The scope of Run n Jump

Basic goals of the project summed up -

* Name of the game - Run n Jump
* 2D side-scrolling (‘side view’) plat-forming game
* Tile-based
* ‘Super Mario’ Clone
* Platforms: Android and Windows 10.
* A number of enemy monsters with different AI
* At least 3 different power ups.
  + Power up ‘mini-game’ at least 2 different mini-games. There will be limited time to complete the mini-game.
    - Multiple choice question at least 20 different questions
    - Sentence ordering mini game 4 sentences or pieces of a sentence will be given and the player will need to arrange them in order. At least 20 different sentence orderings
    - Mini games give more score points the faster they are completed.
  + A key will need to be pressed to pick up a power up.
* Accessibility features for the
  + Visually impaired and colour blind
  + Physically impaired
* 2 game modes
  + Campaign/main game mode with different levels
  + Endless run - procedurally generated endless level where you need to survive as long as possible
* Scoring and leader-board system.
  + Ways to earn score points - picking up coins, power-up mini-games, time taken to complete the level
  + Online integration
* It aims to raise awareness about the serious issue of human trafficking

I will be recreating the old Super Mario arcade game which I will call Run n Jump, it will be a tile-based side-scrolling platformer aiming at mainly the Android platform but also Windows 10. It will feature enemy monsters with differing AI based on its type. It will compete on the Android and Windows markets with other arcade games.

It will be a serious game in that it will raise awareness of human trafficking and kidnapping through the use of ‘mini-games’ embedded into the power up system - when a power-up is picked up a mini-game appears, the first mini-game type will be a multiple choice question, with at least 2 and at most 4 options, a question will appear related to trafficking and you will need to answer correctly, the second mini-game type will be a sentence ordering mini-game, initially at the start of the game this type will be more common - this will be the main way the player is educated about the issue of trafficking. There will also be text signs around the game world which the player can interact with and read about the issue - the incentive is to be able to complete the power-up mini-games quicker. Super Mario series often features kidnapping in a light-hearted joking manner, as a bare-bones story for the game. My game will feature trafficking in a serious way and seek to educate people on what to look out for.

Furthermore I want the game to have two game modes, firstly a classic Mario mode with different levels and the goal is to reach the end as quickly as possible and collecting as many coins as possible. The second game mode will be an ‘endless run’ mode, with the idea that a giant enemy is chasing the player and he needs to run through obstacles as quickly as possible to survive, in this game mode the key will be the proper utilisation of power ups gained through the ‘mini-games’. The goal of the second game mode is to survive/run as long as possible, the map for the second game mode will be procedurally generated. Although in theory this will be an ‘endless’ game mode, the obstacles will become progressively harder to get through and the enemy will be quicker; the longer the player has survived ensuing there is a moment when the player loses.

The game will also strive to be accessible to all potential players. I will make the game playable for people visually impaired by making the text large and readable on all resolutions and screen configurations with a simple clean font and high contrasting colours, ie no green text on a light green background. Furthermore I will aim to use colours which will enable a colour blind person to play with no issues. Furthermore there will be a special configuration in the options which will allow the user to play with only one hand to be more accessible towards people physically impaired.

The game will feature a scoring system, where a player will need to pick up coins to gain score. There will be optional routes during the main game mode levels where players can go a side path to acquire extra coins. The scores and leaderboard will be kept online and updated often allowing for the player to compare his/her scores with other people all over the world.

The Platforms

The gameplay experience will not differ in a significant manner between the different platforms, although there will be a few differencs. One difference will be that the Windows version will have the ability to change resolution of the game. Another difference will be between the control schemes, due to android being a touch screen device and Windows being mostly mouse and keyboard based. Additionally the game likely will be harder to play on Android due to the touch screen and typically smaller screen, due to this I will keep the leader-boards separate for each platform. The game will use basic graphics so the visual differences will not be significant.

Android will be my main platform I’m focusing my development efforts on, Android is a mobile platform for mainly smart-phone and tablet devices with a touch screen. It will be the platform with the best ‘user experience’ due to it being where I will conduct most of my development and testing for. The user will control the character through the use of an on-screen touch ‘joystick’ and an interact and jump keys. The API level I will target will be 21, or android platform version 5.0, which will make my game compatible with over 94.1% Android devices which use that version or newer. Although this may change during development if I will be requiring newer Android API features or if my library dependencies require an older version.

Windows will be my secondary platform, it will have a lower priority than my Android version but the goal is to make it fully playable and have a good user experience. I will be developing and testing for Windows 10, the game may or may not work with older versions of Windows. The player will control the game through the use of the keyboard and mouse. The mouse will mainly be used for navigating the main menu and mini-games though there will be a way for utilising only the keyboard for this as well. The keyboard controls will be used for controlling the player character.

Deliverables to be achieved by Week 11

The goal of my first 11 weeks is to develop a Minimum Viable Product for Android which will be a playable but limited game experience, it will contain the following features

* Menu and Sound systems
* On-screen joystick touch controls
* Working character movement
* Campaign mode
  + 1 playable level
* Basic offline scoring and leader-board system
* At least 1 power up type
* 1 mini-game type
* 1 type of enemy
* Animations

Development Plan

The intention is to build an Android MVP by week 11 which would be my foundation for further development. I will implement the project using libGDX game programming framework/library and Android Studio using the Java programming language I will have two sub-projects for the Android and Windows 10 versions where platform specific code will be. I will use the in-built Android Studio emulator and my own Android device Huawei P8 to test the application. I will use the Overlap2D library for my 2D level development. Initially my priority will be to get a basic minimum viable product by week 11, to achieve this goal I will focus on the Android version and only do minimal Windows development and testing during this time. I will only have a playable MVP for windows 10 by week 11 in the unlikely best case scenario of the project being ahead of schedule.

The game systems and their estimated use of work-hours to implement to an acceptable standard (‘basic’ implementations will take much less time) -

* Menu - 10 hours
* Sound - 5 hours
* Camera - 10 hours
* Player character control - 35 hours
* Tile-based map - 20 hours
* Campaign mode levels - 50 hours
* Endless game mode - 50 hours
* Enemy AI - 20 hours
* Mini-games - 20 hours
* Power-ups - 10 hours
* Scoring and leader-board - 20 hours
* Animation system - 15 hours
* Graphics - 15 hours
* Accessibility - 10 hours

Total projected implementation time - 290 hours

Below is my weekly plan, each week’s work is intended to be started during that week, but it is intended to be finished by the following week’s end. This is so there is sufficient time for every week’s work.

Week 2 - This week aside from Challenge week deliverables I will begin the development by creating a libGDX project using the libGdx project setup tool. I will create the project with all the official libGDX extensions aside from the Bullet extension which is used for 3D development. I will also install the third party libGDX extension used for tile-based level development ‘Overlap2D’. I will read the libGDX and android documentations extensively.

Week 3 - This week I will design the game main menu, options menu, level choice window and in game user interface. I will draw and label it describing its functionality in a word report, the drawings will be done with paint or a similar program. I will also check if the libGDX project is set up properly and do basic programming if it appears necessary.

Week 4 - I will start developing the menu system in Android Studio using libGDX, according to my design. Making changes and adjustments if necessary. I will draw some menu elements myself or use free assets. I will ensure the menu can be traversed properly on Android.

Week 5 - I will start development on the sound system, I will add some background music and menu sound effects. I will utilise free sound assets for this. I will start developing the 2D tile-based map for the first level of the campaign game mode using Overlap2D. I will utilise a mix of free assets and my own drawings for the graphics.

Week 6 - I will also begin work on a bare-bones prototype version of the player character control system allowing for the character to run and jump.

Week 7 - I will continue working on the full player character control system. I will implement on-screen touch joystick controls for Android based on the Android UI design from week 3. I will also implement the player camera system - make the camera follow the player character. It is intended that by now there is at least part of level one done and that the player can walk and jump around the level with basic working collisions.

Week 8 - I will implement enemies and their AI into the game and add them to the level. I will fully implement collisions and their interactions I will also utilise a mix of free assets and my own drawings for the enemy sprites.

Week 9 - I will work on adding animations to the game for the enemies, terrain and player character and a basic scoring system.

Week 10 - I will work on the mini-games and add lower gravity and speed boost power ups. After this the focus will be on bug-fixing and code clean up. If project is ahead of schedule work should be done on the Windows version - platform specific code such as the windows keyboard and mouse control schemes. This is with the intention that there will be a playable Windows MVP. But this will only be done if the android MVP is finished to an acceptable standard.

Week 11 - This is the last week before the interim oral examination, by now the first level of the campaign game-mode should be fully playable. With the systems outlined above being implemented at a basic level at worst, if not; work should be directed to those unfinished systems so that the Minimum Viable Product will be fully playable on Android. Bug fixing and code maintenance work should be done during this week.

Weeks 12-14 - I will perfect enemy AI, and add more types of enemies and new AI types and obstacles to levels. I will also begin the process of making everything function on Windows. From this moment forward I will develop both versions in parallel.

Weeks 15-17 - I will finish fully the development of the first level of the campaign and begin developing additional levels. I will do more sound effect work, This will be done over the following several weeks.

Weeks 18-21 - I will intensively work on the endless run game mode. I will do work on procedural generation of the level allowing for an endless game experience. Also I will implement the endlessly chasing enemy and its AI; the player will need to run from this enemy in this game mode.

Weeks 22-24 - I will focus focus on the development and perfection of the mini-game and power-up systems and add new power-ups.

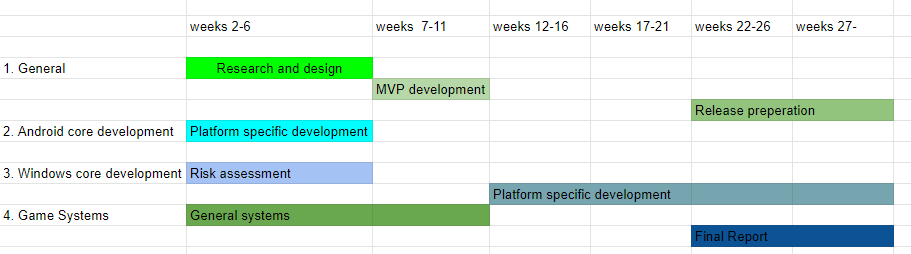
Weeks 25-26 - I will perfect and implement the advanced form of the scoring and leaderboard system and add online integration.

Week 27 - I will do final finishing up of the game and its systems.

Week 28 - This week I will focus on user accessibility features.

Weeks 29-30 -I will focus on polishing the game in every way. Making collisions more accurate, fixing bugs, making the game more optimised etc. I will conduct extensive testing of the game before my final report submission.

Implementation Plan



Risks

Every software development project has risks due to many factors. Projects can fail or be delayed due to unforeseen factors such as unexpected difficulties in implementation, badly-defined system requirements, wrong estimates of project resource requirements etc. As the project gets bigger more time will need to be spent towards maintenance and code clean up to avoid failure and ‘spaghetti code’.

To mitigate these risks when planning my deliverables my focus was on under-promising and over-delivering. This is why I am focusing on just the Android platform up until week 11 despite my project being intended for Windows too. Only in the case of being ahead of schedule I will put my work time resources into fully developing the Windows version by week 11. Additionally I will use my implementation plan as my template for development. I will use Jira extensively to plan my work. Many project managers overestimate their team’s capability and downplay the possible risks that is why my plan will be realistic in its estimation of what I am capable in eleven weeks. It is better to be conservative in planning and be pleasantly surprised rather than be over-optimistic and then disappointed in the development result.