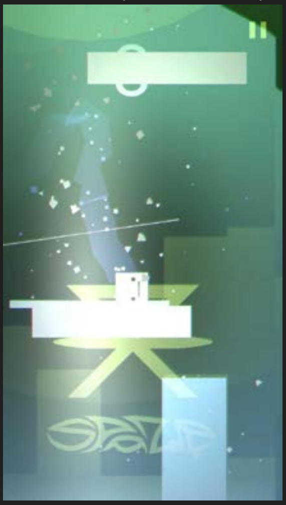
# Software Design & Specification Document – Beat Stomper

Stefan Yuzhao Heng, Kevin Jianxiang Gao

Tues. Aug. 29th 2017

## Product

### Idea

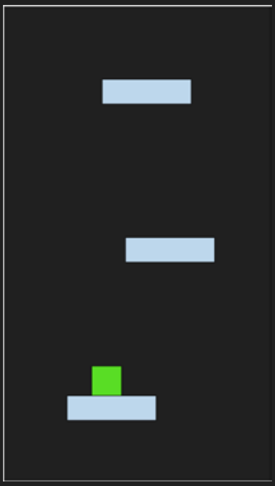
- The *Beat Stomper* on cell phones
  - 
  - Idea like *Temple Run*: To survive and score
  - Move a block as planks fall

## Process

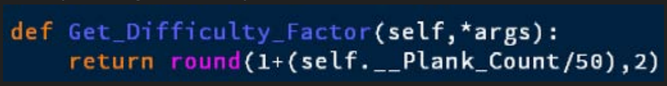
### Design

Idea

- Canvas
  - 16:9
- 3 Planks bouncing & 1 Block jumping

- 

- ➢ Constantly rising difficulty

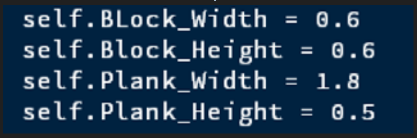
- 

```
def Get_Difficulty_Factor(self,*args):
    return round(1+(self.__Plank_Count/50),2)
```

- ➢ Pause, Restart and Record breaking
- ➢ Music

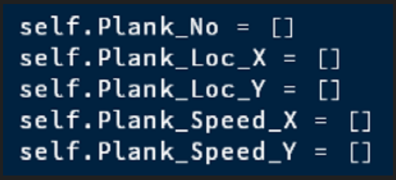
Details

- ➢ Plank width, Block shape

- 

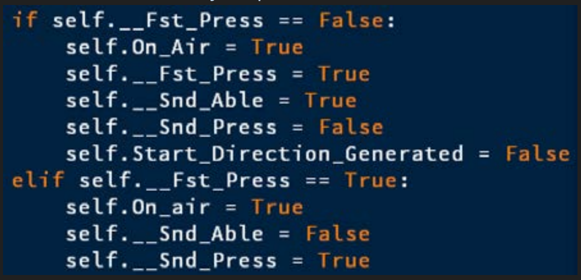
```
self.Block_Width = 0.6
self.Block_Height = 0.6
self.Plank_Width = 1.8
self.Plank_Height = 0.5
```

- ➢ Arrays to store location & speed

- 

```
self.Plank_No = []
self.Plank_Loc_X = []
self.Plank_Loc_Y = []
self.Plank_Speed_X = []
self.Plank_Speed_Y = []
```

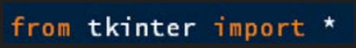
- ➢ Plank starts at random position from top
- ➢ Jump movement
    - Accelerate downwards at all time
    - 1st press: also move horizontally
    - 2nd press: fall straight down
    - Move parallel with on plank
    - Move faster horizontally on air
- ➢ Booleans for 1st or 2nd jump

- 

```
if self.__Fst_Press == False:
    self.On_Air = True
    self.__Fst_Press = True
    self.__Snd_Able = True
    self.__Snd_Press = False
    self.Start_Direction_Generated = False
elif self.__Fst_Press == True:
    self.On_air = True
    self.__Snd_Able = False
    self.__Snd_Press = True
```

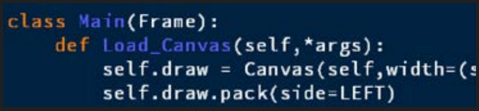
## Coding

- ➢ TKInter for GUI

- No extra installation
- 

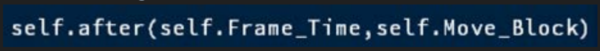
```
from tkinter import *
```

➢ Class and *self.*

- 

```
class Main(Frame):
    def Load_Canvas(self,*args):
        self.draw = Canvas(self,width=(s
        self.draw.pack(side=LEFT)
```

➢ Frame refreshing

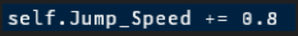
- 

```
self.after(self.Frame_Time,self.Move_Block)
```

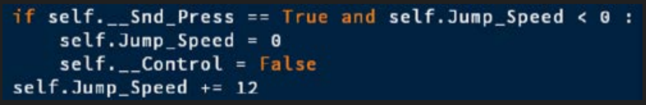
➢ Move plank as speed

Move block as plank or own function

➢ Jump

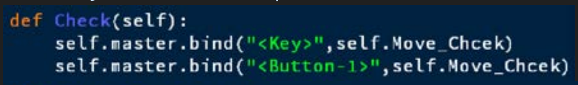
- The delta each frame increases linearly
  - 

```
self.Jump_Speed += 0.8
```

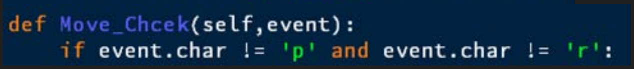
- 2nd press
  - Falls downwards no matter what
  - 

```
if self.__Snd_Press == True and self.Jump_Speed < 0 :
    self.Jump_Speed = 0
    self.__Control = False
self.Jump_Speed += 12
```

➢ Event for key and mouse input

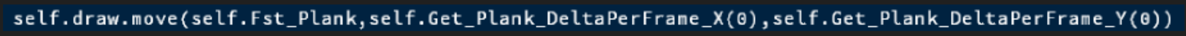
- 

```
def Check(self):
    self.master.bind("<Key>",self.Move_Chcek)
    self.master.bind("<Button-1>",self.Move_Chcek)
```

- 

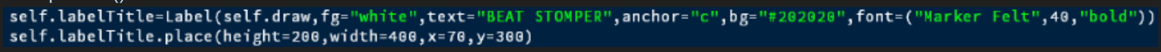
```
def Move_Chcek(self,event):
    if event.char != 'p' and event.char != 'r':
```

➢ Function *move*

- 

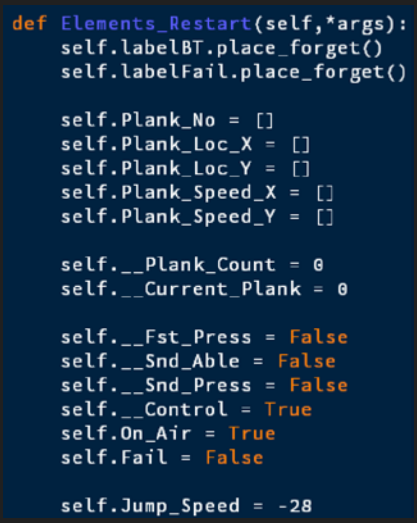
```
self.draw.move(self.Fst_Plank,self.Get_Plank_DeltaPerFrame_X(0),self.Get_Plank_DeltaPerFrame_Y(0))
```

➢ Label & place()

- 

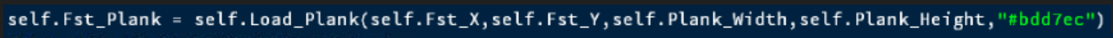
```
self.labelTitle=Label(self.draw,fg="white",text="BEAT STOMPER",anchor="c",bg="#202020",font=("Marker Felt",40,"bold"))
self.labelTitle.place(height=200,width=400,x=70,y=300)
```

➢ Restart variables

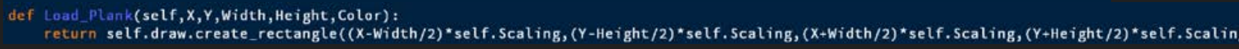
- 

```
def Elements_Restart(self,*args):
    self.labelBT.place_forget()
    self.labelFail.place_forget()

    self.Plank_No = []
    self.Plank_Loc_X = []
    self.Plank_Loc_Y = []
    self.Plank_Speed_X = []
    self.Plank_Speed_Y = []

    self.__Plank_Count = 0
    self.__Current_Plank = 0

    self.__Fst_Press = False
    self.__Snd_Able = False
    self.__Snd_Press = False
    self.__Control = True
    self.On_Air = True
    self.Fail = False

    self.Jump_Speed = -28
```

➢ Formatting & Generalization

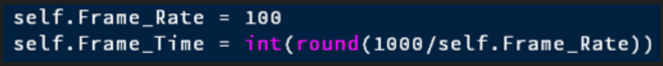
- 960 → self.Scaling * self.Canvas_Width
- 

```
self.Fst_Plank = self.Load_Plank(self.Fst_X,self.Fst_Y,self.Plank_Width,self.Plank_Height,"#bdd7ec")
```

- 

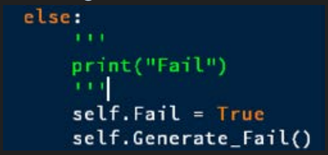
```
def Load_Plank(self,X,Y,Width,Height,Color):
    return self.draw.create_rectangle((X-Width/2)*self.Scaling,(Y-Height/2)*self.Scaling,(X+Width/2)*self.Scaling,(Y+Height/2)*self.Scalin
```

➢ Better control when coding

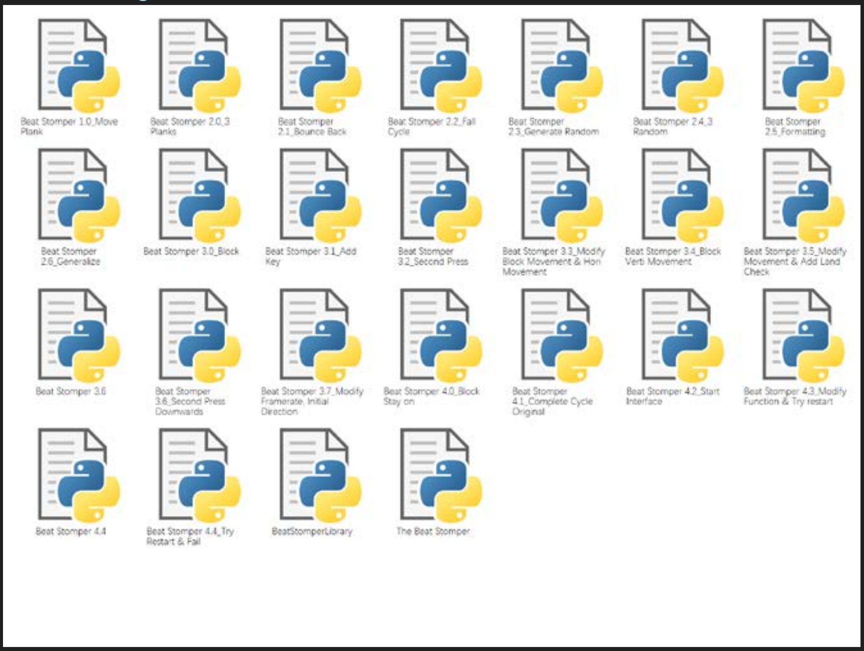
- 

```
self.Frame_Rate = 100
self.Frame_Time = int(round(1000/self.Frame_Rate))
```

## Testing

- Stub Testing
  - 

```
else:
    '''
    print("Fail")
    '''
    self.Fail = True
    self.Generate_Fail()
```

- Black-Box Testing
- Record each bug-free version
  - 
Beat Stomper 1.0_Move Plank
Beat Stomper 2.0_3 Planks
Beat Stomper 2.1_Bounce Back
Beat Stomper 2.2_Fall Cycle
Beat Stomper 2.3_Generate Random
Beat Stomper 2.4_3 Random
Beat Stomper 2.5_Formatting
Beat Stomper 2.6_Generalize
Beat Stomper 3.0_Block
Beat Stomper 3.1_Add Key
Beat Stomper 3.2_Second Press
Beat Stomper 3.3_Modify Block Movement & Hori Movement
Beat Stomper 3.4_Block Verti Movement
Beat Stomper 3.5_Modify Movement & Add Land Check
Beat Stomper 3.6
Beat Stomper 3.6_Second Press Downwards
Beat Stomper 3.7_Modify Framerate, Initial Direction
Beat Stomper 4.0_Block Stay on
Beat Stomper 4.1_Complete Cycle Original
Beat Stomper 4.2_Start Interface
Beat Stomper 4.3_Modify Function & Try restart
Beat Stomper 4.4
Beat Stomper 4.4_Try Restart & Fail
BeatStomperLibrary
The Beat Stomper

## Reflections & Improvements

- Reflection
  - Start earlier when you can; you never know how much time debug could take
  - For codes, it's not *the longer, the better*
  - Its' not simple as adding *self.*
  - Always remember to save your library after change
- Improvements
  - Gradually changing background color
  - Background music
  - History record
    - File
  - User settings
    - Define Difficulty