DM 2295 Game Development Game

Academic Year 2014/2015 Semester 2

Final Report

Team number / Name: 10 - Something

Project Title: Racial Harmony

Members : Kennard Kee (130214X)

Pang Kee Yang (132385W)

Isaac Lee (132248J)

Sherwyn Sin (133911P)

Pang Han Liang (132373L)

Game Idea/Concept:

Player will start off as a human. Either as a Chinese or a Malay, John or Mala. Conflicts between two races sparks unhappiness among the separate community. However, love go beyond boundaries. Nothing can stop the relation between John and Mala. They endured through criticisms and hardship. At the end of the conflict, a mutual view has been seen upon, they lived happily ever after.

Project Schedule:

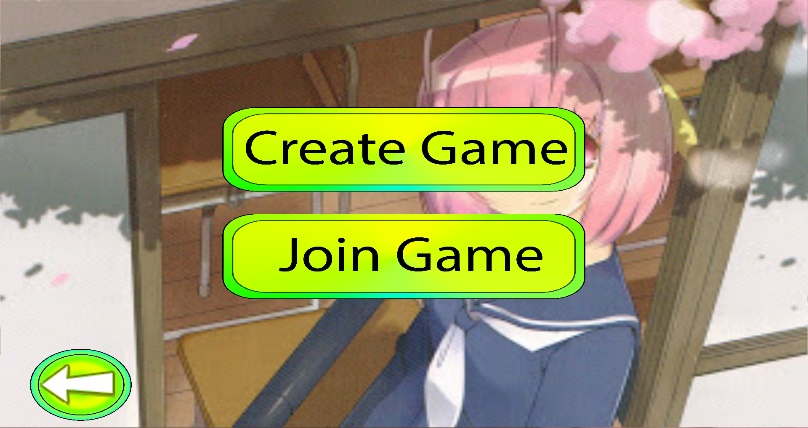
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Done by | Features | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 1 | isaac | Movement |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | isaac | Controls |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | isaac | health system |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | isaac | point system |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | isaac | ai |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | kee yang | object mangement |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | kee yang | sprite |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | kee yang | game states |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | kee yang | ui |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | kennard | physics |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | kennard | gravity |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | kennard | levers and doors |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 | kennard | multiplayer |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | sherwyn | map |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | sherwyn | Specifc character blocks |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 | sherwyn | power ups |  |  |  |  |  |  |  |  |  |  |  |  |  |

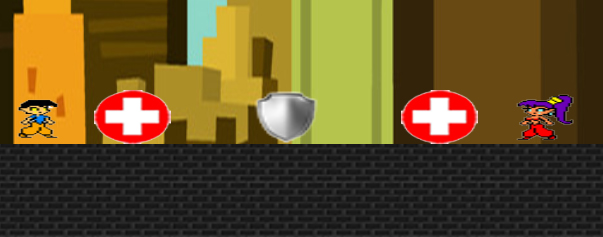
Task Breakdown:

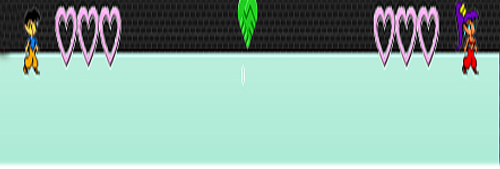
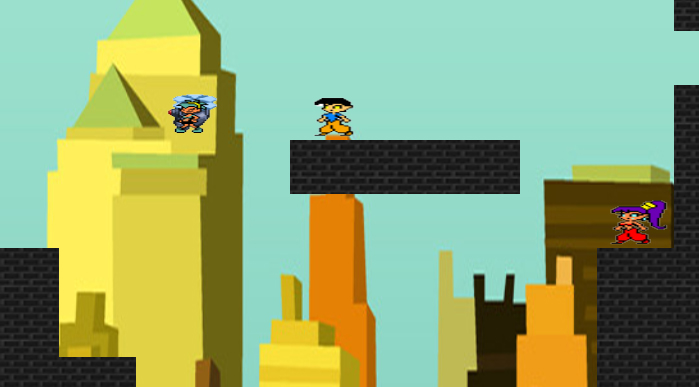
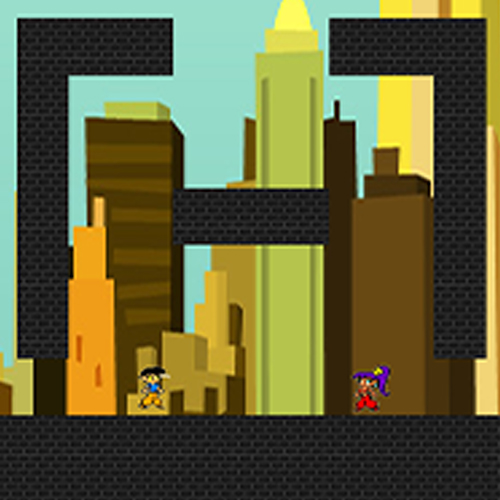
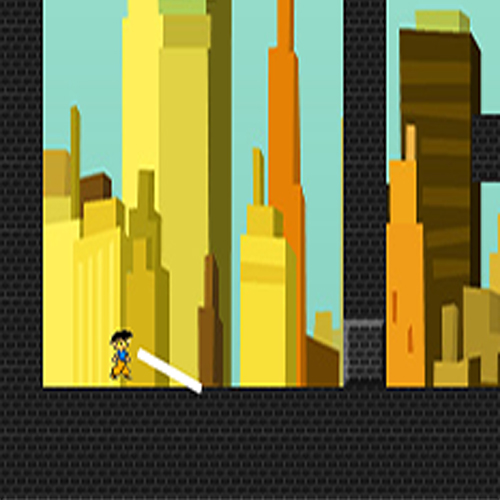
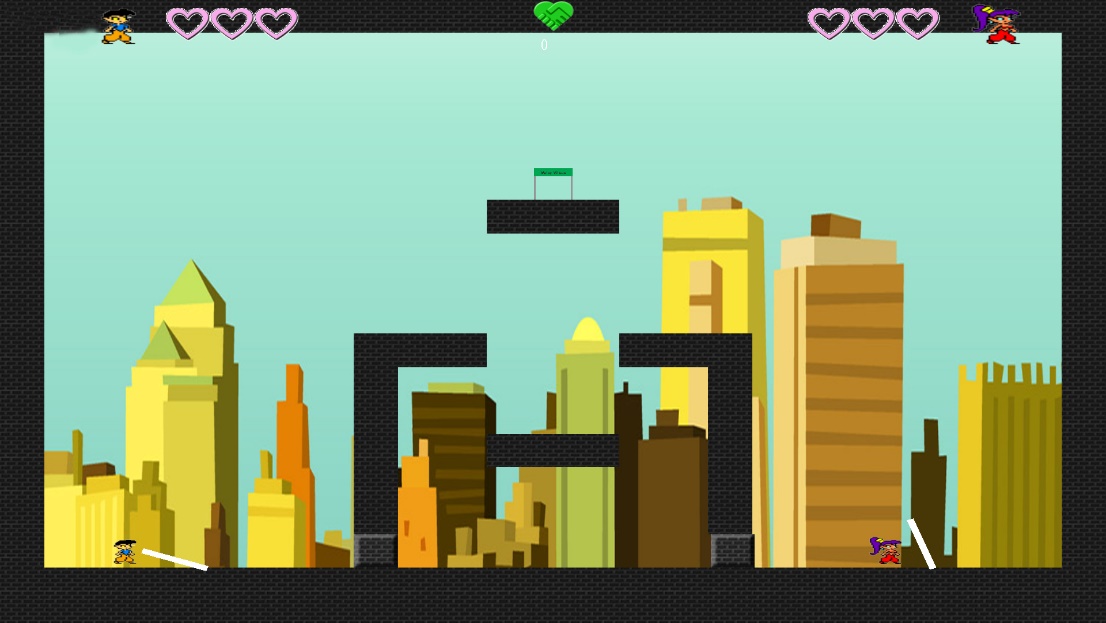
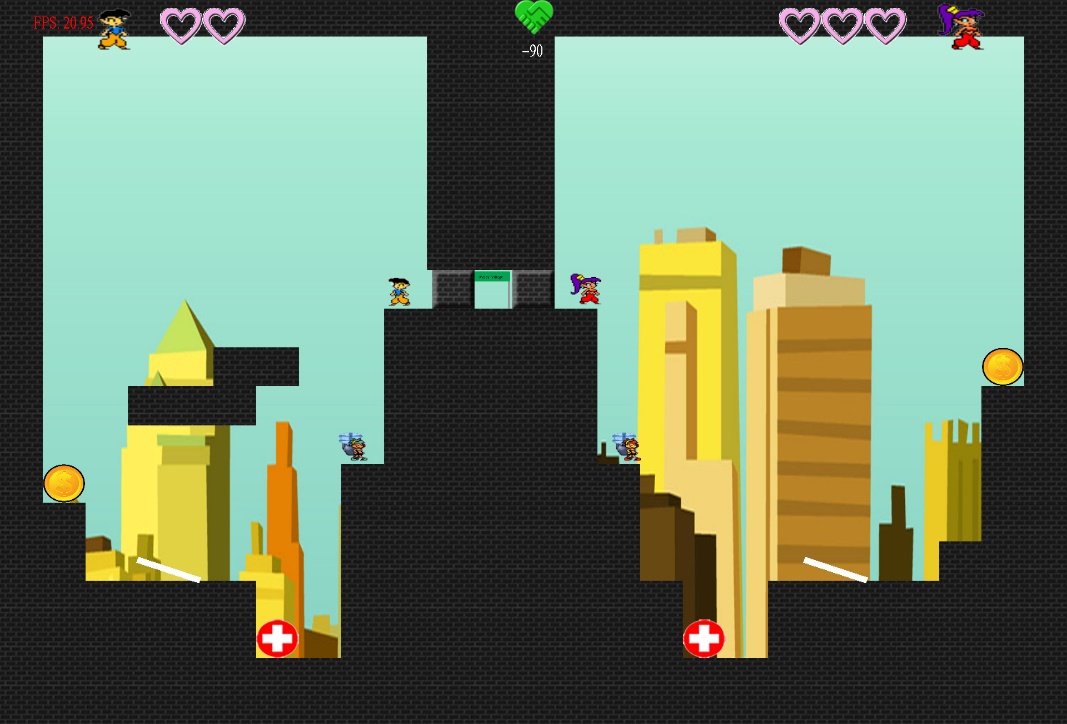
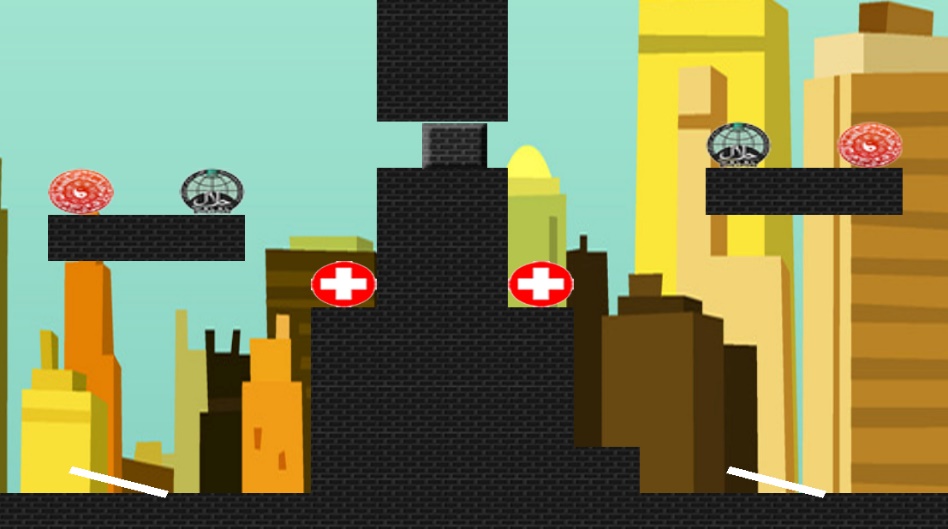
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Task | Description | Workload/Effort Points |
| Isaac | Character Base | Base Class for Character, movement is done here | 1 |
| Isaac | Character Sub | Sub Classes for Character, rendering of sprite and animation is done here | 1 |
| Isaac | Keyboard Inputs | Buttons on the keyboard to be pressed to interact with the game | 1 |
| Isaac | AI Base | the base and subclasses of the AI, rendering of sprite and animation is done here | 7 |
| Isaac | AI pathfinding | A\* AI pathfinding, AI can move to its destination properly, read the map and change accordingly | 9 |
| Isaac | AI Logic | AI logic, Finite state machine and movement updates. AI chase the specific character its suppose to chase when in detection range and return to original position when out of range from it, player and original position alike | 7 |
| Isaac | Health | Class for storing and managing health that each character class contains | 2 |
| Isaac | Points | Singleton Class for storing and managing points | 2 |
| Sherwyn | Map Loading | Loads map for a csv file | 2 |
| Sherwyn | PowerUps | Type include, health recovery, points,in | 5 |
| Kennard | Multiplayer | Uses raknet for multiplayer | 9 |
| Kennard | Physics | A component that gives physics and collision to objectsa | 4 |
| Kennard | LeverDoor | Despawn or respawn a door based on lever state | 8 |
| Kee Yang | All of UI | Overall Ui Design, all of the coding related, such as rendering, alighment, mouse detaching on the buttom | 5 |
| Kee Yang | Ui objects | Rectangle and circular button | 4 |
| Kee Yang | Sprite ,sprite fade class | The sprite class. And a decorator class for fading effect | 5 |
| Kee Yang | Object Creation,factory,builder, and overall management,object manager | Main control point of all object update, collision calls,creation ,addition , deletion,recycling of objects | 8 |
| Kee Yang | Game State and Game State manager | All the states in the game such as Menu,intro,option,. | 7 |
| Kee Yang | Music System | The main control point of all audio related objects. | 5 |
| Kee yang | Image manager | The main control point of management to store and return the TextureImage upon requested. | 5 |
| Kee yang | Framework | The starting of framework for the SP4,which includes others class brought from other modules.  Such as Mouse,Keyboard,LUA ,Game state framework | 6 |
| Kee yang | Spatial partition | A class used to segegrete the world of program for better efficient. | 8 |
| Kee Yang | Intro animation | Intro animation for the program | 3 |
| Kee yang | Frame rate manager | A main object to calculate FPS and deltatime | 3 |
| Kee Yang | Window Manager | A Main object to store information about the window,and calculate essential information such as ratio. | 3 |
| Kee yang | Minor marcos | Small header file to do code definition sentry to allows code versioning,such as preload texture or load as state proceeed. | 1 |
| Everyone | Finding of resource,texture and audio |  | 6 |
| Everyone | Debug and testing playing |  | 6 |

Screenshots:









Features:

**Physics**

It’s a component that gives basic gravity to the objects that calls it in the update

This includes a general aabb collision that can be used for all objects

**Power Up**

Power Ups include adding to health, adding of points and the invulnerability

These are displayed on the game screen using the add object and tile map. As such, one is from csv and the other is bind above on screen. Players collects the power ups and changes are made to the character.

**Obstacles**

Obstacles include deducting health to specific character. Displaying of the obstacles is almost similar to the power ups. However, obstacles does not make changes to all the characters. Instead, it only targets to specific character. For instance, one sign only makes changes to the Chinese character while the other character and just walk over it without taking any damage.

**Map**

Loading of Csv files, level loading and adding objects to the space. Objects are placed in both tile maps as well as adding object to the space on the screen in the space of the tile.

**Music System**

A central point where all member can use it to play music and sound.The main intention is to encapsulate,organize and manage all audio related object at background ,so that the user can have no knowledge of irrklang to use. However it any fanciful requirement is not support as it is meant to manage all the background data management.

**Ui**

All of Ui related stuff including all rendering,alignment,uibutton classes.

Button class uses simple collision detection check from programming physic.

**Sprite,Image manager.**A simple sprite class that allow user to have multiple animation layer,varying speed and some small animation trick such as lateral inverted rendering.  
Image manager is a central point of all image data, which the manager will find and return the valid requested image upon request from user.It is used as a singeton object.It main intention is to have a simplier storage and finding of image data.

**GameState and Manager**Multiple Game state such as menu,credit,option, are linked and transitied using GameStateManager.

**ObjectManager,Manufacture manager,Spatial partion.**

Spatial partion is object is able to register into spatial partition and used for more efficient upate calls.Able to handle moving object and object with varying sizes.

manufacture manage is a central point object that allow all member to create each other object with any knowledge about what went behind the creation.

Object manager is the highest level of central point of all object management,which manage collision calls,render calls, deletion,creation and addition.

**AI**

AI logic containing a finite state machine to control the states of the AI along with deterring the AI characteristics . AI moves towards and chase players when too close and returns to it original position after losing track of the player realizing its out of the saferange away from its original position. AI uses A\* pathfinding to find a path to the

destination it wish to go, be it towards players or back home. Interaction response to the player is done as well,

decreasing his/her health

**Character**

keyboard inputs to control both characters, base classes that handles its movements, sub classes to handle its

specific characteristics like its ID and Tag as well as the sprite rendering, what sprite to render when turning,

moving or staying still as well as the timer for invulnerability after being hit by enemy AI or after picking up

invulnerable powerup.

**Health and Point Sytem**

A Health class that holds value of the health points of the chracters, each character has one health class in them.

A singleton Point class that holds the value of the points of both characters, both character share the same point

system. UI use this two class for the values to display.

Applied Knowledge:

Kennard:

Data Structure and Algorithm

Knowledge from programming physics was used for the lever collision

Stuff from multiplayer helped a lot as well such as the limiting of the number of packets sent per second

Issac

Data Structure and Algorithm

Ai Module used for AI for finite state machine

Self-research for A\* pathfinding in AI

Sherwyn

Data Structure and Algorithm

Knowledge from Gdev has been applied for map

Kee yang

Data Structure and Algorithm

Design Pattern(Builder,faced,decorator,singeton,object pool)  
 AGdev stuff, Spatial partition

Computer graphic, rendering.  
 Irrklang sound .

Lessons & Problems:

* Lack of communication between members lead to all of us not knowing what the game will be like
  + This lead to us having to have a meeting after our supervisor pointed this out to us to clear out all the misunderstandings about the framework and codes
* Underestimating certain features lead to lesser time being allocated to that particular feature
  + This forced us to reallocate time from other features

Future Improvements:

* More Levels
* More Game Elements