Personals

Name: Joshua Brook
Student ID: 13160869

Overview of plans

This game will expand on and collate both the 'Mondriaan' and 'Red Circle' interactive programs to create an arcade-style game in which one attempts to dodge moving rectangles. I think that one if the biggest problems to overcome will be programming this collision detection. One can also shoot these rectangles to decrease their size. This will be achieved my means of a laser emanating from the player in the direction of movement. I will also be adding a timer to score the player as well as functionality for ending the game when the player loses.
I'm a bit young to have grown up on space invaders, but I've been playing video games my whole life and have always wanted to build my own. This project will draw lots of inspiration from mobile games that I played during my childhood.

Technical challenges, contingencies and stretch goals

Uncertainty: I'm not sure how I'm going to code the collision detection, but I think with some time and some research I should get there. I plan on taking small steps and thinking ahead so that the project can go as smoothly as possible.
Stretch goals: Adding collectable powerups which can make the player move faster or destroy rectangles more quickly. I might also add custom sounds and animations for shooting/destroying rectangles. I think there's a lot of different functionality that I could add to this game, so I plan to continuously add new functions as I go.

WorldState

```
(struct Rectangle [x y vx vy col wid hei] #:transparent)
; A Rectangle is a structured type: it contains
; - x, y: (Numbers) its position
; - vx, vy: (Numbers) its movement speed
; - col: its colour (a String)
; - wid, hei (Numbers): width and height of the rectangle.

(struct Player [cx cy keys] #:transparent)
; Player is a structured type which defines the position of
the player on the background
; It contains:
; - cx: a Number: x position of the player
; - cy: a Number: y position of the player
; - keys: a Keys, indicating which directional keys are
currently pressed

(struct Keys [l r u d] #:transparent)
; Keys is a structured type. it contains:
```

```
; - l, r, u, d to track which directional keys are currently
pressed

(struct WorldState [player reclst] #:transparent)
; The WorldState is a structured type: it contains
; - player: a Player structured type
; - reclst: the list of currently shown rectangles

This will most likely need to be updated again during the
course of the project, to include aspects such as shooting
```

Steps

Step	Description	Estimated time	Confidence
1	Update the WorldState to include information about Player and Rectangles	20m	95%
2	Add new colours and shapes to rectangles Design player model/background etc	15m	95%
3	Adapt BigBang (including handlers) to accept new WorldState	1h	80%
4	Introduce Player and adapt code again to include new movement controls	1h30	85%
5	Add collision detection (will take a lot of research and testing I think)	2h	70%
6	Add shooting mechanics – either in direction of movement or on mouse	45m	90%
7	Make rectangles shrink/get destroyed when shot Should hopefully be able to use same collision detection	45m	85%
8	Add timer function to give scores to the player	30m	90%
9	Add game over screen	20m	90%
10	Design collectable powerups (extra speed, better shooting, etc) Might also add negative aspects, like more or faster moving rectangles	1h	85%
11	Custom sounds for shooting, destroying rectangles, movement, etc	1h15	95%