Pipeline Training

Jake O’Connor

UAT MS509

Pipeline Project: Part 4

Overview

This document aims to outline the proposed pipeline for the Victorian Motocross team and provide instructional material for new members of the team that come on board in the next semesters. The instructions are laid out into four process steps: inception, ideation, implementation, and validation. These four steps are rough boundaries for the stages that a single unit of work often goes through in a professional game development environment.

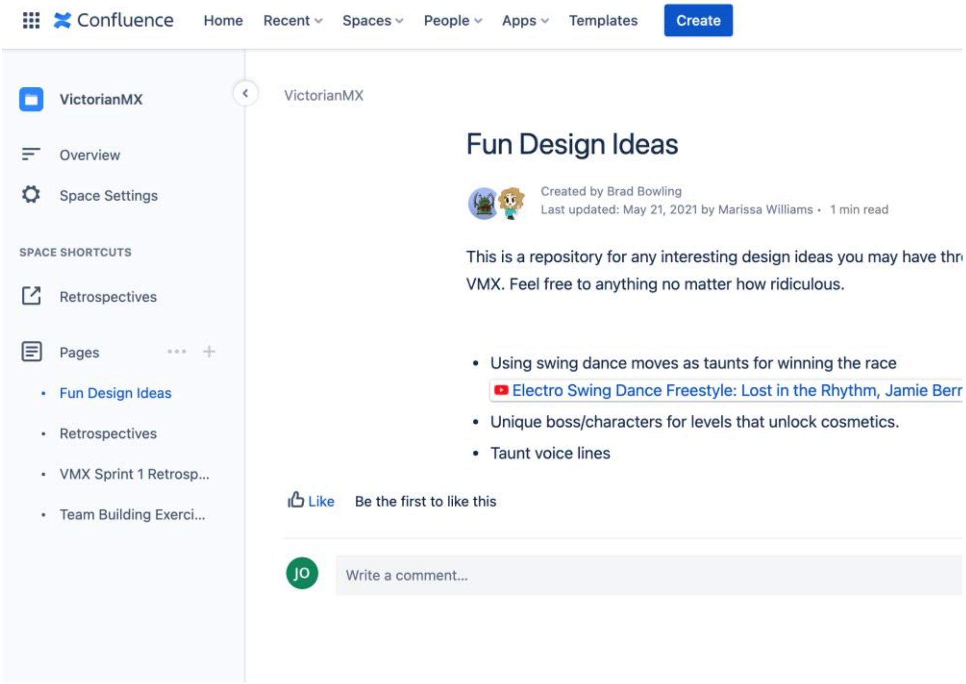
No two development teams are the same and the Victorian Motocross team is exceptionally small, so these steps have been tailored specifically to the VMX team's process without inserting any additional baggage that could hold up the team's work.

Inception Step

The inception step is the first and simplest step of the game development pipeline. This step is when and by whom ideas for the game are generated, whether those ideas are content, functionality, or something else. The Victorian Motocross team has a very open and freeform idea generation process, which is a good thing, but these ideas need to be recorded in order for them to be adequately designed out, costed, and prioritized for the team.

When a team member has an idea for a new piece of content or functionality for the game, they should add a new bullet point to the team's "Fun Design Ideas" page in the team Confluence site. If the idea is self-explanatory or obvious, a single bullet point with one to two sentences will suffice. If the idea is more complex or needs deep explanation, examples, or mockups, a new Confluence page should be created under "Fun Design Ideas" and that page should be linked on the main "Fun Design Ideas" page in a bullet point.

It is important to track ideas for the team in a concrete manner, versus exclusively discussing new ideas in an ephemeral format such as team meetings. Not only does text form guarantee the idea will exist past the end of a call but by using Confluence multiple stakeholders can add to, modify, and leave comments on any design idea for the game.



Fun Design Ideas

Created by Brad Bowling
Last updated: May 21, 2021 by Marissa Williams • 1 min read

This is a repository for any interesting design ideas you may have thr
VMX. Feel free to anything no matter how ridiculous.

- Using swing dance moves as taunts for winning the race
  Electro Swing Dance Freestyle: Lost in the Rhythm, Jamie Berr
- Unique boss/characters for levels that unlock cosmetics.
- Taunt voice lines

Ideation Step

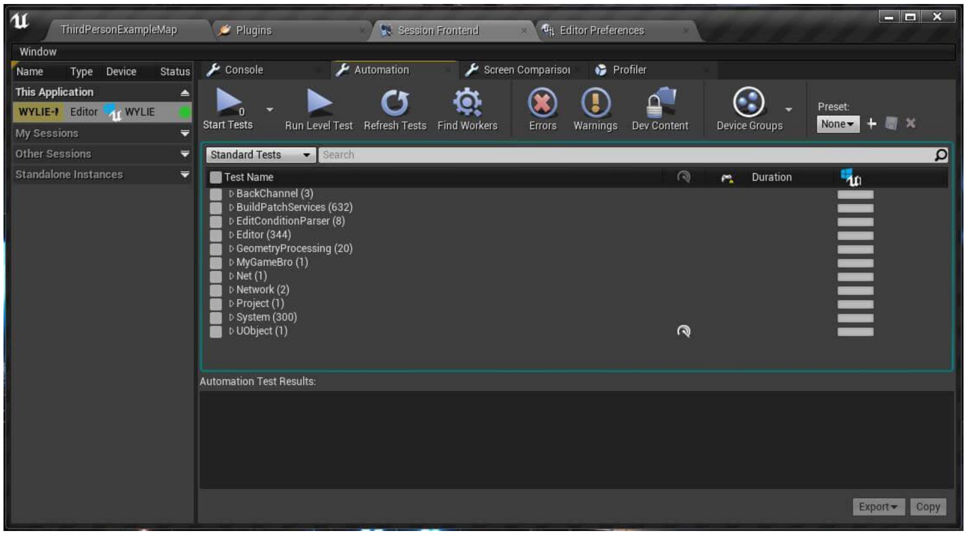
The ideation step is the second step of the game development pipeline and can vary in size a great deal by the type of work being done. This step is when the members of the team, or the team as a whole, iterate on and flesh out an idea before they spend the time and effort to implement it. The Victorian Motocross team currently has very little ideation process in place, which can easily lead to churn as tasks are completed which don't satisfy their creators' intentions.

All major changes and discoveries during the ideation step should be recorded in the idea's individual Confluence page. Using Confluence ensures that work is not lost or forgotten, due to the page edit history and comment threading. Art tasks should include samples, reference images, and mockups where possible in their ideation pages in order to ensure the team has a fuller picture of the expected content. Engineering tasks should include necessary tooling, plugins, and best practices for the implementation step.

Implementation Step

The implementation step is the third step of the game development pipeline and is where the actual content is implemented and integrated into the rest of the game. This step varies by the discipline and type of task, but for all disciplines and tasks the result should be functioning changes within the game.

For engineering tasks specifically, one part of the implementation step should be the creation of unit tests based on the feature's design which confirm the correctness of the engineer's solution. Not all tasks are well suited for unit testing, but Unreal does support both unit tests and integration tests through the "Automation" window. Adding unit tests ensures that both the code being worked on succeeds and that it will be less likely to be broken by future changes to the game's code.

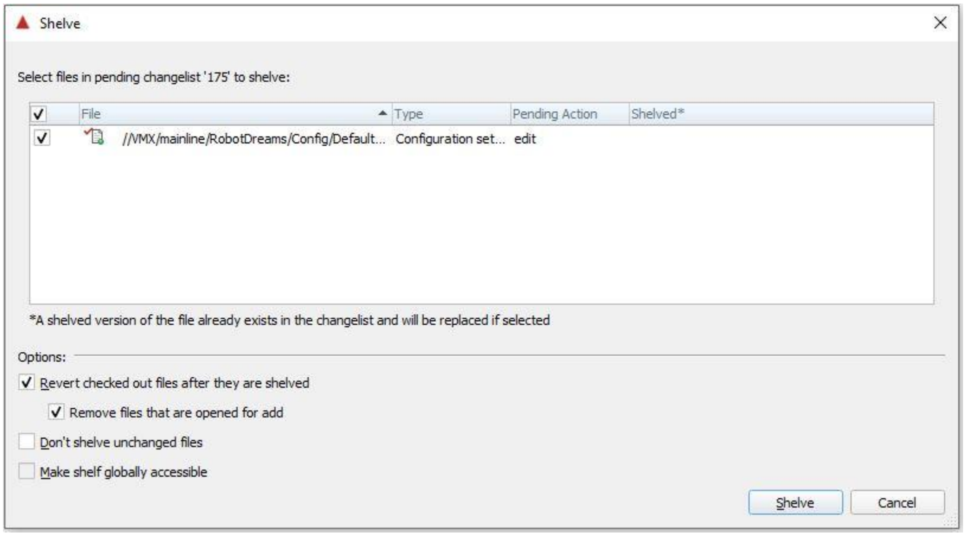


Test Name	Duration
BackChannel (3)	
BuildPatchServices (632)	
EditConditionParser (8)	
Editor (344)	
GeometryProcessing (20)	
MyGameBro (1)	
Net (1)	
Network (2)	
Project (1)	
System (300)	
UObject (1)	

Validation Step

The validation step is the final stage of the general game development pipeline. This step sees to it that the changes made during the implementation step correctly satisfy the intended goals created and developed during the inception and ideation steps. This step also confirms that the build is functional and that the work has been recorded properly.

The best way to validate that a change is correct and functional is through the peer review (aka "buddy build") process. Once a task is believed to have been completed, the team member performing the work should shelve their completed work in Perforce, annotated it with an accurate and verbose description, and send it to a peer for review. The team member should ensure their shelved changelist contains all relevant modified files so that their peer(s) can successfully review their work.



Select files in pending changelist '175' to shelve:

☑	File	Type	Pending Action	Shelved*
☑	//VMX/mainline/RobotDreams/Config/Default...	Configuration set...	edit	

*A shelved version of the file already exists in the changelist and will be replaced if selected

Options:

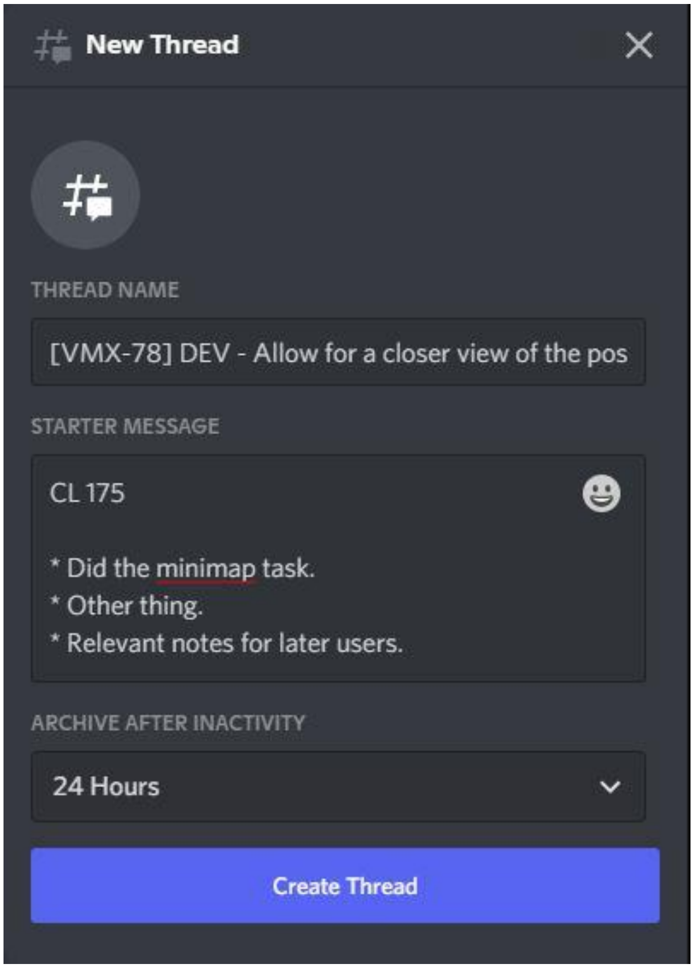
☑ Revert checked out files after they are shelved

☑ Remove files that are opened for add

☐ Don't shelve unchanged files

☐ Make shelf globally accessible

Since the Victorian Motocross team's primary method of open communication is the Discord server, the best way to initiate a peer review is by posting in the relevant discipline's channel. This should be done in a new thread in order to keep crosstalk and confusion to a minimum, and the new thread should be labeled with the information about the changelist to be reviewed.



THREAD NAME

[VMX-78] DEV - Allow for a closer view of the pos

STARTER MESSAGE

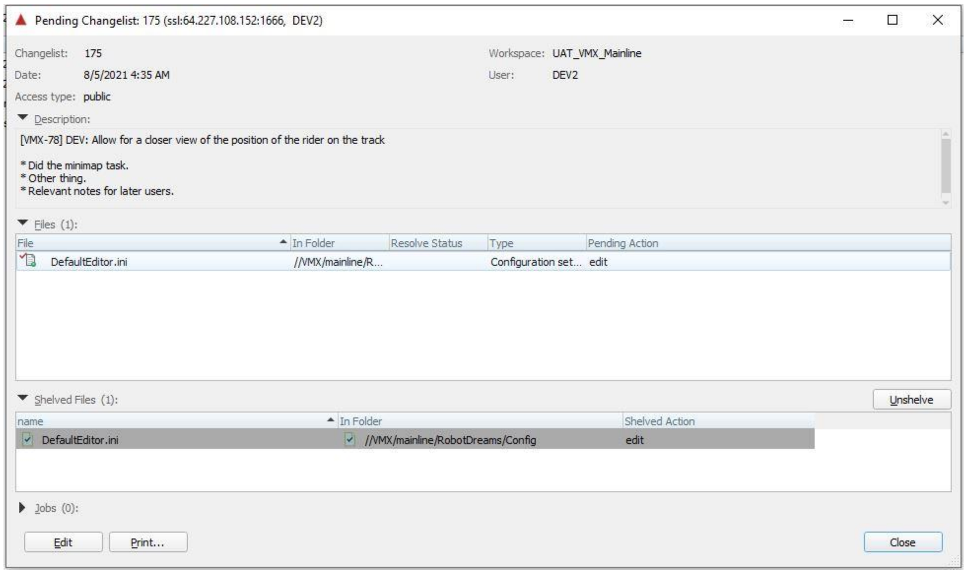
CL 175

* Did the minimap task.
* Other thing.
* Relevant notes for later users.

ARCHIVE AFTER INACTIVITY

24 Hours

Once a peer review request has been posted, other team members should strive to review it as promptly as possible in order to prevent work being slowed by a large degree. Reviewing changes should not be limited to those of the same discipline as the implementor, as the benefit of a “buddy build” is that the changes are confirmed to successfully function on another teammate’s machine instead of just the original implementors. “Buddy build” style peer reviews can prevent significant issues with breaking changes being checked into the source repository directly.



Pending Changelist: 175 (ssl:64.227.108.152:1666, DEV2)

Changelist: 175

Workspace: UAT_VMX_Mainline

Date: 8/5/2021 4:35 AM

User: DEV2

Access type: public

Description:

[VMX-78] DEV: Allow for a closer view of the position of the rider on the track

*Did the minimap task.
*Other thing.
*Relevant notes for later users.

Files (1):

File	In Folder	Resolve Status	Type	Pending Action
DefaultEditor.ini	//VMX/mainline/R...		Configuration set...	edit

Shelved Files (1):

name	In Folder	Shelved Action
☑ DefaultEditor.ini	☑ //VMX/mainline/RobotDreams/Config	edit

Jobs (0):

Once a peer review and/or “buddy build” has been completed, the reviewer should update the Discord thread with their findings and any comments they have. The implementor should reply to any questions or concerns about their implementation and make changes as appropriate to the actionable feedback provided. Once a changelist has been successfully peer reviewed by at least one teammate, it should then be submitted to source control and the task should be updated in JIRA to inform the producer and product owner of the task’s completion.