**“Shield Knight” GDD**

**Game Name:** “Shield Knight”

**Genre:** Casual Action

**Game Elements:**

The player acts as a character capable of jumping and summoning shields to block and attack moving NPCs across a left-to-right 2D plane while having to navigate vertical obstacles and horizontal gaps.

**Elevator Pitch:**

“Shield Knight” is a comedic 2D side-scroller that will have the player invade a castle using good timing and platforming skills to reclaim their nation’s lost holy artifact.. only to find out that the artifact is in fact unholy and their nation are demon worshippers!

**Player:**

Single-player-only.

**Technical Form:**

2D Graphics (Flat)

**View:**

Flat-view pane. Side-view 2D.

**Platform:**

PC

**Language:**

C#

**Device:**

PC

**Key Features**

The main features of the game that will attract its audience are its very colourful and comedic tone and nature; the setting of medieval/fantasy knights and ‘holy’ imagery is played for a very exaggerated effect, with the story that unfolds revealing the unserious humour. Alongside the humour is a serious and challenging platformer that will let the player quickly learn its mechanics based around the use of the in-game controlled character’s shield intuitively with fluid animations that are visually appealing.

**GAME PLAY**

The player controls a non-denominational “knight” on a quest to reclaim what they have been told is the holy sword of their nation at the behest of the royal family they are sworn to by invading the castle of an enemy nation.  
The knight can be controlled to move horizontally left and right, with the player progressing in the world by finding platforms that will launch them upwards, ultimately having to make a zig-zag pattern left-to-right and right-to-left as they advance higher and higher in the circular-seeming tower.

The player meets several forces of opposition along the way, including enemy spearmen that horizontally charge-dash at the player, spike-pits in the floor, and Holy smiting beams that shoot from the ceiling at specific points, with timed intervals for the player to navigate past them. The player can optionally throw their shield into the Holy smiting beams to disable them and walk underneath the newly created blank space.

The player’s shield serves as both a means of attacking and platforming. The player is able to throw their shield several meters ahead of themselves, allowing them to either leave it in free-floating space as an obstacle to block enemies, or as a means of platforming to jump on top of. Well-timed blocking against enemy dashes enable the player to comedically knock-out enemies by having them trip and fly off of the screen.

At the end of the first level, the player acquires the ceremonial holy sword. Functionally this changes nothing, however, from that point on, when enemies are successfully blocked, the sword floats around quickly from the player’s back to stab at enemies and have them be actually killed in comical fashion.

The second level progresses in the same fashion without a lot of change, however there are more instances of the player needing to perform multiple actions at once (Jumping and shield-throwing; jumping and parrying, shield-throwing and quickly moving past an enemy before recalling and re-throwing their shield to prevent that same enemy from advancing closer).  
As the player has now taken possession of the ‘holy’ ceremonial sword, the sword exists on the player’s back. When the player blocks, the sword will swing around the player of its own accord to stab enemies.

Upon finishing the second level and reaching the top of the tower, the player realises that the sword is not actually ‘holy’, but is possessed by a demon with their nation truthfully worshipping a demonic entity, and that they are in fact the ‘bad guy’ of the story.

**Game Play Outline**

Front menu;  
*Play:* Move to start of level one.  
*Exit/Quit:* Back to desktop  
*Options:* Gore on/off, Music volume slider, SFX volume slider.

After downloading from an app store/website, the player will double-click the game’s icon from their PC to launch the game.  
After a brief splash screen animation (Impulse Squared™), the front menu will appear with the selectable options:  
  
Upon tapping the *Play* button, the player will be taken to a series of a few text boxes that will outline the story and the story reasoning for their ‘quest’.  
The royal family of the player’s home kingdom has beseeched them to seek out a ceremonial holy sword that they claim has been stolen from their treasury. This is told through a very brief (Two or three text box) recounting of the player’s character.

After pressing space or enter to advance the text boxes, the player will be able to control their Knight and begin advancing through the world space of level one. At release, there will be two levels included; both levels are within the same castle ‘spiral’ tower. Both levels will see the player will advance through the different floors of each level from left-to-right and right-to-left in a zig-zag pattern, overcoming platforming obstacles and enemies by utilising their shield as a projectile that can be thrown ahead of themselves that can be utilised as a new floating platform that can be recalled, or used to block enemies. When thrown ahead, the shield will prevent enemies from advancing toward the player, though they will still remain as a persistent threat that will re-orientate once the player either moves past them (By jumping over the shield and the enemy, for instance) or otherwise recalls their shield. The only way to permanently remove an enemy as a threat is to successfully block an attack from that particular enemy.  
Enemies and the player cannot move through each other. Enemies physically “line-up” behind and in front of one another, leading to lines that will attack the player one after the other (Not by intentional design or coding, but as an artifact of the way they are coded to simply ‘chase’ after the player but are unable to clip through each other).

Obstacles that the player finds on their path up the tower several instances of spikes along the floor that require them to throw their shield over the top of the spikes to be able to jump onto it and act as a ‘bridge’ to jump again and clear the spikes.  
There are several places where beams of holy light shoot down from the ceiling, impeding the player’s progress. These may be passed in one of two ways; either by waiting for periods where the beams cease for three to four seconds, or by the player jumping an throwing their shield into the path of the beam, thereby blocking it and creating a gap beneath the shield that they can traverse under without suffering any ill-consequences.

During the first level, enemies that are parried/blocked successfully will ‘trip’ and comically fly off of the screen in a very childish and light-hearted way. After the first level, the sword that is acquired will autonomously swing around to ‘stab’ enemies and kill them, instead of the enemies being tripped and flying off-screen.

*On Death:* Continue/quit

There is no save-game feature.  
The only checkpoint to continue from that is made is at the beginning of the second level.

**DESIGN DOCUMENT**

Game Objects:

*Player*  
The player is a colourful knight without any distinctive religious iconography.  
  
*Health*The of the numeric number of mistakes the player can make. Each time the player touches a stage hazard or is hit by an enemy, they lose a unit of health. The player starts with five units of health. When all units are lost, the player may choose to Continue from the start of the current level, or to quit to the main title screen.  
The player does not have any means of regaining health units other than their natural reset to five units upon choosing to Continue after death.

*Shield*  
The shield is under the control of the player, able to be thrown several metres in front of themselves, embedded into walls, or left as a persistent floating obstacle to enemies. While held by the player, the player is able to parry enemies.  
  
*Enemies  
{Spearman,  
}*  
The enemies of the player are similarly-styled colourful knights, using different colours for differentiation from the player (And possibly different unit types).  
*Spearmen* linearly dash at the player along the horizontal plane, constantly moving toward the player. If any part of the *Spearman* touches the player, the player loses a unit of health.  
  
*Spikes*  
Spikes exist in small dug-out sections of the floors in the castle, posing a stage hazard the player must traverse by jumping over them. If the player touches any part of the *Spikes*, a unit of health is lost.

*Holy Beams*  
Holy Beams serve as top-down stage hazards the player must traverse by either waiting for beams’ period of inactivity, or by utilising their shield’s persistent floating ability when thrown in front of themselves to prevent the beams from reaching their full length to the floor (Along with a jump to create the necessary spacing gap that allows the player to move under the shield). If the player touches any part of the *Holy Beam*, the player loses two units of health.

**Design Guidelines**

The core design of the game is to challenge the player’s intuition to experiment with different ways of using their shield to traverse the world or handle the different threats they face along their vertical progression through the tower.  
There are no mechanical changes to the player’s abilities between each of the levels.

**World Visual Assets**

High Priority: *Player Knight  
Player Knight, with Shield  
Shield Disc (Thrown, in-world)  
Player Knight (v2: model(s) including Holy Ceremonial Sword)  
Player Knight (v2: model(s) including Holy Ceremonial Sword), with Shield  
Enemy Knight (Charger)  
Holy Ceremonial Sword, in-world (As collectible)  
Holy Ceremonial Sword, as-attack  
Spike Hazard (Wooden spikes to hurt the player)  
Holy Light Beam*Medium priority: *Main Menu Background  
Castle Parallax Backdrop (level 1)  
Castle Parallax Backdrop (Level 2)  
Castle Brickwall (Tessellating world block)*Low Priority: *Castle Kingdom Banner  
Castle Flaming Torch Sconce (Animated)  
Castle Flaming Chandelier (Animated)  
Castle Furniture;  
{Table, chair, table-candelabra, dinner plate, cutlery set, bookshelf (Empty, half-and-half [STRETCH:v1, v2, v3], filled), weapon rack (Empty, [STRETCH: Half-and-half v1, v2, v3], filled)  
Castle Door*

**UI Visual Assets**

*Font Style  
Layout as-relevant  
{ Front Menu Choices: Play, Option (Checkbox, Slider), Exit }  
{ Player Info/Interaction: Individual Health Unit, Text Box }  
{ Death Menu Choices: Continue, Quit }*

**Audio Assets**

*Music;  
{ Front Menu theme  
Game start theme  
Level one theme  
Level two theme }  
  
SFX;  
{ Menu: Menu Selection, Menu Interaction, Text Letter Scroll }  
{ Player: Player Footstep, Shield Throw, Shield Hover, Shield Return, Shield-Wall-Impact, Shield Block, Sword-Parry, Player Hurt, Player Jump, Player-Spike-Touch, Player-Holy-Beam-Touch, Player Shield Bounce }  
{ Enemy: Spearman Charger, Spearman Trip, Spearman Fly, Spearman Gibbed }  
{ World: Holy Beam Start, Holy Beam Idle, Holy Beam Fade, Flaming Sconce Idle }*

**Development Hard and Soft requirements:**

Desktop workstation(s), Unity Engine (v.2022.3.18f), Adobe Photoshop, Fruity Loops, Monday.com.

**Project Timeline**

*Gantt Chart:*

A screenshot of a computer

Description automatically generated

A white rectangular object with black lines

Description automatically generated

A screenshot of a computer

Description automatically generated

*Task list:*  
**~~Session Zero/Project Initialisation~~**~~: Feb 28 – 1 Mar~~  ***Complete*  
Research:** 1 Mar – 6 Mar  
**Audio Asset Creation:** 4 Mar – 25 Mar  
**Prototyping v0.1 (Greybox):** 6 Mar – 14 Mar  
**Visual Asset Creation:** 6 Mar – 14 Mar  
**Playtesting v0.1:** 14 Mar – 20 Mar  
**Prototyping V0.2 (Visual Asset Integration):** 14 Mar – 28 Mar  
**Playtesting v0.2:** 28 Mar – 5 Apr  
**Issue Management (v0.1+v0.2):** 4 Apr – 8 Apr  
**Prototyping v0.3 (Audio Asset Integration):** 8 Apr – 19 Apr  
  
*( Red means Critical Path )*

**Visual Guide & Inspiration**

**Game Flowchart**



should represent Objects, Properties, and Actions present in the game. Each of these items exist witu



**Sprite resolution detail/inspiration**

**Team Credits**

|  |  |
| --- | --- |
| Name | Role(s) |
| Phillip Moss | Producer, Writer |
| David Wiggins | Programming, Art Prototyping |
| Jaryie Wong | Testing, Debugging, Proofreading/editing Documentation |
| Sienna Muga |  |
| Yusuf |  |
| “Rick” |  |

**Document Version**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Changes | Date | By |
| 0.1a | Original Creation | 28/02/2024 | Phill |
| 0.2a | Added Team Credits, Asset lists (World Visual, World Visual, UI Visual, Audio). Added Project Timeline (Prospective and dependant on the world being a nice place. Don’t stress, people!). Ends April 19, will be altered as-needed. | 28/02/2024 | Phill |
| 0.3a | Organised Visual Assets by priority list, added further visual directive materials (Sprite sheets/style examples) for artists’ conduct. | 1/03/2024 | Phill |
|  |  |  |  |
|  |  |  |  |