**“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 challenge the player to use their awareness and timing to be able to block enemies from advancing closer to them, make well-timed attacks on enemies, and navigate over obstacles in the landscape.

**Player:**

Single-player-only.

**Technical Form:**

2D Graphics (Flat)

**View:**

Flat-view pane. Side-view 2D.

**Platform:**

PC

**Language:**

C#

**Device:**

PC

**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 will meet several forces of opposition along the way, including enemy spearmen that will horizontally charge-dash at the player, spike-pits in the floor, and Holy smiting beams that will 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 will serve as both a means of attacking and platforming. The player will be 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 will 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 will acquire the ceremonial holy sword. Functionally this changes nothing, however, from that point on, when enemies are successfully blocked, the sword will float around quickly from the player’s back to stab at enemies and have them be actually killed in comical fashion.

**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:  
*Play*  
*Options* (Consisting of volume sliders for music, sound effects, and an on/off checkbox for gore effects to be toggled to the player’s choosing)  
*Exit*  
  
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.

Each of the game levels will follow the theme of a differently configured conveyor belt. Each fruit that needs be sorted will be randomly generated without any pattern. Dependant on the level, a conveyor belt with up to seven ‘paddles’ is presented to the player. Each layout is unique to each level, but is static to that level and does not change. Each paddle is a binary state, blocking off a path of the conveyor belt but leaving another path open. For each paddle, the path of the conveyor belt offers a fork and split in the path.

The player achieving a “win-state” will be dependant entirely on their successful accuracy in sortin the fruits through-out the level. For easy difficulties, this will require 40% accuracy. For medium difficulties, this will be 60% accuracy. For hard difficulties, this will require 80% accuracy. Only fruits that are successfully sorted will be counted toward the player’s score. Unsuccessful fruits do not penalise the player outside of their accuracy rating.

The player failing to achieve the required accuracy per their difficulty setting choices will result in being unable to progress to the next level.  
After three consecutively failed attempts, the player will be offered a ‘pity win’.

The game ‘ends’ after 25 levels, though each level is infinitely replayable.

Enjoyment for the player is derived from the satisfaction to react and manage each fork and split in the conveyor’s path to successfully guide each fruit, making moment to moment choices across each offered paddle placement that will alter the potential outcomes of each fruit’s destination.

The story of the game is that the player is a farmer that needs to organise and sort their fruits to be ready for delivery to market. Unfortunately for the farmer-player, their farmhand has used the same harvest bins while harvesting the different varieties of fruits grown on the player-farmer’s farm, and it is up to the player to rectify the issues to their business brought about by the farmhand.

Levels will end after an allotted timespan.  
Upon completion, the player will be taken to a splash screen showing a synopsis including their accuracy. If successful, they will be able to progress to the next level straight from this screen. If unsuccessful, they will instead be able to replay the current level.  
Both win and lose states of the screen will allow the player to return to the main menu.

**Key Features**

The main features of the game that will attract players are smooth, bright, and flashy animations with whimsical, light-hearted, upbeat sound and music design. The initial attraction is eventually replaced by the intrinsic self-satisfaction of achieving a higher score and clearing harder difficulties.

**DESIGN DOCUMENT**

Game Objects:

*Conveyor Belt*  
The conveyor belt is uncontrolled by the player. The conveyor belt ferries each of the “fruits” continuously along its path at consistent speeds, terminating at several possible “bins” with designations for each fruit  
  
*Paddle*  
Paddles are binary-state barriers that the player can tap on to alternate them between two possible positions. Each position blocks off one path or split along the conveyor belt’s paths, forcing each fruit down the exposed path.  
  
*Fruit*  
*{ Apple  
Banana  
Grape  
Tomato }*

Fruit is non-interactable by the player, and constantly move along the conveyor belt’s path. The player is only able to interact with them by changing the paddles and denying their movement down the pathway that the paddle is positioned across.  
  
*Fruit Bins*  
Each of the fruit has a single bin that they are allowed to be terminated into via the above control of each paddle. Points scores are awarded for successfully bringing each correct fruit into their designated fruit bin.

**Design Guidelines**

The design of the game is to challenge the player’s reactions and ability to think ahead at a steadily increasing rapid pace. The player is restricted in their interactions, only able to interact with the binary-state paddles.  
The player should not be given any direct ability to influence the fruits, nor the conveyor belt itself.

**Development Hard and Soft requirements:**

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

**Visual Guide & Inspiration**

**Game Flowchart**



should represent Objects, Properties, and Actions present in the game. Each of these items should have a number reference

to where they exist within the game mechanics document.

• Menu

• Synopsis

• Game Play

• Player Control

• Game Over (Winning and Losing

Player Definition

● Use this section for quick descriptions that define the player

● Use the Player Properties section (below) to define the properties for each player. Player Properties can be

affected by the player’s action or interaction with other game elements. Define the properties and how they affect

the player’s current game.

● Use the Player Rewards section to make a list of all objects that affect the player in a positive way. Define these

objects by describing what affect they cause and how the player can use the object.

**Player Definitions**

A suggested list may include:

• Health

• Weapons

• Actions

**Player Properties**

Each property should mention a feedback as a result of the property changing.

User Interface (UI)

This is where you’ll include a description of the user’s control of the game. Think about which buttons on a device would be

best suited for the game. Consider what the worst layout is, then ask yourself if your UI is it still playable. A visual

representation can be added where you relate the physical controls to the actions in the game. When designing the UI, it may

be valuable to research quality control and user interface (UI) design information.