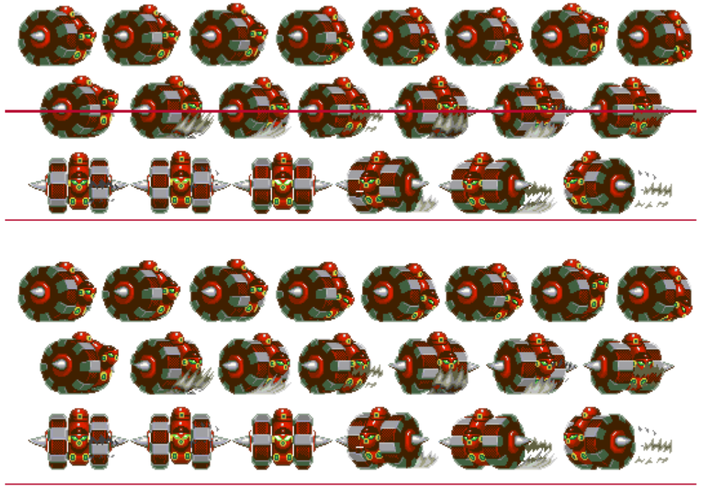
Item	Cost
Repair kit	50 scrap
health upgrade level 1	50 scrap
damage upgrade level 1	50 scrap
health upgrade level 2	75 scrap
damage upgrade level 2	75 scrap
health upgrade level 3	100 scrap
damage upgrade level 3	100 scrap

the next rank. There will always be one repair kit to be purchased before the end of the level.

~~11.~~12. Enemy Types

(Some sprite sheets are cut for space, refer to sprite sheets if you want to complete sheet.)

Rolling Mech - Ground enemy that moves in a single direction does damage to the player on touch



Rolling Mech

Air enemy that flies in a direction does damage to the player on touch

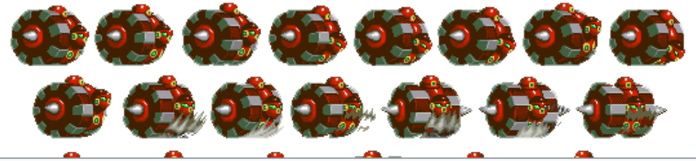
Sky mech

the level.

Enemy Types

(Some sprite sheets are cut for space, refer to sprite sheets if you want to complete sheet.)

Rolling Mech - Ground enemy that moves in a single direction does damage to the player on touch



修订的文档(Design-DocumentRev1 - 作者)

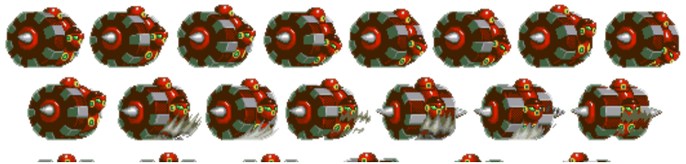
11. Item Availability

There are repair kits, health upgrade, and damage upgrade that can be purchased to improve player's stats. The stock will not change over time. Each level will have different levels of health upgrade and damage upgrade. For example, level 1 will have the health upgrade level 1 and damage upgrade level 1 and each successive level will have the next level of upgrade items of the next rank. There will always be one repair kit to be purchased before the end of the level.

12. Enemy Types

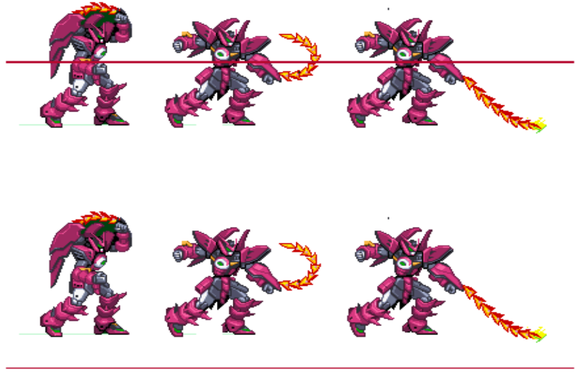
(Some sprite sheets are cut for space, refer to sprite sheets if you want to complete sheet.)

Rolling Mech - Ground enemy that moves in a single direction does damage to the player on touch



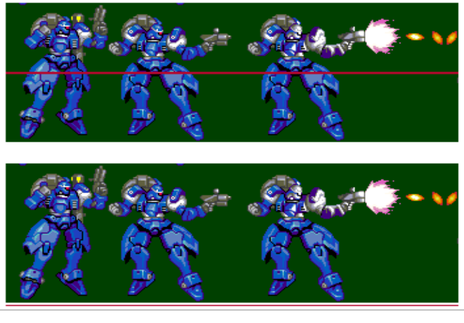
Sky mech

Enemy Gundam (Epeon) that performs a melee attack when in range of the player (refer to enemy controller for implementation)



Epeon

Enemy gundam (Tallgeese) that shoots 3 projectiles if the absolute distance of the x and y coordinate is within 10 on the x and 5 on the y.



Sky mech

Enemy Gundam (Epeon) that performs a melee attack when in range of the player (refer to enemy controller for implementation)

Epeon

Enemy gundam (Tallgeese) that shoots 3 projectiles if the absolute distance of the x and y coordinate is within 10 on the x and 5 on the y.

修订的文档(Design-DocumentRev1 - 作者)

Sky mech

Enemy Gundam (Epeon) that performs a melee attack when in range of the player (refer to enemy controller for implementation)

Epeon

Enemy gundam (Tallgeese) that shoots 3 projectiles if the absolute distance of the x and y coordinate is within 10 on the x and 5 on the y.

Static hazards

Spikes - set health to zero on touch

Out of bounds/Fall Death - set health to zero on touch

Boss one - Mattrex

Boss AI Patterns has a sequence of moves that he randomly loops through.

1.) Fire Ball Attack - Fire dino shoots a fireball projectile at the player, each fireball will do 10 damage.

2.) Jump Attack—; He jumps high into the air and attempts to stomp the player, if the player at any point touches the dino in the air he takes damage. Jump attack does 10 damage.

Static hazards

Spikes - set health to zero on touch

Out of bounds/Fall Death - set health to zero on touch

Boss one - Mattrex

Boss Ai Patterns

Has a sequence of moves that he randomly loops through

1.) Fire Ball Attack - Fire dino shoots a fireball projectile at the player, each fireball will do 10 damage

修订的文档(Design-DocumentRev1 - 作者)

Tall Geese

Static hazards

Spikes - set health to zero on touch

Out of bounds/Fall Death - set health to zero on touch

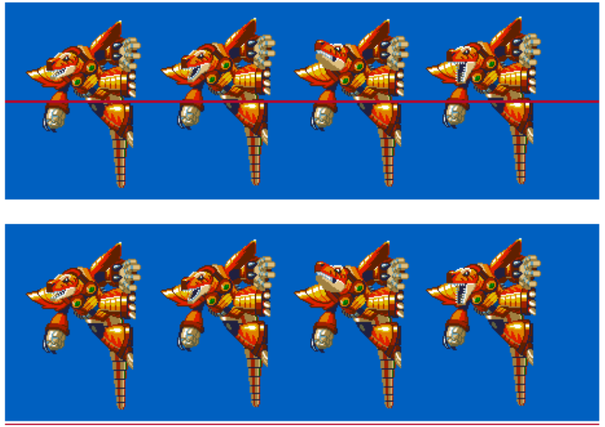
Boss one - Mattrex

Boss AI Patterns has a sequence of moves that he randomly loops through.

1.) Fire Ball Attack - Fire dino shoots a fireball projectile at the player, each fireball will do 10 damage.

3.) Charge Attack- Fire dino charges towards the player location, can be done standing on the ground and while touching the wall. Charge attack does 10 damage

4.) hanging from wall fireball attack - Fire Dino jumps and hangs to the wall and shoots 5-7 fireball projectiles towards the player. He then lands and get tired allowing the player to get in a few hits



When fire dino has his hp < %50 he will perform the hanging wall fireball attack

~~12.~~13. Fail Conditions

The player fails/"dies" in a level by ~~have an~~having a HP value less than zero, In the event that they fall off screen there should be an invisible trigger boundary that surrounds the level and when the player touches it, it will ~~sets~~set their HP level to zero causing a death. For whatever reason, ~~a~~the player manages to get stuck within the level he/she should be able to restart the level in the pause menu. There will also be a move the player can use that self-destructs his/her Gundam causing the player's death but destroying everything that surrounds him/her. It is a more thematic way of restarting the level and adding a new ending within the character's story.

damage

4.) hanging from wall fireball attack - Fire Dino jumps and hangs to the wall and shoots 5-7 fireball projectiles towards the player. He then lands and get tired allowing the player to get in a few hits

When fire dino has his hp < %50 he will perform the hanging wall fireball attack

Fail Conditions

The player fails/"dies" in a level by have an hp value less than zero, In the event that they fall off screen there should be an invisible trigger boundary that surrounds the level and when the player touches it, it will sets their hp level to zero causing a death. For whatever reason, a player manages to get stuck within

修订的文档(Design-DocumentRev1 - 作者)

3.) Charge Attack- Fire dino charges towards the player location, can be done standing on the ground and while touching the wall. Charge attack does 10 damage

4.) hanging from wall fireball attack - Fire Dino jumps and hangs to the wall and shoots 5-7 fireball projectiles towards the player. He then lands and get tired allowing the player to get in a few hits

When fire dino has his hp < %50 he will perform the hanging wall fireball attack

~~12.~~13. Fail Conditions

The player fails/"dies" in a level by ~~have an~~having a HP value less than zero, In the event that they fall off screen there should be an invisible trigger boundary that surrounds the level and when the player touches it, it will ~~sets~~set their HP level to zero causing a death. For whatever reason, ~~a~~the player manages to get stuck within the level he/she should be able to restart the level in the pause menu. There will also be a move the player can use that self-destructs his/her Gundam causing the player's death but destroying everything that surrounds him/her. It is a more thematic way of restarting the level and adding a new ending within the character's story.

~~13.~~14. Beating level, enemies and Boss

The player beats an enemy by getting their HP to be less than zero~~, this~~. It is done through various methods depending on the enemy~~, however~~. However, the most standard way will ~~to~~ have the player's attack hitbox collide with the enemies hurt boxes. Each enemy will have a different hurt box that will encapsulate the enemy. If it is a boss they may only have a hurt box specific to their weak point on their character. Depending on the attack the player used and the level of their ~~weapon, attributes~~, each will ~~due~~do a different amount of damage to the enemy.

~~14.~~15. Enemy AI

For an enemy gundam that attacks it gets the player's position and calculates the distance between them if the distance is less than 10 on the transform, it will move towards the player. This is done by checking if his x coordinate is greater than the players coordinate. Refer to Enemy Controller script from implementation.

~~15.~~16. Changing gameplay Modes

The gameplay modes are selected through the main menu of the game. The two modes are New Game (start story mode) and Free Mode (select particular stages). In Free Mode, another menu will appear to select the desired level to be played. See diagram below for the current iteration of the menu. There will not be a way to change gameplay modes while in a particular mode. The player has the option to pause, exit to the main menu and then select another gameplay mode.

Fail Conditions

The player fails/"dies" in a level by have an hp value less than zero, In the event that they fall off screen there should be an invisible trigger boundary that surrounds the level and when the player touches it, it will sets their hp level to zero causing a death. For whatever reason, a player manages to get stuck within the level he should be able to restart the level in the pause menu. There will also be a move the player can use that self destructs his Gundam causing the player's death but destroying everything that surrounds him. It is a more thematic way of restarting the level and adding a new ending within the character's story.

Beating level, enemies and Boss

The player beats an enemy by getting their hp to be less than zero, this is done through various methods depending on the enemy, however the most standard way will to have the player's attack hitbox collide with the enemies hurt boxes. Each enemy will have a different hurt box that will encapsulate the enemy. If it is a boss they may only have a hurt box specific to their weak point on their character. Depending on the attack the player used and the level of their weapon, each will due a different amount of damage to the enemy.

Enemy Ai

For an enemy gundam that attacks it gets the player's position and calculates the distance between them if the distance is less than 10 on the transform, it will move towards the player. This is done by checking if his x coordinate is greater than the players coordinate. Refer to Enemy Controller script from

修订的文档(Design-DocumentRev1 - 作者)

13. Fail Conditions

The player fails/"dies" in a level by having a HP value less than zero, In the event that they fall off screen there should be an invisible trigger boundary that surrounds the level and when the player touches it, it will set their HP level to zero causing a death. For whatever reason, the player manages to get stuck within the level he/she should be able to restart the level in the pause menu. There will also be a move the player can use that self-destructs his/her Gundam causing the player's death but destroying everything that surrounds him/her. It is a more thematic way of restarting the level and adding a new ending within the character's story.

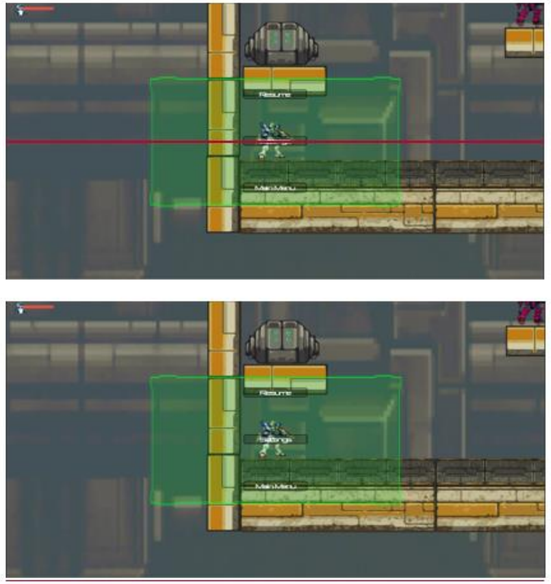
14. Beating level, enemies and Boss

The player beats an enemy by getting their HP to be less than zero. It is done through various methods depending on the enemy. However, the most standard way will have the player's attack hitbox collide with the enemies hurt boxes. Each enemy will have a different hurt box that will encapsulate the enemy. If it is a boss they may only have a hurt box specific to their weak point on their character. Depending on the attack the player used and the level of their attributes, each will do a different amount of damage to the enemy.

15. Enemy AI

For an enemy gundam that attacks it gets the player's position and calculates the distance between them if the distance is less than 10 on the transform, it will move towards the player. This is done by checking if his x coordinate is greater than the players coordinate. Refer to Enemy Controller script from

~~16.~~17. Mission Select Menus



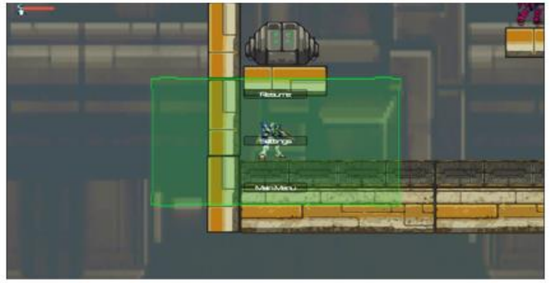
Pause Menu

Screenshots from the above is what the menus are look like. The options available to the players from main menu are new game, free mode, settings and exit game, from pause menu are pause/resume, settings and main menu, from settings player should be able to turn on/off the background music and sounds. There are two menus: main menu and pause menu. Yes, there is a level-select screen. There are five different levels in the free mode that player could select.

~~17.~~18. Soft Lock Prevention

No, the player should not be able to soft-lock the game and be forced to restart but at the moment in our current implementation there are potentially ways to soft-lock the game. Fall boundaries have not been currently implemented, if the player somehow manages to clip through the wall he will fall forever. There is hit stun on the player when they are hit by certain moves, if enough enemies manage to trap a player in a wall and constantly attack the player causing him to be stuck in a infinite loop of hitstun may cause a soft lock. If the game is started without the correct initial conditions level loading may not be correct. Our guarantee to prevent a soft lock from happening within in the game is to ~~allows~~allow the

Mission Select Menus



Pause Menu

Screenshots from the above is what the menus are look like. The options available to the players from main menu are new game, free mode, settings and exit game, from pause menu are pause/resume, settings and main menu, from settings player should be able to turn on/off the background music and sounds. There are two menus:main menu and pause menu. Yes, there is a level-select screen. There are five different levels in the free mode that player could select.

Soft Lock Prevention

修订的文档(Design-DocumentRev1 - 作者)

17. Mission Select Menus

Pause Menu

Screenshots from the above is what the menus are look like. The options available to the players from main menu are new game, free mode, settings and exit game, from pause menu are pause/resume, settings and main menu, from settings player should be able to turn on/off the background music and sounds. There are two menus: main menu and pause menu. Yes, there is a level-select screen. There are five different levels in the free mode that player could select.

certain moves, if enough enemies manage to trap a player in a wall and constantly attack the player causing him to be stuck in a infinite loop of hitstun may cause a soft lock. If the game is started without the correct initial conditions level loading may not be correct. Our guarantee to prevent a soft lock from happening within in the game is to ~~allows~~allow the player to restart a level from a menu and to have access to the main menu. When the player goes to the menu of the game that scene will reinitialize the game by setting all the variables required by the game to be in their correct state.

~~18.~~19. Level Design

There are five levels:
The first level will be the tutorial level where the player gets to know the mechanics of the game. There are messages of how to control the player's avatar at the beginning of the stage. There are also falling spikes that forces the player to use the dash mechanic to evade getting hit. There is one dummy enemy to practice attacks with and one actual enemy to get past. The goal of the player is to reach the end destination of the tutorial. When the goal is reached the next scene will load.

The second level is significantly larger in scale. The level design is fairly straight forward, as this is the first level after the tutorial. There will be around a dozen ~~Epyons~~Epeons in this level. Due to the linearity of this level, combat against enemies is the focus. The level completes when all enemies are eliminated and the destination at the end of the level is reached.

The third level focuses on platforming. There are branching paths with hidden items that boosts a stat. There are places that the player cannot reach by standard jump, and must attack an air canister in the air to be knocked farther. There are also gate puzzles that unlock when the correct sequence of pressure plates are pressed. There will be fewer enemies in this level. Enemies are not required to be eliminated to complete this level.

The fourth level has branching paths, hard to reach places, as well as multiple enemies. Enemies are required to be eliminated this time. This level is intended as a combination of the previous levels.

The final level features a boss battle.

All non-boss enemy types will be featured in all levels, ~~except for the~~and different boss enemy type present in each level. All enemies were decided to be in every level because the current enemy types are very basic. At later levels, the increase in difficulty comes from the number of enemies and the percentage composition of the different enemy types.

In between levels are story cutscenes. These cutscenes are in the style of motion comic. The choice to use motion comic was made for ease of implementation, less time intensive to produce the assets, and sufficient way to provide narrative to the game. The motion comic also enhances the retro style of the game.

on the player when they are hit by certain moves, if enough enemies manage to trap a player in a wall and constantly attack the player causing him to be stuck in a infinite loop of hitstun may cause a soft lock. If the game is started without the correct initial conditions level loading may not be correct . Our guarantee to prevent a soft lock from happening within in the game is to allows the player to restart a level from a menu and to have access to the main menu . When the player goes to the menu of the game that scene will reinitialize the game by setting all the variables required by the game to be in their correct state.

Level Design

There are five levels:
The first level will be the tutorial level where the player gets to know the mechanics of the game. There are messages of how to control the player's avatar at the beginning of the stage. There are also falling spikes that forces the player to use the dash mechanic to evade getting hit. There is one dummy enemy to practice attacks with and one actual enemy to get past. The goal of the player is to reach the end destination of the tutorial. When the goal is reached the next scene will load.

The second level is significantly larger in scale. The level design is fairly straight forward, as this is the first level after the tutorial. There will be around a dozen Epyons in this level. Due to the linearity of this level, combat against enemies is the focus. The level completes when all enemies are eliminated and the destination at the end of the level is reached.

The third level focuses on platforming. There are branching paths with hidden items that boosts a stat. There are places that the player cannot reach by standard jump, and must attack an air canister in the air to be knocked farther. There are also gate puzzles that unlock when the correct sequence of pressure plates are pressed. There will be fewer enemies in this level. Enemies are not required to be eliminated to complete this level.

修订的文档(Design-DocumentRev1 - 作者)

soft-lock the game. Fall boundaries have not been currently implemented, if the player somehow manages to clip through the wall he will fall forever. There is hit stun on the player when they are hit by certain moves, if enough enemies manage to trap a player in a wall and constantly attack the player causing him to be stuck in a infinite loop of hitstun may cause a soft lock. If the game is started without the correct initial conditions level loading may not be correct. Our guarantee to prevent a soft lock from happening within in the game is to allow the player to restart a level from a menu and to have access to the main menu. When the player goes to the menu of the game that scene will reinitialize the game by setting all the variables required by the game to be in their correct state.

19. Level Design

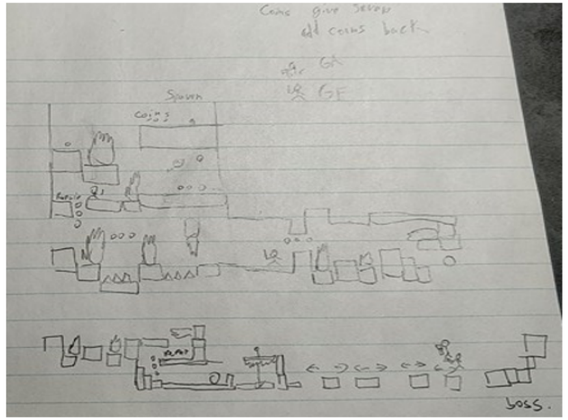
There are five levels:
The first level will be the tutorial level where the player gets to know the mechanics of the game. There are messages of how to control the player's avatar at the beginning of the stage. There are also falling spikes that forces the player to use the dash mechanic to evade getting hit. There is one dummy enemy to practice attacks with and one actual enemy to get past. The goal of the player is to reach the end destination of the tutorial. When the goal is reached the next scene will load.

The second level is significantly larger in scale. The level design is fairly straight forward, as this is the first level after the tutorial. There will be around a dozen Epeons in this level. Due to the linearity of this level, combat against enemies is the focus. The level completes when all enemies are eliminated and the destination at the end of the level is reached.

9. Added new level mock-ups. Explanation: Straightforward graphic illustration for better visibility.

10.Added a new section BGM and sound effect. Including their usage and source. Explanation: We forgot this part last time, now we have it! ☺

Sketch of the new level



20. Background Music and Sound Effect

Sounds and music assets used in this game are a mix from Gundam and Megaman. We chose to use these tracks for our game because Megaman music and sound effects has a futuristic vibe, while Gundam also has a futuristic and militaristic vibe. These sounds suit the feel of our game.

Background Music:
Fighting boss: 06_Opening_Stage_X from Megaman X5
Level 1: 16_Magma_Dragoon_Stage from Megaman X5
Level 2: RX 0 from MS Gundam Unicorn
Main Menu: Mobile Suit from MS Gundam Unicorn

Sound Effect:
We decide to have Mix between Megaman X5 and M obile Suit Gundam Federation vs Zeon sound effects for our game. They are: attack, boss land, can't buy item, click, collect item, dash, enemy death, explosion, fire shot, hit, jump, open door, range attack, repair, use item, take damage and walk. For specific list refer to asset folder in repository.

The second level is significantly larger in scale. The level design is fairly straight forward, as this is the first level after the tutorial. There will be around a dozen Epyons in this level. Due to the linearity of this level, combat against enemies is the focus. The level completes when all enemies are eliminated and the destination at the end of the level is reached.

The third level focuses on platforming. There are branching paths with hidden items that boosts a stat. There are places that the player cannot reach by standard jump, and must attack an air canister in the air to be knocked farther. There are also gate puzzles that unlock when the correct sequence of pressure plates are pressed. There will be fewer enemies in this level. Enemies are not required to be eliminated to complete this level.

The fourth level has branching paths, hard to reach places, as well as multiple enemies. Enemies are required to be eliminated this time. This level is intended as a combination of the previous levels.

The final level features a boss battle.

All enemy types will be featured in all levels, except for the boss level. All enemies were decided to be in every level because the current enemy types are very basic. At later levels, the increase in difficulty comes from the number of enemies and the percentage composition of the different enemy types.

In between levels are story cutscenes. These cutscenes are in the style of motion comic. The choice to use motion comic was made for ease of implementation, less time intensive to produce the assets, and sufficient way to provide narrative to the game. The motion comic also enhances the retro style of the game.

修订的文档(Design-DocumentRev1 - 作者)

implementation, less time intensive to produce the assets, and sufficient way to provide narrative to the game. The motion comic also enhances the retro style of the game.

Sketch of the new level