Assessment 2 Design Doc

SOUTH PARK CLICKER GAME

Jacob stringer

Contents

[Prototype 1 (Interface Design) 2](#_Toc485504686)

[Main Story Board 1 2](#_Toc485504687)

[Click Story Board 3](#_Toc485504688)

[Power up Story board 3 4](#_Toc485504689)

[Narrative 4](#_Toc485504690)

[Action Lists 5](#_Toc485504691)

[Main Action list 5](#_Toc485504692)

[A cookie click 5](#_Toc485504693)

[A primary power up click 5](#_Toc485504694)

[CDT 5](#_Toc485504695)

[Questions 7](#_Toc485504696)

[Prototype 2 (Read/Write pseudo code): 8](#_Toc485504697)

[Minimum Viable Product 1: Cookie Click Works 8](#_Toc485504698)

[Narrative 9](#_Toc485504699)

[VARS Needed 9](#_Toc485504700)

[Action list 9](#_Toc485504701)

[DDt 9](#_Toc485504702)

[Minimum Viable Product 2: Upgrade CookieClickValue 10](#_Toc485504703)

[Naritive 10](#_Toc485504704)

[Vars 10](#_Toc485504705)

[Action List 10](#_Toc485504706)

[DDT 10](#_Toc485504707)

[Minimum Viable Product 3: Buy an auto clicker 11](#_Toc485504708)

[Narrative 12](#_Toc485504709)

[Vars 12](#_Toc485504710)

[Action List 12](#_Toc485504711)

[DDT 12](#_Toc485504712)

[Minimum Viable Product 4: Buy New Shoes 14](#_Toc485504713)

[Narrative 14](#_Toc485504714)

[Vars 15](#_Toc485504715)

[Action List 15](#_Toc485504716)

[DDT 15](#_Toc485504717)

[Minimum Viable Product 5: Enable Cheat Mode 16](#_Toc485504718)

[Narrative 16](#_Toc485504719)

[Vars 16](#_Toc485504720)

[Action List 16](#_Toc485504721)

[DDT 17](#_Toc485504722)

[Minimum Viable Product 6: Check Prices 17](#_Toc485504723)

[Narrative 17](#_Toc485504724)

[Vars 17](#_Toc485504725)

[Action List 17](#_Toc485504726)

[DTT 18](#_Toc485504727)

[Prototype 3 (Sub tasking) 18](#_Toc485504728)

[Questions 18](#_Toc485504729)

[Main Code 18](#_Toc485504730)

[In the Module Z code 18](#_Toc485504731)

[Updated Story Board 20](#_Toc485504732)

[Prototype 4 (Refactor and Polish) 21](#_Toc485504733)

[Interface Changes 21](#_Toc485504734)

[Before 21](#_Toc485504735)

[After 21](#_Toc485504736)

[Changes 22](#_Toc485504737)

[Final Prototype (End Game) 23](#_Toc485504738)

[Main Story Board 23](#_Toc485504739)

[Naritive 23](#_Toc485504740)

[Enemy Story Board 24](#_Toc485504741)

[Narrative 24](#_Toc485504742)

[CDT 24](#_Toc485504743)

[TimerTakeAwayPoints 25](#_Toc485504744)

[Narrative 25](#_Toc485504745)

[Action List 25](#_Toc485504746)

[DDT 26](#_Toc485504747)

[Power Up Arrows 26](#_Toc485504748)

[Narrative 26](#_Toc485504749)

[DDT 27](#_Toc485504750)

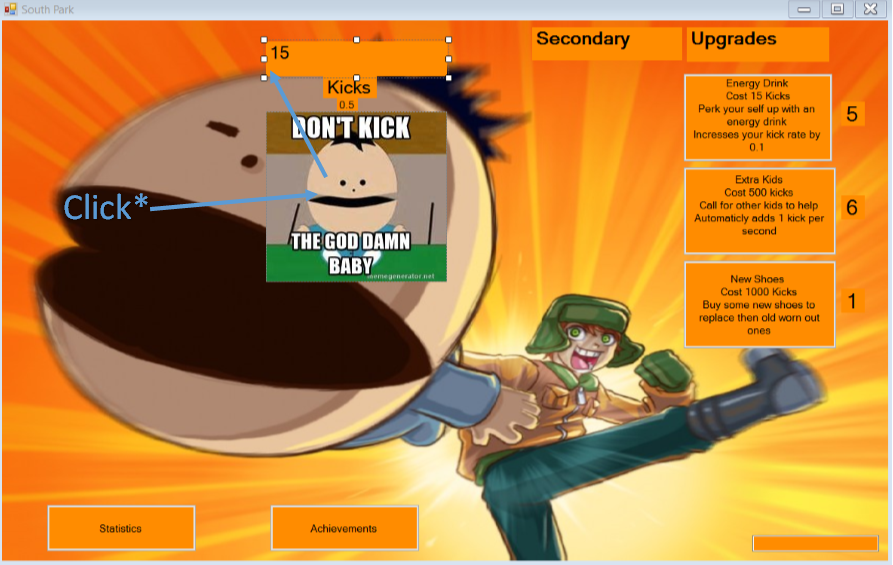
# Prototype 1 (Interface Design)

## Main Story Board 1



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

## Click Story Board



## Power up Story board 3



## Narrative

This is a south park themed clicker it is aimed at people between 18-30. When the game opens, the user can start kicking the baby to earn points. Once they have enough points they can buy 1 of 3 upgrades. The 1st upgrade makes it so your click value increases my 0.1 per each power up brought. The 2nd power up will auto click for you on a timer, It will click once for every power up owned And the 3rd will increases your click value by 1 for each power up brought. You can also save and load you game at any time. You will be able to keep track of your statistics and achievements they have earnt on another form.

## Action Lists

### Main Action list

1-Form Opens

2-User clicks on baby

2.1-Score goes up

3-User buys power up

3.1-Price of power up gets take off score

3.2-User gets extra click value or auto click per second

### A cookie click

1-Form Opens

2-User clicks on picture box

3-Label Updates with amount of clicks

### A primary power up click

1-User clicks power up

2-Removes point from there score

2.1 Checks users score

2.2 Removes the correct amount from the score

3-**Timer()**

4-Updatescreen

#### **TIMER ()**

1-Set timer interval to 1 second

2-Timer Starts

3-Counts to 60

4-Reset timer back to 0

## CDT

|  |  |  |  |
| --- | --- | --- | --- |
| **Control Name** | **Type** | **Events** | **Comments** |
|  |  |  |  |
| 1-FrmSouthPark | Form | - | The Main Window |
| FrmAchievements | Form | - | Achievements window |
| FrmEndGame | Form | - | Enemy window |
| 2-PbKickTheBaby | Picture Box | Click | What you click on |
| 3-lblScore | Label | - | Where Score is Displayed |
| 4-btnAchievements | Button | Click | Opens form to Achievements |
| 5-lblUpgrades | Label | - | This is where the upgrades are |
| 6-lblSecondary | Label | - | This is where you secondary Upgrades are |
| 7-btnStatistics | Button | Click | Opens form to show statistics |
| 8-btnKids | Button | Click | Increases clicks by 0.1 per second |
| 9-btnEnergyDrink | Button | Click | Increases click value |
| 10-btnNewShoes | Button | Click | Increases clicks by 1 per second |
| 11-txtCheatBox | Text Box | - | Allows you to test/cheat |
| 12-lblKicks | Label | - | Name of currency |
| 13-lblScorePerSecond | Label | - | Automatic score per second |
| 14-lblEnergyDrinksOwned | Label | - | Number of power ups owned |
| 15-lblKidsOwned | Label | - | Number of power ups owned |
| 16-lblNewShoesOwned | Label | - | Number of power ups owned |
| 17-btnCheat | Button | Click | Adds cheat value to score |
| 18-btnSave | Button | Click | Saves the game |
| 19-btnLoad | Button | Click | Loads Previous Save |
| 20-btnOpenFile | Button | Click | Shows Variables that are saved |
| 21-btnAchivement100MileStone | Button | Click | Achievement Button |
| 22-btnAchivement500MileStone | Button | Click | Achievement Button |
| 23-btnAchivement1000MileStone | Button | Click | Achievement Button |
| 24-btnAchivementMileStone5PowerUps | Button | Click | Achievement Button |
| 25-btnAchivementMileStone20PowerUps | Button | Click | Achievement Button |
| 26-btnAchivementMileStone50PowerUps | Button | Click | Achievement Button |
| 27-btnAchivementMileStone1000KicksHeld | Button | Click | Achievement Button |
| 28-btnAchivementMileStone5000KicksHeld | Button | Click | Achievement Button |
| 29-btnAchivementMileStone10000KicksHeld | Button | Click | Achievement Button |
| 30-lblAchivements | Label | - | Achievements Appear below |
| 31-lblTimeTillEnemyAppares | Label | - | Time appears below |
| 32-lblCountDownTillEnemyAppares | Label | - | Time till enemy appears |
| 33-btnLoadFromEnemyScreen | Button | Click | Load button if saved from enemy screen |
| 34-lblAllPowerUpsOwned | Label | - | Total amount of power ups owned |

## Questions

1-What modules do you see could come in handy to reduce the amount of code you are writing.

Modules are only used when you want to re-use code over multiple forms. So the only thing I can think of that I would use it for would me updating the screen with scores. I have different forms for achievements and statistics so I think they may help with that.

2-What functions or subtasks could be used that would be reusable and useful?

1. The power ups themselves could possibly be reused if we can figure out how to pass values to it
2. Updating labels would definitely be used
3. And updating the screen may be used a lot.

3-What 3 data types would you consider as top choices to hold the cookie clicker value and why?

1. Decimal Would be good to keep track of the amount of clicks you are doing and to show the score on the form. I could also use double but found out that decimal is a lot more accurate than double is
2. I would use single to keep track of my prices of power ups because it can do decimal places but also uses less memory than decimal would use.
3. I could use Ushort to keep track of how many power ups are owned. I would use this because I don’t think someone will play the game long enough to accumulate 65535 of each power up.

# Prototype 2 (Read/Write pseudo code):

## Minimum Viable Product 1: Cookie Click Works

### Narrative

When you click the baby the user will gain 1 point which is displayed on the screen.

### VARS Needed

Mdkick: Records how much to give you when you click on the baby. Starts at 1

MdScore: Records how many baby you own.  Starts as 0

### Action list

1-Program Starts

2- Clicks Mouse Down

3-Update Screen

### DDt

**On Kicks Mouse Down (Sub Task)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **MdKicks** | Double | 0 -? | Records how many Kicks you own. |
| **MdClickValue** | Double | 0 -? | Records how much to give you when you click on the Kick. |

MdKicks = MdKicks + MdClickValue

Update screen (Sub task)

**Update Screen (Sub Task)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **prDisplayValue** | Label | 0-? | Label used for Main score |
| **prScore** | Double | 0-? | Value Displayed on label |
|  |  |  |  |

prDisplayValue = prScore

## Minimum Viable Product 2: Upgrade CookieClickValue

### Naritive

I can click a power up that will increase how many Kicks I do every time I click the baby. The power up cost me 15 Kicks to purchase and will increases in price by \* 1.2 after purchase. I can only purchase it if I have enough Kicks.

### Vars

MdClickValue: how much a click is worth

MdKicks: Records how much to give you when you click on the baby. Starts at 1

MdDrinkCost: Cost of the power up

lcAmmountOwned: Amount of power ups owned

LcValue: Value added per click

### Action List

1-On Button Click

2-Adds extra 0.1 points per click

3-Takes Cost off power up off Score

4-X 1.2 the amount of Cost next for next click

5-Updates Button text with new Cost

6-Adds 1 to Power up owned

7-UpdateScreen(Subtask)

### DDT

On Baby Click Power Up Click(subtask)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **MdClickValue** | Double | 1-? | Sets the value of each click |
| **MdKicks** | Double | 1-? | Records and displays the amount of kicks done |
| **MdDrinkCost** | Integer | 15-? | Cost of the power up |
| **LcAmmountOwned1** | Short | 1-? | Number of power ups owned |
| **LcValue** | Decimal | 0.1 | Value added per power up purchase |

if MdDrinkCost is less than MdKicks

(Purchases Powerup)

Mdkicks = Mdkicks – MdDrinkCost

(Give Powerup)

MdDrinkCost = MdDrinkCost \*2

(Makes the powerup cost more next time)

lcAmountOwned1 = lblOwned1.Text + 1

(Assigns Value to Variable)

(Update Screen) Subtask

**Update Screen (Sub Task)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **prDisplayValue** | Label | 0-? | Label used for Main score |
| **prScore** | Double | 0-? | Value Displayed on label |

prDisplayValue = prScore

## Minimum Viable Product 3: Buy an auto clicker

### Narrative

An auto clicker is an upgrade that gives Kicks AUTOMATICLY. It runs off a timer that ticks every second. There will be a place on the screen to show how many auto clickers you own.

For each auto clicker you own, every second you get a free auto click BUT auto clicks are only worth 1 Kick.

The cost also goes up my \*1.2 after every purchase

UNLIKE the other power up, it does not increase in click power at every purchase. It stays static at 1 Kick. Upgrading this could be another power up.

### Vars

Mdkps: Variable: used for auto click

MdKicks: Records how much to give you when you click on the baby. Starts at 1

MdKidsCost: Cost of the power up

LcAmmountOwned2: Amount of this power ups owned

LcCountPerSecond: Amount of auto clicks per second

### Action List

1-On Button Click

2- Starts auto clicking on a timer

3-Takes Cost off power up off Score

4-X 1.2 the amount of Cost next for next click

5-Updates Button text with new Cost

6-Adds 1 to Power up owned

7-UpDateScreen(Subtask)

8-UpDateScreen(Subtask)

9-UpDateScreen(Subtask)

### DDT

**On Auto Click Purchas Click(subtask)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **Mdkps** | UInteger | 1-? | Automatically adds 1 every second |
| **Mdkicks** | Decimal | 1-? | Records and displays the amount of kicks done |
| **MdkidsCost** | Single | 200-? | Cost of the power up |
| **LcAmountOwned2** | Short | 1-? | Record how many power ups owned |
| **LcCountPerSecond** | Decimal | 1-? | Records how many auto clicks per second |

If MdKidsCost is less than MdKicks

(Purchases Powerup)

Mdkicks = Mdkicks – MdKidsCost

(Give Powerup)

MdKidsCost = MdKidsCost \*2

(Makes the powerup cost more next time)

lcAmmountOwned2 = lblOwned2.Text + 1

lcCountPerSecond = lblScorePerSecond.Text + 1

Update Screen (Sub task)

**On Timer Tick(subtask)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **MdKicks** | Decimal | 1-? | Records and displays the amount of kicks done |
| **Mdkps** | UInteger | 1-? | Stores the value of the auto click |
|  |  |  |  |
|  |  |  |  |

MdKicks = MdKicks + Mdkps

(Current Score + Mdkps Auto click value)

Update screen (Sub task)

**Do Auto Click (subtask)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **MdKicks** | Decimal | 1-? | Records and displays the amount of kicks done |
| **Mdkps** | UInteger | 1-? | Stores the value of the auto click |

MdKicks = MdKicks + Mdkps

## Minimum Viable Product 4: Buy New Shoes

### Narrative

I can click a power up that will increase how many Kicks I do every time I click the baby. The power up cost me 500 Kicks to purchase and will increases in price by \* 1.2 after purchase. I can only purchase it if I have enough Kicks.

### Vars

MdClickValue: how much a click is worth

MdKicks: Records how much to give you when you click on the baby. Starts at 1

MdNewShoesCost: Cost of the power up

LcAmountOwned3: Amount of this power ups owned

LcValue: Value added per Click

### Action List

1-On Button Click

2-Adds extra 1 points per click

3-Takes Cost off power up off Score

4-X 1.2 the amount of Cost next for next click

5-Updates Button text with new Cost

6-Adds 1 to Power up owned

7-UpdateScreen(Subtask)

### DDT

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **MdClickValue** | Decimal | 1-? | Stores the value of the auto click |
| **MdKicks** | Decimal | 1-? | Records and displays the amount of kicks done |
| **MdNewShoesCost** | Single | 500-? | Cost of power up |
| **LcAmountOwned3** | Short | 1-? | Records how many power ups owned |
| **LcValue** | Decimal | 1 | Value added per power up purchase |

if MdNewShoesCost is less than MdKicks

(Purchases Powerup)

Mdkicks = Mdkicks – MdNewShoesCost

(Give Powerup)

MdNewShoesCost = MdNewShoesCost \*2

(Makes the powerup cost more next time)

lcAmountOwned1 = lblOwned1.Text + 1

(Assigns Value to Variable)

(Update Screen) Subtask

## Minimum Viable Product 5: Enable Cheat Mode

### Narrative

When I type a number into the text box and click on the button it will add 1000 point to my score

### Vars

LcCheat: Amount of score wanting to me added

MdKicks: Records how much to give you when you click on the baby. Starts at 1

### Action List

1-Type in Value In to text box

2-On Button Click

3-Adds Text box value to Score value

### DDT

**Cheat(subtask)**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Range | Purpose |
| LcCheat | Double | 1-? | Value added to Score |
| MdKicks | Decimal | 1-? | Records and displays the amount of kicks done |
|  |  |  |  |

LcCheat = txtCheater.text

MdKicks= LcCheats + MdKicks

UpDateScreen(Subtask)

## Minimum Viable Product 6: Check Prices

### Narrative

Checks to see if you have enough kicks to buy a power up. Runs on a timer

### Vars

prScore: Current amount of kicks owned

prPowerup: Power up button

prCost: Cost of power up

### Action List

1-Checks if score is greater than power up cost

2-If true enables button

3-If false disables button

### DTT

**Check Prices(subtask)**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Range | Purpose |
| prScore | Double | 1-? | Holds Current amount of kicks owned |
| prCost | Double | 15-? | Holds Cost of power up |
| prPowerup | Button | 0 | Gives a power up |

If prScore is greater than prCost

Then prPowerup enabled

Else prPowerup disabled

# Prototype 3 (Sub tasking)

## Questions

### Main Code

**1-What are the first 4 lines of code doing?**

They are diming a variable to = a data type and then passing those variables to a subtask and updating them

**2-What is the Zvars updating line of code doing?**

It’s taking the new Variables and updating the screen.

### In the Module Z code

1. **Why are the vars dimed as md and not made public?**

Because these Variables are only going to be used on this form and nowhere else.

1. **Up-to-date ZVars is doing what?**

It’s making the mdVaribles = the Value that has been handed to them.

1. **Why is it using by ref?**

By ref is always used when passing values, you are expecting to change.

1. **Why is it public?**

All sub tasks or functions in a module must be made public or other form will not be able to see them.

1. **ZVars Updating is doing what?**

It is taking the new values and using them to update the screen.

1. **Why is it using by val?**

By Val is always used when passing values that are not going to be changed.

1. **Why is it public?**

All sub tasks or functions in a module must be made public or other form will not be able to see them.

1. **Update Screen is doing what?**

It’s updating the labels with the new values passed to it.

1. **Why is it private?**

This subtask is only being passed values within its own module not from another form therefore does not need to be public.

1. **What dependency STILL exists?**

All these subtasks still depend on modal level Variables. The main form is dependent on the two public subs.

1. **How would this improve our original code example?**

It would improve it by making it so all variables get pass and passed back from these few subtasks. So, if there are any changes in Values other sub tasks will not grab the wrong value.

1. **What issues or problems might this fix create?**

The main problem will be if you ever want to add any more mdVaribles later down the line you will have to go through every subtask and add it in manually.

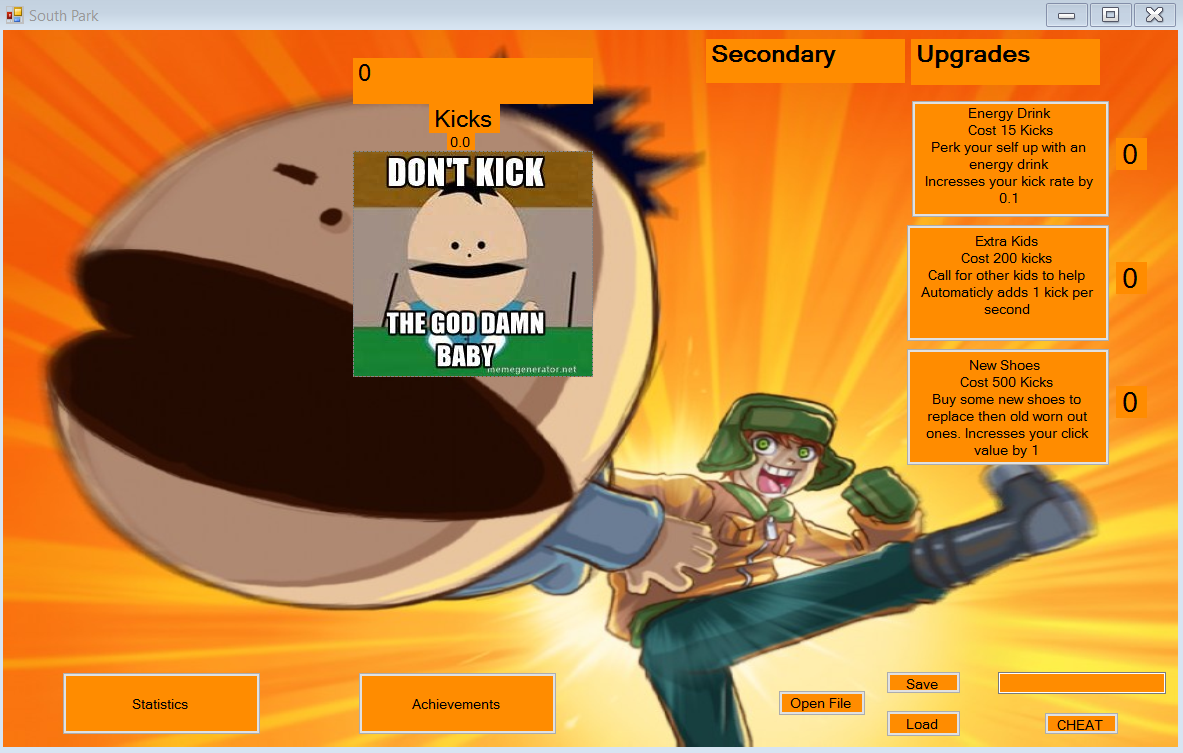
### Updated Story Board



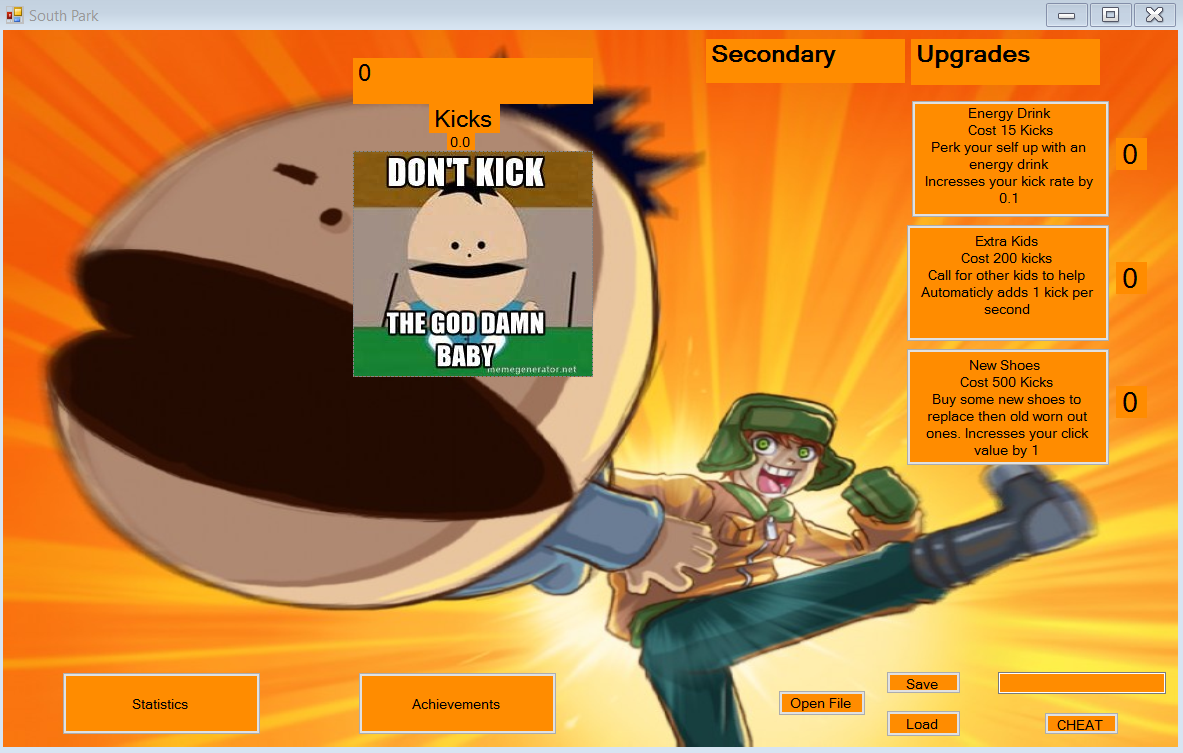
# Prototype 4 (Refactor and Polish)

## Interface Changes

### Before



### After

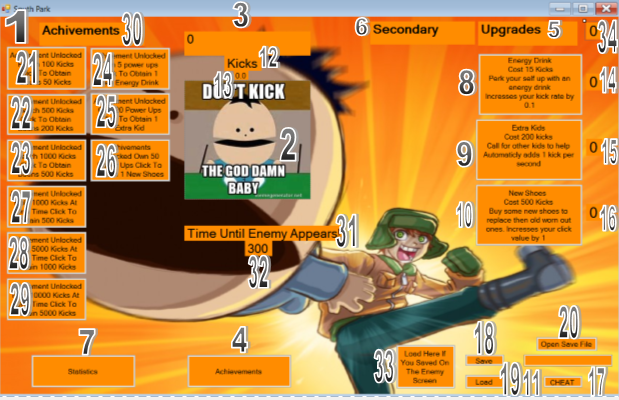


### Changes

I was not able to get a screen shot of the changes but first I added in a message box before the program opens telling them about the game and what to do if they are not too sure how to play. The next thing I added was some tool tips if you place your curser over any element on the screen it will tell you what it does. I made these changes for people playing my game for the first time and are not too sure on what they are meant to be doing.

# Final Prototype (End Game)

## Main Story Board



## Naritive

This is a south park themed clicker it is aimed at people between 18-30. When the game opens, the user can start kicking the baby to earn points. Once they have enough points they can buy 1 of 3 upgrades. The 1st upgrade makes it so your click value increases my 0.1 per each power up brought. The 2nd power up will auto click for you on a timer, it will click once for every power up owned and the 3rd will increases your click value by 1 for each power up brought. You can also save and load you game at any time. You will be able to keep track of your statistics and achievements they have earnt on another form. Once The 5-minute timer on the screen hits zero another form will open with an enemy to kill.

## Enemy Story Board



## Narrative

Once the timer runs out this form will open where you must defeat the enemy. You Click on the enemy to kill it and once it’s health points reach 0 it will take you back to the main form but if you point hit zero a game over screen will show and exit the game. Every time you click it will take away a point off the main score. You can buy 2 power ups one will increases your damage per click on the enemy by 1 per power up owned and the other will decrees your cost per click. You can also save and load the game from this screen.

## CDT

|  |  |  |  |
| --- | --- | --- | --- |
| **Control Name** | **Type** | **Events** | **Comments** |
|  |  |  |  |
| 1-frmEndGame | Form | - | The Main Window |
| 2-pbEnemy | Picture Box | - | Enemy To Click On |
| 3-lblInstructions | Label | - | Instructions For User |
| 4-lblHealth | Label | - | Health |
| 5-lblHealthPoints | Label | - | Has Health Point Value |
| 6-lblKicksLeft | Label | - | Kicks Left Label |
| 7-lblEnemyKicks | Label | - | Where Score Is Displayed |
| 8-btnPowerUpArrows | Button | Click | Power Up |
| 9-btnProtectPowerUp | Button | Click | Power Up |
| 10-btnSaveFromEnemyScreen | Button | Click | Save Button |
| 11-btnLoadFromEnemyScreen | Button | Click | Load Button |
| 12-lblProtectOwned | Label | - | Amount Of Power Ups Owned |
| 13-lblArrowsOwned | Label | - | Amount Of Power Ups Owned |

## TimerTakeAwayPoints

### Narrative

The timer on this screen is running every second. It takes one point off the users score every second it is active. Once the score = 0 the timer will stop and a game over message will show then it will close the form.

### Action List

1-Form Opens

2-Timer Starts

3-UpdateEnemyTimerValues (SubTask)

4-Decreases Score

5-UpdateScreenEnemy (SubTask)

6-If Score = 0 then

6.1-Timer Stop

6.2-Message Box Show

6.3 Close frmSouthPark

### DDT

**TimerTakesAwayPoints (SubTask)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **LcKicks** | Decimal | 1-? | Records and displays the amount of kicks done |
|  |  |  |  |

UpdateEnemyTimerValues (SubTask)

**LcKicks** = **LcKicks** – 1

UpdateScreenEnemy (SubTask)

ReturnEnemyTimerValues(SubTask)

If **LcKicks** = less than zero then

TimerTakeAwayPoints.Stop

MessageBox.Show

frmSouthPark.Close

## Power Up Arrows

### Narrative

When Power up is brought it will add 1 to your click value on the enemy. Then it will add one to Amount of power ups owned. Then it takes way the cost of the power up off your score and increases the cost of the power up. Lastly, the button text will update with additional cost.

### DDT

**Power Up Arrows**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Type | Range | Purpose |
| **LcKicks** | Decimal | 1-? | Records and displays the amount of kicks done |
| **LcDammageToEnemy** | Decimal | 1-100 | Amount of damage done to enemy per click |
| **LcArrowsCost** | Decimal | 1000-? | Cost of power up |
| **LcArrowsOwned** | UShort | 1-99 | Amount of power ups owned |
| **LcValue** | Byte | 1-100 | Value added to click value |

UpdateArrowPowerUp(SubTask)

IncressClickValue(SubTask)

**LcArrowsOwned** = **LcArrowsOwned** + 1

TakeAwayCost(SubTask)

IncressCost(SubTask)

Update Button with additional cost

UpdateScreen(SubTask)

ReturnArrowPowerUp(SubTask)